UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Routing and Mobility on IPv6 over LoWPAN Wireless Mesh Networks

Luís Miguel Lopes de Oliveira

Tese para obtenção do Grau de Doutor em
Engenharia Informática
(3º ciclo de estudos)

Orientador: Prof. Doutor Joel José Puga Coelho Rodrigues
Co-orientador: Prof. Doutor Amaro Fernandes de Sousa

Covilhã, junho de 2018

Dedication

To Elisabete and Inês, the sunshine of my life.

Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor Professor Joel José Puga Coelho Rodrigues and to co-advisor Amaro Fernandes de Sousa for the continuous support of my Ph.D study and related research, for their patience, motivation, generosity and immense knowledge share. Their guidance helped me in all the time of research and writing of this thesis.

I am most grateful to the University of Beira Interior, and the Instituto de Telecomunicações , Covilhã Delegation and Aveiro Polo, for all the support that was given to me.

I would also like to thank all my colleagues of the Next Generation Networks and Applications (NetGNA) research group for all the help, guidance and collaboration.

I would like to thank to Withus, especially to Álvaro Corga, Víctor Abreu, António Oliveira, João Reis, Vítor Silva, for all knowledge share and support regarding to the sensor's hardware and firmware development.

To Instituto Politécnico de Tomar for all the conditions and support. In particular to my great fellows of Informatics Engineering Course.

To my friends for keeping me in the right way and for their persistent support. In particular, António Amaral (Brother), António Trincão (Tó), Frederico Conde (Fred), Luís Almeida (LAA), Rogério Fanha (Roger) and Telmo Silva (Prof. Silva). They are the most accurate definition of trueth friends.

Also, heartfelt thanks to Elisabete and Inês (the sunshine of my life) for their constant help, love and understanding.

My last and deepest gratitude goes to my parents Maria Isabel and Ezequiel Oliveira and Deolinda and Agostinho Conde, and my sister Anabela for constant love and support.

Foreword

This thesis describes the research work performed in the scope of the doctoral research programme and presents its main contributions and achievements. This doctoral programme and inherent research activities were carried out at the Next Generation Networks and Applications Group (NetGNA) research group of the Departamento de Informática, University of Beira Interior, Covilhã, Portugal and Instituto de Telecomunicações, Covilhã Delegation, Portugal. The research work was supervised by Prof. Dr. Joel José Puga Coelho Rodrigues from Universidade da Beira Interior and Prof. Dr. Amaro Fernandes de Sousa from Universidade de Aveiro.

List of publications

Papers included in the thesis resulting from this doctoral research programme.

1. **Routing and mobility approaches in IPv6 over LoWPAN mesh networks**

   L. Oliveira, A. de Sousa and J. Rodrigues
   International Journal of Communication Systems, vol. 24, no. 11, pp. 1445-1466, 2011.

   DOI: 10.1002/dac.1228

2. **Wireless Sensor Networks: A Survey on Environmental Monitoring**

   L. Oliveira and J. Rodrigues
   Journal of Communications, vol. 6, no. 2, pp. 143-151, 2011.

   DOI: 10.4304/jcm.6.2.143-151

3. **Ubiquitous monitoring solution for Wireless Sensor Networks with push notifications and end-to-end connectivity**

   L. Oliveira, J. Rodrigues, A. Elias and B. Zarpelão
   Mobile Information Systems, vol. 10, no. 1, pp. 19-35, 2014.

   DOI: 10.3233/MIS-130170

4. **Wireless sensor networks in IPv4/IPv6 transition scenarios**

   L. Oliveira, J. Rodrigues, A. Elias and G. Han
   Wireless Personal Communications, vol. 78, no. 4, pp. 1849-1862, 2014.

   DOI: 10.1007/s11277-014-2048-9

5. **A network access control framework for 6LoWPAN networks**

   L. Oliveira, J. Rodrigues, A. de Sousa and J. Lloret
   Sensors, vol. 13, no. 1, pp. 1210-1230, 2013.

   DOI: 10.1002/cpe.2850

6. **Denial of service mitigation approach for IPv6-enabled smart object networks**

   L. Oliveira, J. Rodrigues, A. de Sousa and J. Lloret
   Concurrency and Computation: Practice and Experience, vol. 25, no. 1, pp. 129-142, 2012.

   DOI: 10.1002/cpe.2850

7. **Network Admission Control Solution for 6LoWPAN Networks Based on Symmetric Key Mechanisms**

   L. Oliveira, J. Rodrigues, A. de Sousa and V. Denisov
   IEEE Transactions on Industrial Informatics, vol. 12, no. 6, pp. 2186-2195, 2016.

   DOI: 10.1109/TII.2016.2601562

Other publications resulting from this doctoral research programme not included in the thesis

1. **IOT based solution for home power energy monitoring and actuating**

   Oliveira, Luís ML, João Reis, Joel JPC Rodrigues, and Amaro F. de Sousa
   In Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on, pp. 988-992. IEEE, 2015.

   DOI: 10.1109/INDIN.2015.7281869

2. **A WSN solution for light aircraft pilot health monitoring**

   Oliveira, Luís ML, Joel JPC Rodrigues, Bruno M. Mação, Paulo A. Nicolau, and Liang Zhou
   In Wireless Communications and Networking Conference (WCNC), 2012 IEEE, pp. 119-124. IEEE, 2012.

   DOI: 10.1109/WCNC.2012.6213959

3. **End-to-end connectivity IPv6 over wireless sensor networks**

   Oliveira, Luís ML, Joel JPC Rodrigues, Bruno M. Mação, Paulo A. Nicolau, Lei Wang, and Lei Shu
   In Ubiquitous and Future Networks (ICUFN), 2011 Third International Conference on, pp. 1-6. IEEE, 2011.

   DOI: 10.1109/ICUFN.2011.5949126

Resumo

A IoT (*Internet of Things*) tem suscitado o interesse tanto da comunidade académica como da indústria, uma vez que os campos de aplicação são inúmeros assim como os potenciais ganhos que podem ser obtidos através do uso deste tipo de tecnologia. A IoT significa uma rede global de objetos ligados entre si através de uma rede de comunicações baseada em protocolos standard. Neste contexto, um objeto é um objeto físico do dia a dia ao qual foi adicionada a capacidade de medir e de atuar sobre variáveis físicas, de processar e armazenar dados e de comunicar. Estes objetos têm a capacidade de interagir com o meio ambiente envolvente e de cooperar com outros objetos vizinhos de forma a atingirem um objetivo comum. Estes objetos também têm a capacidade de converter os dados lidos em instruções e de as comunicar a outros objetos através da rede de comunicações, evitando desta forma a intervenção humana em diversas tarefas. A maior parte das concretizações de sistemas IoT são baseados em pequenos dispositivos autónomos com restrições ao nível dos recursos computacionais e de retenção de energia. Por esta razão, inicialmente a comunidade científica não considerou adequado o uso da pilha protocolar IP neste tipo de dispositivos, uma vez que havia a perceção de que era muito pesada para os recursos computacionais disponíveis. Entretanto, a comunidade científica e a indústria retomaram a discussão acerca dos benefícios do uso da pilha protocolar em todos os dispositivos da IoT e atualmente é considerada a solução para estabelecer a conetividade entre os dispositivos IoT independentemente do protocolo da camada dois em uso e para os ligar à Internet. Apesar do uso da pilha protocolar IP em todos os dispositivos e da quantidade de soluções propostas, são vários os problemas por resolver no que concerne à integração contínua e sem interrupções da IoT na Internet e de criar as condições para a adoção generalizada deste tipo de tecnologias.

Esta tese versa sobre os desafios associados à integração da IoT na Internet e dos aspetos de segurança da IoT. Relativamente à integração da IoT na Internet o problema é como fornecer informação válida aos dispositivos ligados à Internet, independentemente da versão do protocolo IP em uso, evitando o acesso direto aos dispositivos IoT. Para a resolução deste problema foram propostas e avaliadas soluções baseadas em *web services* REST e em mecanismos de transição IPv4 para IPv6 do tipo pilha dupla (*dual stack*). O *web service* e o mecanismo de transição são suportados apenas no *router* de fronteira, sem penalizar os dispositivos IoT. No que concerne à segurança, o problema é mitigar os efeitos dos ataques de segurança internos e externos iniciados local e remotamente. Foram propostas três soluções diferentes, a primeira é um mecanismo que minimiza os efeitos dos ataques de negação de serviço com origem na Internet e que evita o uso de mecanismos de *firewalls* ineficientes e de gestão complexa. Este mecanismo filtra no router de fronteira o tráfego com origem na Internet é destinado à IoT de acordo com as condições anunciadas por cada um dos dispositivos IoT da rede. A segunda solução, é uma *framework* de *network admission control* que controla quais os dispositivos que podem aceder à rede com base na autorização administrativa e que aplica políticas de conformidade relativas à segurança aos dispositivos autorizados. A terceira é um mecanismo de *network admission control* para redes 6LoWPAN que evita que dispositivos não autorizados comuniquem com outros dispositivos legítimos e com a Internet o que reduz drasticamente o número de ataques à segurança. Este mecanismo também foi explorado como um mecanismo de gestão uma vez que pode ser utilizado a dimensão da rede quanto ao número de dispositivos, tornando-a mais fácil de gerir e aumentando a sua fiabilidade e o seu tempo de vida.

Palavras-chave

Internet das Coisas; Redes de Sensores Sem Fios; 6LoWPAN; IPv6; Network Admission Control; RESTFul web services.

Resumo alargado

Esta secção resume, de forma alargada, o trabalho de investigação no âmbito da tese de doutoramento com o título: "Routing and Mobility on IPv6 over LoWPAN wireless Mesh Networks". O foco desta tese é o estudo e proposta de soluções que facilitem a integração da Internet das coisas (IoT) na Internet e dos aspetos de segurança associados à sua integração. Numa primeira fase é descrito o enquadramento da tese, são definidos os problemas abordados e os principais objetivos do estudo. De seguida é apresentada a hipótese de investigação e as principais contribuições deste trabalho para o avanço do estado da arte. No final são apresentadas as principais conclusões assim como são identificados tópicos para trabalho futuro.

Enquadramento da tese

A evolução nos últimos anos da microeletrónica, dos sistemas eletromecânicos e dos sistemas de comunicação foram determinantes para o desenvolvimento da IoT. A IoT tem suscitado o interesse tanto da comunidade académica como da indústria, uma vez que os campos de aplicação são inúmeros assim como os potenciais ganhos que podem ser obtidos através do uso deste tipo de tecnologia [1][2]. De entre as muitas áreas de aplicação podem destacar-se os sistemas militares, a proteção do meio ambiente, a gestão de energia, a saúde, a domótica e os transportes. Do ponto de vista semântico, a IoT significa uma rede de objetos inteligentes de âmbito global suportada em protocolos standard e que podem ser endereçados através de um identificador único [3].

Neste contexto, um objeto inteligente é um objeto do nosso quotidiano ao qual foi adicionada a capacidade de medir e atuar sobre variáveis físicas (tais como a temperatura ou a luminosidade), de processar e de armazenar dados e de comunicar através de uma rede de dados. Estes objetos inteligentes são capazes de interagir com o meio ambiente envolvente e de cooperar com outros objetos similares de forma a atingirem um objetivo comum. Os objetos inteligentes têm também a capacidade de converter os dados que recolhem em instruções que enviam a outros objetos através de uma rede de comunicações, evitando desta forma a intervenção humana em diversas tarefas. A origem do termo IoT e do seu significado inicial não é conhecido com precisão. O termo IoT foi provavelmente definido por Kevin Ashton e foi inicialmente associado à tecnologia RFID [4]. O protocolo IEEE802.15.4 é utilizado pela maioria de sensores e atuadores de baixo custo, razão pela qual veio substituir o RFID como tecnologia base da IoT. É neste contexto que aparece o termo redes de sensores sem fios [5][6]. Recentemente, outras tecnologias da camada 2, nem sempre destinadas ao uso em dispositivos com baixos recursos, foram associadas reforçando o conceito de IoT anteriormente descrito.

Apesar da inclusão de outras tecnologias, onde a quantidade de recursos e de energia necessária para o seu funcionamento não é uma preocupação, na maioria das situações as aplicações da IoT são suportadas por pequenos dispositivos com baixos recursos computacionais e de capacidade de retenção de energia reduzida, formando desta forma uma rede sem fios de baixa potência e com perda de pacotes (LLN) [7].

Numa fase inicial a comunidade científica não considerou adequado o uso da pilha protocolar IP em sistemas de baixo consumo e com capacidade de processamento e de armazenamento reduzido, uma vez que havia a perceção de que os recursos exigidos não eram compatíveis com

os deste tipo de dispositivos [8]. Entretanto, a comunidade científica e a indústria começaram a repensar muitas das ideias erradas acerca do uso da pilha protocolar IP em todos os dispositivos, independentemente da quantidade de recursos disponíveis. Em primeiro lugar, a pilha protocolar IP permite a comunicação entre todos os dispositivos independentemente dos protocolos da camada 2 em uso, tais como o WirelessHart e o ZigBee, perpetuando desta forma o paradigma *"end to end"* sob o qual a Internet se tem desenvolvido [9]. Em segundo lugar, o desenvolvimento de aplicações para funcionar em redes IP é aberto, segue um paradigma conhecido e aceite de forma generalizada que usa protocolos standard e não sujeitos ao pagamento de utilização (*royalty free*). Acresce o facto de já existirem ferramentas para o desenvolvimento de aplicações, para a configuração, para a gestão e para o *debugging* que podem ser utilizadas tais como estão ou com pequenas adaptações. Finalmente, uma solução baseada na pilha protocolar IP evita o uso de *Gateways* e de *Proxies* necessários para ligar redes incompatíveis com o protocolo IP à Internet que são difíceis de manter e de gerir e cujo funcionamento é complexo [8]. Apesar do protocolo IPv6 disponibilizar endereços suficientes para ligar todos os dispositivos, ele não foi projetado para funcionar sobre dispositivos com baixos recursos computacionais e com baixa capacidade de retenção de energia. De forma a ultrapassar estes constrangimentos foi definida uma camada de adaptação, designada por 6LoWPAN, que opera entre as camadas de rede e de ligação e que define novas abordagens de encaminhamento, a compressão de cabeçalhos, a fragmentação de pacotes a baixo dos 1280 Bytes, de autoconfiguração e os mecanismos de descoberta de vizinhos (*neighbor discovery*) mais adaptados a este tipo de dispositivos [10].

Uma das primeiras arquiteturas propostas para a IoT considera três camadas: a de perceção, a de rede e a de aplicação [11]. A camada de perceção é executada nos sensores e é responsável por medir e adquirir informação a partir do meio ambiente envolvente. A camada de rede é responsável pela comunicação entre sensores e com todos os outros elementos da rede, tais como, servidores, *Gateways*, entre outros. Finalmente, a camada de aplicação é responsável por extrair a informação a partir do enorme volume de dados recolhido e por os disponibilizar aos utilizadores através das interfaces mais adequadas.

Apesar do modelo de três camadas refletir as ideias chave da IoT, não é suficiente para descrever com detalhe todos os aspetos mais relevantes. Assim, foram propostos mais modelos de arquitetura com mais camadas [12]. Um desses modelos apresenta cinco camadas e adiciona ao anterior modelo mais duas: a de processamento e a de negócio. Por conseguinte, as camadas deste modelo são: a de perceção, a de transporte, a de processamento, a de aplicação e a de negócio. A camada de perceção e a de aplicação têm a mesma função que as do modelo anterior de três camadas. A de transporte é responsável por transferir através da infraestrutura de rede os dados recolhidos pelos dispositivos da IoT entre as camadas de perceção e de processamento. A infraestrutura de rede pode ser baseada em Ethernet, IEEE 802.15.4, WiFi ou 5G, entre outras. A camada de processamento é responsável por armazenar, analisar e processar os dados provenientes da camada de transporte. Esta camada, também designada por middleware processa a informação e toma decisões de forma automática baseadas nos resultados obtidos. Os resultados são posteriormente enviados para a camada de aplicação. As bases de dados, a computação na nuvem e o processamento de grandes volumes de dados são exemplos de tecnologias utilizadas nesta camada. A camada de negócio é responsável por gerir todo o sistema de IoT, incluindo as aplicações, os modelos de negócio e os aspetos relacionados com a privacidade dos utilizadores. Apesar de ser considerado um novo paradigma, a IoT não pode ser considerada uma tecnologia completamente nova, uma vez que se baseia em outros paradigmas e protocolos utilizados atualmente na Internet. Contudo, as arquiteturas de rede existentes

não permitem concretizar a visão subjacente à IoT, uma vez que não é possível a integração de forma suave deste tipo de dispositivos na Internet. A quantidade de dispositivos a incorporar, o facto de serem baseadas em dispositivos de baixos recursos computacionais e a sua diversidade estão na origem da maioria dos problemas de integração [13].

Uma grande parte do trabalho de integração já está concluído, mas persistem ainda muitos problemas por resolver antes de se atingir a plena integração das redes IoT na Internet e do uso de forma generalizada deste tipo de tecnologia. Os problemas a resolver e os requisitos podem ser classificados de acordo com: a interoperabilidade, a flexibilidade, a escalabilidade, a energia, a gestão da mobilidade e a segurança [13].

São três os tipos principais de problemas relacionados com interoperabilidade [13]: os técnicos, os semânticos e os pragmáticos. Os desafios técnicos estão relacionados com os recursos dos dispositivos e com os protocolos que suportam. Os desafios semânticos estão relacionados com a capacidade dos componentes da solução de IoT interpretarem e processarem os dados recebidos por outros dispositivos. Os desafios pragmáticos estão relacionados com a capacidade dos componentes do sistema de analisarem as intenções de outros componentes. A flexibilidade é necessária para assegurar que estão disponíveis os recursos indispensáveis ao funcionamento da grande diversidade de aplicações da IoT. A escalabilidade é também um fator crítico, uma vez que as arquiteturas de IoT necessitam de crescer de forma a acomodar o número elevado de dispositivos.

Grande parte dos dispositivos utilizados na camada de perceção são caracterizados por terem capacidade limitada de processamento e de armazenamento de dados e de retenção de energia [3]. Por conseguinte, os protocolos que necessitem de uso intensivo da unidade de processamento, de armazenamento de grandes quantidades de dados e da transmissão intensiva de mensagens devem ser evitados. A eficiência do uso de energia é atingida quando estes dispositivos são mantidos no estado adormecido, no qual a interface de rádio é mantida desligada na maior parte do tempo e o processador mantido num estado de baixo consumo sempre que possível [14]. Não obstante, esta abordagem de poupança de energia deve ser utilizada com cuidado devido ao impacto que provoca no funcionamento de toda a solução de IoT e pode ser preferível recorrer a soluções que permitam recolher energia a partir do meio ambiente envolvente de forma a minimizar o impacto do uso de soluções de poupança de energia no funcionamento global da rede [15].

O suporte da mobilidade e a sua gestão é crucial para a maioria das redes compostas por dispositivos com baixos recursos computacionais e de retenção de energia, sendo que neste caso, tanto a mobilidade entre redes como a mobilidade dentro da mesma rede devem ser consideradas. Enquanto que a mobilidade que resulta de uma mudança de localização do dispositivo é fácil de detetar, a que resulta das mudanças de topologia provocadas pelo desvanecimento do sinal e pelas variações frequentes da relação sinal ruído das ligações não é tão óbvia [16].

A segurança é provavelmente o desafio que mais constrangimentos tem provocado à adoção generalizada dos serviços da IoT [17]. A segurança é fundamental para evitar a inoperacionalidade dos serviços, a quebra da confidencialidade e integridade dos dados durante a transmissão e armazenamento e a violação da privacidade. Tal como todos os sistemas abertos e integrados na Internet, as LLN estão sujeitas a um número elevado de vulnerabilidades, algumas delas são consequência do facto dos dispositivos terem baixos recursos computacionais e restrições ao nível do consumo de energia [18]. Razão pela qual este tipo de dispositivos está mais exposto a ataques de segurança que os dispositivos que não apresentam tais constrangimentos.

A maioria dos novos desafios relativos à segurança são também consequência da ausência de uma infraestrutura de rede estável e bem definida. Acresce ainda o facto de não ser possível aplicar nestes dispositivos os mesmos métodos e mecanismos que normalmente se utilizam para proteger os dispositivos sem constrangimentos ao nível da capacidade de processamento e de armazenamento e de consumo de energia. Nestes cenários a utilização de mecanismos de segurança está muito condicionada aos recursos disponíveis, pelo que qualquer acréscimo no processamento e no envio e receção de mensagem deve ser avaliado com cuidado. Foram entretanto propostos novos mecanismos de segurança, alguns deles destinados a mitigar os efeitos de ataques bem definidos. Novas abordagens e mecanismos de segurança são necessários de forma a endereçar os novos desafios colocados pela IoT [19].

São vários os critérios que podem ser utilizados para classificar os ataques de segurança. Em primeiro lugar, os ataques de segurança podem ser classificados em internos e externos, de acordo com a propriedade dos recursos utilizados na concretização do ataque [20]. Nos ataques externos, o atacante usa os seus recursos para concretizar do ataque. Nos ataques internos, o atacante em primeiro lugar compromete um ou mais recursos legítimos (normalmente através da injeção de código malicioso ou acedendo a dados críticos armazenados em sistemas legítimos). Esses recursos são posteriormente utilizados na consecução do ataque de segurança. Note-se que os ataques internos são mais difíceis de detetar porque os recursos comprometidos aparentam ser legítimos, tendo por isso a sua confiança dos recursos legítimos. Em segundo lugar, os ataques podem ser classificados em passivos e ativos, de acordo com o facto de modificarem os dados ou o seu fluxo normal [21]. Os ataques passivos baseiam-se na aquisição de dados sem que exista alteração dos dados ou do seu fluxo. Os ataques ativos envolvem a modificação dos dados, através da injeção de dados falsos ou do normal fluxo dos dados. Finalmente, os ataques de segurança podem ainda ser agrupados em três grandes grupos, de acordo com a garantia de segurança que pretendem comprometer, em ataques contra a confidencialidade e autenticidade, ataques contra a disponibilidade e em ataques do tipo *stealthy*. Enquanto a escuta, o reenvio de pacotes, a falsificação e os ataques de *spoofing* são exemplos de ataques pertencentes ao primeiro grupo, a negação de serviço (*Denial of Service*) é um ataque contra a disponibilidade [22]. Um ataque de negação de serviço pode ser definido como qualquer ação que reduza ou elimine a capacidade do sistema para realizar as tarefas para o qual foi concebido. Os ataques de negação de serviço são comuns porque podem ser concretizados com recurso a equipamentos vulgares e não necessitam de conhecimentos técnicos avançados. Note-se que o efeito de um ataque de negação de serviço é mais penalizador nas redes LLN do que nos outros tipos de redes. Este facto resulta da diferença de recursos que existe entre os dispositivos da LLN e os dispositivos regulares que normalmente se ligam à Internet. É por este motivo que a mitigação dos efeitos dos ataques de negação de serviço é considerada como um problema em aberto. Em primeiro lugar, os dispositivos das LLN não podem suportar o acréscimo de processamento provocado pelos mecanismos que normalmente são utilizados para conter este tipo de ataques. Em segundo lugar, um número limitado de pacotes é suficiente para esgotar a energia disponível nos dispositivos tornando a LLN inoperacional.

Definição do problema e dos objetivos de investigação

Os objetivos deste trabalho evoluíram ao longo da sua realização. Inicialmente, o problema consistia na proposta e validação de soluções que endereçassem simultaneamente os requisitos de encaminhamento e de mobilidade das redes IPv6 sem fios em malha e de baixa potência e com

perda de pacotes. No início, estes eram os problemas mais importantes a serem resolvidos e era considerado cedo para se endereçarem os problemas relacionados com a segurança, uma vez que persistiam problemas relacionados com a conectividade. Contudo em 2012 foi proposto um protocolo de encaminhamento, designado por RPL, que endereça os problemas mais relevantes de encaminhamento nas redes LLN em malha e com múltiplos saltos (*multihop*) e a micro mobilidade [23]. O RPL foi aceite como standard, o foco da investigação em consequência destes desenvolvimentos, evoluiu para o estudo dos desafios associados à interoperabilidade entre as redes da IoT e a Internet e dos aspetos de segurança relacionados com a IoT.

No que concerne à interoperabilidade entre os dispositivos da IoT e a Internet, o problema é como permitir que os dispositivos ligados à Internet obtenham dados úteis, sem que seja necessário aceder diretamente a estes dispositivos. O objetivo de investigação é o desenvolvimento e a validação de soluções que (i) minimizem a quantidade de comunicações necessárias na rede LLN (aumentando desta forma o tempo de vida dos dispositivos), (ii) o uso de mecanismos aceites de forma generalizada que facilitem o acesso dos dispositivos ligados à Internet aos serviços da IoT e (iii) suportar dispositivos ligados à Internet independentemente da versão do protocolo IP em uso. Para atingir estes objetivos, foram estudados os *Web Services* e os mecanismos de transição IPv4 para IPv6 como potenciais soluções para resolver os problemas de conectividade ao nível das camadas de rede e de aplicação, respetivamente [24][25].

Os *web services* tornaram-se um standard de facto para a distribuição de serviços entre sistemas remotos e heterogéneos, uma vez que disponibilizam interfaces bem definidas aos sistemas distribuídos as quais são independentes do hardware, do sistema operativo e das linguagens de programação utilizadas no desenvolvimento do servidor e do cliente [24]. A interoperabilidade é maioritariamente fornecida pelo standard *Extensible Markup Language* (XML) e por isso não existem problemas relativos à conversão de formatos. O *Simple Object Access Protocol* (SOAP) é um protocolo de comunicação entre aplicações em que as mensagens são definidas em XML e são auto descritivas. Os *web services* podem continuar a ser utilizados para disponibilizar serviços da IoT aos dispositivos móveis, tais como *smart phones*. Contudo, aplicar a arquitetura atual dos *web services* aos dispositivos móveis pode resultar na redução de desempenho provocado pela codificação e descodificação do XML que é utilizado nas mensagens SOAP [26]. Os RESTful *web services* tentam emular os protocolos HTTP e outros similares restringindo a interface a operações a comandos standard, tais como, o GET, o PUT, o POST e o DELETE. Assim, os dados e os serviços são considerados recursos e são acedidos a partir de ligações *web*, ou seja, de identificadores *Uniform Resource Identifiers* (URIs). A arquitetura REST é do tipo cliente servidor e foi desenvolvida para ser utilizada em conjunto com protocolos do tipo stateless, tais como o HTTP [27]. Por se tratar de uma arquitetura leve é adequada para disponibilizar os serviços da IoT diretamente aos dispositivos móveis ou com recurso à computação na nuvem. Os *web services* devem ser suportados pelo *router* de fronteira que é utilizado para ligar a rede IoT à Internet, uma vez que este dispositivo é normalmente alimentado através de uma fonte de energia externa e tem habitualmente mais recursos que os dispositivos e utilizados nas redes LLN.

Tal como mencionado anteriormente, o IPv6 é o protocolo de rede mais adequado à IoT. Contudo, apesar da grande maioria dos sistemas operativos atuais já suportarem de forma nativa as duas versões do protocolo IP (IPv4 e IPv6), a maioria das redes de acesso apenas disponibiliza acesso baseado no protocolo IPv4. Assim, disponibilizar mecanismos de transição IPv4 para IPv6 no router de fronteira pode potenciar a adoção dos serviços baseados na IoT sem sobrecarregar

os dispositivos da rede LLN.

Relativamente à segurança, mitigar os efeitos dos ataques internos e externos iniciados dentro e fora da LLN é o problema de investigação desta tese. O objetivo de investigação é a formulação e validação de soluções de segurança adequadas às redes LLN com suporte *multi-hop* que (i) detetem e autentiquem os dispositivos da rede, (ii) impeçam que dispositivos não autorizados usem a infraestrutura de rede para comunicar com os dispositivos legítimos e com a Internet, (iii) assegurar que os dispositivos estão de acordo com a postura de segurança definida, e (iv) filtrar no *router* de fronteira as mensagens provenientes da Internet e destinadas aos dispositivos da LLN de acordo com a regras definidas. De forma a satisfazer estes objetivos de investigação, foram estudados e avaliados os mecanismos de *firewall* e de *network admission control*.

Tradicionalmente recorre-se ao uso de *firewalls* para filtrar o tráfego entre dois domínios de segurança diferentes, permitindo apenas o que está de acordo com a política de segurança em vigor [28]. O tráfego é filtrado de acordo com as regras definidas estaticamente pelo administrador da rede. As *firewalls* não são eficazes para proteger as redes LLN, porque se tratam de redes compostas por um elevado número de dispositivos heterogéneos relativamente ao hardware e às funcionalidades que disponibilizam e com comportamento muito dinâmico. Assim, o objetivo é propor outro tipo de mecanismo que filtre igualmente o tráfego indesejado evitando as desvantagens do uso de *firewalls*. Este novo mecanismo de filtragem deve ser suportado apenas pelo router de fronteira e as regras de filtragem devem ser definidas dinamicamente de acordo com a informação enviada pelos dispositivos da rede LLN. Tal como nas *firewalls* tradicionais, o tráfego não suportado deve ser filtrado no router de fronteira e o tráfego suportado deve ser regulado de forma a garantir que não são encaminhados para a LLN mais pedidos, mesmo que válidos, do que o limite imposto.

A capacidade de auto-organização e de autoconfiguração são duas das características das redes LLN. De facto, estas características são desejáveis uma vez que minimizam o esforço de configuração e simultaneamente aumentam a robustez da rede. No entanto podem ser exploradas por ataques de segurança [19]. Uma solução de *network admission control* pode ser utilizada com dois propósitos. Em primeiro lugar, pode ser utilizada para gerir a dimensão da rede em termos do número de dispositivos, tornando-a mais fácil de gerir, aumentando a fiabilidade e o tempo de vida da rede. Em segundo lugar, pode ser utilizada como mecanismo de segurança uma vez que se os dispositivos maliciosos forem impedidos de usar a infraestrutura de rede, as comunicações com outros dipositivos não são possíveis e, por conseguinte, o número de ataques internos e externos é drasticamente reduzido. Uma solução de *network admission control* deve compreender os mecanismos que detetem a presença de dispositivos na rede, que realizem a autenticação e a verificação da integridade dos dispositivos, verifiquem a integridade das mensagens e a autenticação da origem.

Hipótese de investigação

Esta tese propõe mecanismos e estratégias que promovam a integração das redes da IoT na Internet e que minimizem o impacto dos ataques internos e externos iniciados por dispositivos ligados à mesma rede e a redes externas. O argumento apresentado nesta tese é o seguinte:

A visão subjacente à IoT considera que existe conectividade e interoperabilidade sem descontinuidades entre as redes LLN e a Internet, sendo que cada LLN é composta por um grande número de dispositivos com baixos recursos e com baixa retenção de energia e que geram um

enorme volume de dados. A concretização da visão da IoT já não é compatível com o paradigma end to end, os mecanismos tradicionalmente localizados nos dispositivos terminais deverão passar a ser suportados pelos dispositivos de rede intermédios. Os routers de fronteira são os elementos da rede mais apropriados para suportar tais mecanismos, uma vez que são alimentados por fontes de energia externa e não apresentam constrangimentos severos ao nível dos recursos computacionais. Através desta abordagem é possível alcançar alguns dos principais objetivos da IoT. Por um lado, é possível disponibilizar os serviços da IoT aos dispositivos ligados à Internet sem que seja suportada conetividade end to end, reduzindo o tráfego nas LLN. Por outro lado, é possível mitigar efetivamente ataques de segurança com origem externa e interna. Os ataques com origem externa podem ser mitigados evitando o recurso às tradicionais firewalls. Os ataques com origem interna podem ser mitigados através do uso de mecanismos de network admission control, sem penalizar o desempenho dos dispositivos da IoT e simultaneamente assegurar o controlo administrativo e que estão em conformidade com as políticas de segurança em vigor.

De forma a sustentar este argumento, foi utilizada a seguinte abordagem:

Considerando a relevância das LLN para o sucesso da IoT, especificamente das redes de sensores sem fios, foram estudados detalhadamente os protocolos das camadas 2 e 3 utilizados neste tipo de redes. Através deste estudo, foram identificadas as soluções disponíveis assim como os problemas para os quais ainda não existem soluções satisfatórias. De seguida foram revistas e estudadas de forma detalhada as principais soluções e projetos aplicados à monitorização ambiental com o objetivo de avaliar o desempenho das tecnologias aplicadas às redes de sensores sem fios quando sujeitas à operação em ambientes severos e sem vigilância. Através destes estudos, foram identificadas as principais contribuições que serviram de ponto de partida para a proposta de novas abordagens que promovam a integração das redes da IoT na Internet e que simultaneamente mitiguem os efeitos dos ataques internos e externos independentemente da sua origem.

Foi configurada em ambiente laboratorial uma rede de sensores sem fios 6LoWPAN, com dispositivos TinyOS e suportada no protocolo IEEE 802.15.4 da camada 2. Nesta experiência, a conectividade *end-to-end* era suportada uma vez que o router de fronteira apenas encaminhava pacotes entre uma rede ethernet e a rede 6LoWPAN. Esta experiência foi utilizada como referência na avaliação dos novos mecanismos propostos nesta tese.

De seguida, foi considerada uma nova abordagem baseada em *web services* RESTfull. O web service foi instalado no router de fronteira e a conectividade *end-to-end* deixou de ser suportada. Nesta abordagem o *web service* foi utilizado para cumprir dois objetivos. Em primeiro lugar, para fornecer um método mais eficiente para aceder aos servidores disponibilizados pelos dispositivos da IoT. Em segundo, para facilitar o desenvolvimento das aplicações móveis que acedem aos serviços disponibilizados pela IoT, uma vez que é fornecida uma interface de acesso bem definida e baseada em tecnologias standard. Foram definidos três métodos que podem ser utilizados pelas aplicações móveis para aceder aos serviços. No primeiro método, a aplicação móvel solicita ao *webserver* o envio de dados históricos, assim como as últimas leituras. Estes valores encontram-se armazenados numa base de dados relacional. No segundo método a aplicação móvel envia os pedidos diretamente ao router de fronteira, o qual funciona como um *proxy*. Finalmente, no terceiro método o mecanismo de *push* envia uma notificação à aplicação quando os dados recolhidos ultrapassam o limite definido. A avaliação e a validação desta solução foram realizadas com recurso a uma rede laboratorial.

Um processo de integração gradual e sem disrupções tem de ter em conta o facto de existir ainda um número considerável de dispositivos ligados à Internet que se encontram privados do acesso por IPv6. Os web services podem também ser utilizados para disponibilizar serviços da IoT independentemente da versão do protocolo IP suportado pelos dispositivos ligados à Internet. Para atingir este objetivo, foi utilizado em conjunto com o *web service* um mecanismo de transição IPv4 para IPv6. Os dois mecanismos são executados no router de fronteira de forma a poupar os recursos, por si só escassos, dos dispositivos da rede LLN.

Ligar os dispositivos IoT à Internet vai expô-los a novos ataques de segurança, mesmo quando a conectividade *end to end* não é suportada ou é controlada. Os ataques de segurança mais relevantes foram identificados e estudados detalhadamente, assim como as soluções propostas para mitigar os seus efeitos. Foi dedicada particular atenção aos ataques de negação de serviço, uma vez que são dos mais frequentes e dos mais penalizadores quando os recursos do atacante excedem os do alvo do ataque, caso que se verifica quando o destino do ataque é um dispositivo da rede LLN. Foram investigados dispositivos que filtrassem no router de fronteira o tráfego que pudesse resultar num ataque de negação de serviço. Foram avaliados em primeiro lugar os mecanismos de *firewall*. No entanto nenhum dos mecanismos estudados demonstrou ser adequado para ser utilizado neste tipo de ambiente. Em primeiro lugar, devido à diversidade de dispositivos que podem pertencer à mesma rede IoT e à multiplicidade de tarefas que desempenham, sendo por isso necessário uma grande quantidade de regras. Em segundo lugar, as redes IoT têm um comportamento muito dinâmico e, por conseguinte, assegurar que num dado momento estão a ser utilizadas as regras mais adequadas é uma tarefa complexa. Assim, foi estudado e desenvolvido um mecanismo para filtrar no router de fronteira os pacotes com origem em dispositivos ligados à Internet e destinados às redes LLN. Neste novo mecanismo, os dispositivos da LLN recorrem às mensagens do protocolo *Neighbor Discovery* do 6LoWPAN para declararem ao *router* de fronteira em que condições pretendem aceitar pacotes provenientes da Internet e com que frequência.

A capacidade de auto-organização e autoconfiguração são características das redes IoT, que permitem minimizar as tarefas de configuração da rede e simultaneamente aumentam a sua robustez devido à capacidade da infraestrutura se adaptar rapidamente às mudanças de topologia. No entanto, estas caraterísticas podem ser exploradas por dispositivos ligados à mesma rede LLN para realizar ataques à segurança. Os mecanismos de *network access control* podem ser utilizados para bloquear os pacotes provenientes de dispositivos potencialmente maliciosos reduzindo desta forma os ataques à segurança que podem ser realizados. Os mecanismos de *network access control* foram também avaliados como ferramentas de gestão uma vez que permitem controlar a dimensão da rede em termos do número de nós, tornando-a mais fácil de gerir e consequentemente aumentar a sua fiabilidade e o tempo de vida. Com base nesta investigação, foi proposto uma *framework* e novo mecanismo de *network access control* que suporta o seu funcionamento nas seguintes operações nucleares: aprovisionamento dos dispositivos, deteção da presença dos dispositivos, autenticação e autorização dos dispositivos, propagação da lista de dispositivos autorizados e filtragem de pacotes. O uso de algoritmos de cifra simétrica e a reutilização das mensagens dos protocolos necessários para o funcionamento deste tipo de redes, tais como, o *Neighbor Discovery* do 6LoWPAN foram tidos em consideração de forma a reduzir a sobrecarga provocada pelos mecanismos de *network access control*. Foi concebida uma rede laboratorial baseada em *hardware* e nos sistemas operativos mais utilizados para validar a solução de *network acess control* proposta.

Principais contribuições

Esta secção resume as principais contribuições resultantes do trabalho de investigação apresentado nesta tese.

A primeira e a segunda contribuições desta tese são dois estudos do estado da arte realizados numa fase inicial deste trabalho. A primeira contribuição consiste no estudo do estado da arte acerca das soluções existentes para suportar o encaminhamento e o suporte para a mobilidade em redes de sensores sem fios em malha compatíveis com o protocolo 6LoWPAN. Uma das principais conclusões desta contribuição é o facto do 6LoWPAN ser considerado a solução de convergência, que permite a conectividade entre dispositivos das redes LLN, independentemente do protocolo da camada 2 suportado e que simultaneamente facilita a integração destas redes na Internet. Este estudo é descrito no capítulo 2 e foi publicado na revista International Journal of Communication Systems [29]. A segunda contribuição é o estudo do estado da arte acerca do uso das redes de sensores sem fios na monitorização ambiental. Este estudo identifica os desafios que devem ser ultrapassados no sentido de construir uma solução de monitorização ambiental baseada em redes de sensores sem fios de baixo custo. Esta contribuição foi publicada na revista Journal of Communications e encontra-se descrita no capítulo 3 [30].

A terceira contribuição é uma solução baseada em redes de sensores sem fios que recorre ao uso de um *web Service*. Os dados recolhidos em tempo real pela rede de sensores sem fios são enviados diretamente para um dispositivo móvel ou armazenados numa base de dados relacional. A aplicação móvel interage com o web service através de uma interface bem definida para aceder aos dados recolhidos, evitando a conectividade *end to end* entre os dispositivos ligados à Internet e os sensores. Um mecanismo do tipo push notification foi especificado para enviar alertas para a aplicação móvel quando ocorrem situações limite previamente definidas. A arquitetura proposta e a aplicação móvel foram avaliadas e validadas usando uma rede laboratorial e estão disponíveis para serem utilizadas. Esta contribuição encontra-se no capítulo 4 e foi publicada na revista Mobile Information Systems International Journal [31].

A quarta contribuição é a solução que combina mecanismos de transição IPv4 para IPv6 com o *web service*, que tem como principal objetivo permitir a interação entre as aplicações móveis independentemente da versão IP suportada pelo dispositivo móvel onde estão instaladas e os serviços alojados nas redes compatíveis com o protocolo 6LoWPAN. Tanto o *web service* como o mecanismo de transição IPv4 para IPv6 são suportados unicamente no router de fronteira de forma a poupar os recursos dos dispositivos da rede de sensores. Esta contribuição foi publicada na revista Wireless Personal Communications International Journal e encontra-se no capítulo 5 [32].

A quinta contribuição propõe um novo mecanismo para mitigar os efeitos dos ataques de negação de serviço iniciados na Internet e com destino às redes 6LoWPAN. É proposto o uso de uma versão adaptada do protocolo *neighbor discovery* do 6LoWPAN, na qual as mensagens deste protocolo são utilizadas pelos dispositivos da LLN para notificar o router de fronteira acerca do tipo de mensagens que estão disponíveis para receber, assim como da sua frequência, enviadas pelos dispositivos ligados à Internet. Esta contribuição está disponível no capítulo 6 e na forma de artigo na revista Concurrency Computation: Practice and Experience international journal [33].

A sexta contribuição é a proposta de uma *network access security framework* que se destina a controlar os dispositivos que têm acesso à rede, baseada na aprovação administrativa e na aplicação de políticas de conformidade relativas à segurança aos dispositivos autorizados. Na

proposta, a *framework* (i) controla quais os dispositivos que podem aceder à rede ao nível da camada de rede e (ii) aplica as regras de conformidade de segurança aos dispositivos. A *framework* pode ser utilizada simultaneamente como uma ferramenta de gestão e como um mecanismo de segurança. A *framework* proposta usa os protocolos LSEND, o RPL e o Seluge. Ao contrário das soluções de *network admission control* definidas, esta solução inclui um mecanismo de remediação automática que permite aos dispositivos corrigirem os problemas detetados de forma a ficarem de acordo com as políticas de segurança definidas de forma a serem aceites. Esta contribuição foi publicada na revista Sensors International Journal e encontra-se no capítulo 7 [34].

A sétima contribuição é uma solução de *network admission control* para redes 6LoWPAN que evita que dispositivos não autorizados comuniquem com outros dispositivos legítimos e com a Internet, reduzindo desta forma os ataques contra a segurança que podem ser concretizados. A solução proposta inclui a deteção e autenticação dos dispositivos que se ligam à rede, a autorização administrativa dos dispositivos e um mecanismo de descarte de mensagens provenientes/destinadas para dispositivos não autorizados. Este mecanismo recorre aos protocolos *neighbor discovery* do 6LoWPAN e ao RPL, o que permite a redução do número de mensagens de controlo. É utilizado o AES para garantir a autenticidade e a integridade dos dispositivos e a autenticidade da origem das mensagens. Foi utilizada uma rede laboratorial para validar a solução proposta. Este contributo está descrito no capitulo 8 e foi publicado na IEEE Transactions Industrial Informatics International Journal [35].

Principais conclusões e perspetivas de trabalho futuro

Esta tese aborda os desafios da integração da IoT na Internet e dos aspetos de segurança da IoT. O trabalho de pesquisa foi organizado em três partes principais. A primeira parte é dedicada ao estudo do estado da arte acerca das soluções para suportar encaminhamento e mobilidade em redes 6LoWPAN em malha e no uso das redes de sensores na monitorização ambiental e foi realizado no início desta investigação. A segunda parte é dedicada à proposta de soluções que permitam o acesso dos dispositivos ligados à Internet a dados úteis disponibilizados pela IoT, independentemente das versões do protocolo IP em uso e evitando o acesso direto aos dispositivos das LLNs. A terceira parte, versa sobre a proposta de mecanismos de segurança para mitigar os efeitos dos ataques iniciados remota e internamente e que minimizem o uso dos recursos da LLN.

A primeira parte deste trabalho de investigação é apresentado nos Capítulos 2 e 3 desta tese. O Capítulo 2 apresenta o estado da arte detalhado sobre o encaminhamento e o suporte de mobilidade nas redes 6LoWPAN em malha. Neste estado da arte, foi dada especial atenção aos protocolos 6LoWPAN e IEEE 802.15.4. As principais conclusões foram as seguintes: é viável o uso da pilha protocolar IP em todos os dispositivos, mesmo quando se trata de dispositivos com baixos recursos computacionais e com baixa capacidade de retenção de energia. O protocolo IPv6 devido à quantidade de endereços disponível e aos mecanismos de autoconfiguração é a solução consensual (i) para garantir a conectividade entre os dispositivos das LLNs mesmo quando são utilizados protocolos da camada 2 incompatíveis, (ii) para facilitar a integração da IoT à Internet e (iii) para simplificar o desenho e o desenvolvimento das aplicações. O IEEE 802.15.4 foi identificado como o protocolo da camada 2 mais adequado para endereçar os requisitos das LLN e, por conseguinte, da IoT. Esta afirmação é justificada não só por apresentar

bom desempenho sobre hardware com baixos recursos e pelo baixo consumo de energia, mas também pela diversidade de hardware comercial compatível disponível no mercado. O Capítulo 3 apresenta o estado da arte sobre as soluções disponíveis e dos projetos de monitorização ambiental baseados no uso de redes de sensores sem fios. Neste capítulo são descritas as principais vantagens da utilização de dispositivos de baixo custo e com transdutores menos precisos quando comparados com as estações meteorológica tradicionais. Neste estado da arte foram identificados os projetos mais relevantes e os seus resultados analisados, as suas conclusões foram utilizadas como base de partida para identificar os desafios ainda em aberto. De facto, a monitorização ambiental coloca muitos desafios às soluções de IoT, pois na maioria dos casos os dispositivos operam autonomamente e em condições ambientais difíceis, sem acompanhamento e sem qualquer infraestrutura de rede estável e pré-definida. Foram identificados vários problemas para os quais ainda não existe uma solução satisfatória e foi dada especial atenção aos problemas relacionados com a integração da IoT na Internet, nomeadamente, nos que estão relacionados com a preservação dos recursos das LLNs e dos aspetos de segurança relacionados com a mitigação dos efeitos dos ataques de segurança internos e externos.

A segunda parte deste trabalho é apresentada nos Capítulos 4 e 5 e versa sobre os desafios relacionados com a integração da IoT na Internet. Disponibilizar dados úteis aos dispositivos ligados à Internet, evitando a conectividade direta com os dispositivos da IoT é o problema que se pretende resolver. É proposto no Capítulo 4 uma solução ubíqua que permite aos dispositivos móveis com acesso à Internet receber as leituras mais atualizadas, bem como o histórico das medidas recolhidas por uma rede de sensores sem fios e ser também notificado quando ocorre uma condição pré-definida. Esta solução é baseada em protocolos e tecnologias abertas e usa *web services* REST, uma base de dados relacional e uma aplicação móvel para ambiente Android. O *web service* é executado no router de fronteira e é utilizado para fornecer o acesso aos dados recolhidos, preservando os recursos da LLN, e em simultâneo para disponibilizar uma interface standard que facilite o desenvolvimento das aplicações móveis independentemente do *hardware* e do *software* utilizado nos dispositivos envolvidos. Foi implementado um sistema de *push* notification para enviar alertas aos dispositivos móveis se algum dos valores medidos atingir uma condição predefinida. Esta solução foi avaliada através de um cenário laboratorial e os resultados revelaram que podem ser obtidas poupanças de energia nos dispositivos LLN e nos dispositivos móveis ligados à Internet. Para atingir a integração contínua e sem interrupções da IoT na Internet, tem de se ter em conta as duas versões do protocolo IP presentes na Internet. Assim, de forma a satisfazer este requisito a solução acima descrita foi aumentada para suportar um mecanismo de transição IPv4 para IPv6 do tipo pilha dupla (*dual stack transition mechanism*). É de realçar que nenhum dos mecanismos de transição IPv4 para IPv6 é adequado ao uso em LLNs. Assim, na nova solução, descrita no Capítulo 5, o *router* de fronteira usado para ligar a LLN à Internet suporta também o mecanismo de transição de pilha dupla de forma a satisfazer os pedidos dos clientes IPv4 e IPv6.

A terceira parte deste trabalho, apresentada nos Capítulos 6, 7 e 8, trata dos aspetos mais relevantes dos desafios de segurança da IoT. O Capítulo 6, propõe um novo mecanismo de segurança que previne os ataques de negação de serviços iniciados remotamente (i.e. por dispositivos ligados à Internet). O mecanismo proposto evita o uso de mecanismos de *firewall* que se revelam ineficientes e difíceis de gerir quando aplicados às redes da IoT. O novo mecanismo é suportado pelo router de fronteira e deixa passar apenas o tráfego proveniente da Internet e destinado à LLN, se cumprir as condições pré-definidas, tal como acontece numa *firewall*. Contudo nesta proposta, os dispositivos da LLN usam uma versão adaptada do *mecanismo neighbor address*

registration do protocolo *neighbor discovery* do 6LoWPAN para notificar o *router* de fronteira acerca das condições que devem ser utilizadas para filtrar o tráfego que lhes é destinado. Este mecanismo não requer outras mensagens para além das utilizadas para realizar o registo do endereço junto do *router* de fronteira e também não aumenta o comprimento das mensagens, porque usa campos não atribuídos das mensagens do protocolo *neighbor address registration*. Vários dos ataques à segurança, sejam internos ou externos, iniciados por dispositivos dentro da mesma rede LLN podem ser evitados se for utilizado um mecanismo de *network access control*, que restrinja o acesso à rede apenas para os dispositivos autorizados e que estão em conformidade com as políticas de segurança definidas. O Capítulo 7 propõe uma *framework* para realizar *network access control* que pode ser utilizada para dar resposta a estes objetivos. A *framework* proposta permite a identificação dos dispositivos através de endereços gerados com recurso a ferramentas criptográficas, a avaliação da conformidade da segurança dos dispositivos e a remediação através da atualização remota de *software*. Esta solução é principalmente baseada nos seguintes protocolos abertos: (i) LSEND, utilizado para a descoberta e identificação segura dos dispositivos, (ii) RPL com as mensagens protegidas pelo mecanismo criptográfico ECC e (iii) Seluge, que permite a instalação remota de software. Nesta proposta foram consideradas as sinergias entre os protocolos envolvidos. O Capítulo 8 descreve a solução de *network admission control* para autorizar quais os dispositivos que podem aceder à rede baseada numa autorização administrativa. Foram concretizadas modificações nas funções de expedição (*forwarding*) de pacotes do TinyOS e foi alterada a aplicação UDPecho do TinyOS 2.1 para realizar a autenticação dos dispositivos e a disseminação da lista de dispositivos autorizados. Foram utilizados mecanismos criptográficos de chave simétrica baseados no AES para garantir a autenticação e integridade dos dispositivos, a confidencialidade, a integridade e a atualidade dos dados. A solução foi avaliada em ambiente laboratorial e os resultados demonstraram que o acréscimo do uso de recursos provocado por este mecanismo não é significativo, principalmente porque baseia o seu funcionamento nos protocolos normalmente utilizados para garantir o funcionamento das redes 6LoWPAN. Os testes de validação demonstraram que a solução proposta funciona corretamente e que atinge os objetivos propostos.

Perspetivas de trabalho futuro

Os próximos parágrafos detalham alguns aspetos do trabalho desenvolvido que foram deixados em aberto e que por isso podem ser endereçados como trabalho futuro.

Em todos os contributos desta tese, os mecanismos propostos destinam-se a cenários que envolvem apenas um *router* de fronteira para assegurar a conectividade entre a LLN e a Internet. De forma a resolver problemas de escalabilidade e para aumentar a robustez da LLN a falhas, as redes IoT devem evoluir para topologias que envolvam múltiplos *router* de fronteira. Adaptar as soluções e os mecanismos propostos para poderem operar em ambientes com múltiplos routers de fronteira não é simples e, por conseguinte, tais adaptações devem ser alvo de estudo.

Nos mecanismos de *network access control*, tal como proposto no Capítulo 8, são dois os aspetos que foram concretizados para minimizar o consumo dos dispositivos das LLN. O primeiro passa pela utilização de uma chave criptográfica global, disponibilizada no processo de autenticação dos dispositivos. Esta abordagem minimiza o número de mensagens necessárias, no entanto torna a sua modificação menos dinâmica. O outro é o uso de mecanismos criptográficos de chave simétrica baseados no AES que são caracterizados por exigirem menos recursos computa-

cionais do que os mecanismos de chave assimétrica. Quanto ao primeiro aspeto, é necessário a proposta de um novo mecanismo que evite a necessidade de se alterar a chave global em todos os dispositivos da LLN, caso a chave global necessite de ser alterada. Relativamente ao segundo, deve ser investigada a possibilidade da utilização de mecanismos criptográficos de chave assimétrica, uma vez que tais mecanismos, facilitam a gestão das chaves criptográficas. De entre os mecanismos de chave assimétrica disponíveis os baseados em curvas elípticas são os mais promissores.

Finalmente, é de referir que a solução de *network access control* proposta continua vulnerável aos ataques contra os protocolos *neighbor discovery* do 6LoWPAN e do RPL. Razão pela qual os mecanismos de filtragem de pacotes propostos na solução de *network access control* devem também ser aplicados a todas as mensagens destes protocolos. As novas soluções devem ser investigadas cuidadosamente de forma a que se atinja o equilíbrio entre o aumento da segurança e o aumento da utilização dos recursos da LLN.

Referências

1. Da Xu, Li, Wu He, and Shancang Li. ”Internet of things in industries: A survey.” IEEE Transactions on industrial informatics10, no. 4 (2014): 2233-2243.
2. Gubbi, Jayavardhana, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. ”Internet of Things (IoT): A vision, architectural elements, and future directions.” Future generation computer systems 29, no. 7 (2013): 1645-1660.
3. Al-Fuqaha, Ala, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. ”Internet of things: A survey on enabling technologies, protocols, and applications.” IEEE Communications Surveys and Tutorials 17, no. 4 (2015): 2347-2376.
4. Zhou, Mengchu, Giancarlo Fortino, Weiming Shen, Jin Mitsugi, James Jobin, and Rahul Bhattacharyya. ”Guest editorial special section on advances and applications of Internet of Things for smart automated systems.” IEEE Transactions on Automation Science and Engineering 13, no. 3 (2016): 1225-1229.
5. Whitmore, Andrew, Anurag Agarwal, and Li Da Xu. ”The Internet of Things—A survey of topics and trends.” Information Systems Frontiers 17, no. 2 (2015): 261-274.
6. Lu, Gang, Bhaskar Krishnamachari, and Cauligi S. Raghavendra. ”Performance evaluation of the IEEE 802.15. 4 MAC for low-rate low-power wireless networks.” In Performance, Computing, and Communications, 2004 IEEE International Conference on, pp. 701-706. IEEE, 2004.
7. Bormann, Carsten, Mehmet Ersue, and A. Keranen. Terminology for constrained-node networks. No. RFC 7228. 2014.
8. Hui, Jonathan W., and David E. Culler. ”Extending IP to low-power, wireless personal area networks.” IEEE Internet Computing 12, no. 4 (2008).
9. Gungor, Vehbi C., and Gerhard P. Hancke. ”Industrial wireless sensor networks: Challenges, design principles, and technical approaches.” IEEE Transactions on industrial electronics 56, no. 10 (2009): 4258-4265.

10. Shelby, Zach, and Carsten Bormann. 6LoWPAN: The wireless embedded Internet. Vol. 43. John Wiley Sons, 2011.

11. Khan, Rafiullah, Sarmad Ullah Khan, Rifaqat Zaheer, and Shahid Khan. ”Future internet: the internet of things architecture, possible applications and key challenges.” In Frontiers of Information Technology (FIT), 2012 10th International Conference on, pp. 257-260. IEEE, 2012.

12. Aazam, Mohammad, and Eui-Nam Huh. ”Fog computing and smart gateway based communication for cloud of things.” In Future Internet of Things and Cloud (FiCloud), 2014 International Conference on, pp. 464-470. IEEE, 2014.

13. Yaqoob, Ibrar, Ejaz Ahmed, Ibrahim Abaker Targio Hashem, Abdelmuttlib Ibrahim Abdalla Ahmed, Abdullah Gani, Muhammad Imran, and Mohsen Guizani. ”Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges.” IEEE Wireless Communications 24, no. 3 (2017): 10-16.

14. Rault, Tifenn, Abdelmadjid Bouabdallah, and Yacine Challal. ”Energy efficiency in wireless sensor networks: A top-down survey.” Computer Networks 67 (2014): 104-122.

15. Shaikh, Faisal Karim, and Sherali Zeadally. ”Energy harvesting in wireless sensor networks: A comprehensive review.” Renewable and Sustainable Energy Reviews 55 (2016): 1041-1054.

16. Islam, Md Motaharul, and Eui-Nam Huh. ”Sensor proxy mobile IPv6 (SPMIPv6) - A novel scheme for mobility supported IP-WSNs.” Sensors 11, no. 2 (2011): 1865-1887.

17. Sicari, Sabrina, Alessandra Rizzardi, Luigi Alfredo Grieco, and Alberto Coen-Porisini. ”Security, privacy and trust in Internet of Things: The road ahead.” Computer Networks 76 (2015): 146-164.

18. Ziegeldorf, Jan Henrik, Oscar Garcia Morchon, and Klaus Wehrle. ”Privacy in the Internet of Things: threats and challenges.” Security and Communication Networks 7, no. 12 (2014): 2728-2742.

19. Granjal, Jorge, Edmundo Monteiro, and Jorge Sá Silva. ”Security for the internet of things: a survey of existing protocols and open research issues.” IEEE Communications Surveys and Tutorials 17, no. 3 (2015): 1294-1312.

20. Singh, Shio Kumar, M. P. Singh, and D. K. Singh. ”A survey on network security and attack defense mechanism for wireless sensor networks.” International Journal of Computer Trends and Technology 1, no. 2 (2011): 9-17.

21. Abomhara, Mohamed, and Geir M. Køien. ”Security and privacy in the Internet of Things: Current status and open issues.” In Privacy and Security in Mobile Systems (PRISMS), 2014 International Conference on, pp. 1-8. IEEE, 2014.

22. Heer, Tobias, Oscar Garcia-Morchon, René Hummen, Sye Loong Keoh, Sandeep S. Kumar, and Klaus Wehrle. ”Security Challenges in the IP-based Internet of Things.” Wireless Personal Communications 61, no. 3 (2011): 527-542.

23. Gaddour, Olfa, and Anis Koubâa. ”RPL in a nutshell: A survey.” Computer Networks 56, no. 14 (2012): 3163-3178.

24. Pautasso, Cesare, Olaf Zimmermann, and Frank Leymann. ”Restful web services vs. big’ web services: making the right architectural decision.” In Proceedings of the 17th international conference on World Wide Web, pp. 805-814. ACM, 2008.

25. Nordmark, Erik, and Robert Gilligan. Basic transition mechanisms for IPv6 hosts and routers. No. RFC 4213. 2005.

26. Newcomer, Eric. Understanding Web Services: XML, Wsdl, Soap, and UDDI. Addison-Wesley Professional, 2002.

27. Fielding, Roy T., and Richard N. Taylor. Architectural styles and the design of network-based software architectures. Doctoral dissertation: University of California, Irvine, 2000.

28. Wool, Avishai. ”Trends in firewall configuration errors: Measuring the holes in swiss cheese.” IEEE Internet Computing 14, no. 4 (2010): 58-65.

29. Oliveira, Luís ML, Amaro F. De Sousa, and Joel JPC Rodrigues. ”Routing and mobility approaches in IPv6 over LoWPAN mesh networks.” International Journal of Communication Systems 24, no. 11 (2011): 1445-1466.

30. Oliveira, Luís ML, and Joel JPC Rodrigues. ”Wireless Sensor Networks: A Survey on Environmental Monitoring.” JCM 6, no. 2 (2011): 143-151.

31. Oliveira, Luis ML, Joel JPC Rodrigues, André GF Elias, and Bruno B. Zarpelão. ”Ubiquitous monitoring solution for Wireless Sensor Networks with push notifications and end-to-end connectivity.” Mobile information systems 10, no. 1 (2014): 19-35.

32. Oliveira, Luís ML, Joel JPC Rodrigues, André GF Elias, and Guangjie Han. ”Wireless sensor networks in IPv4/IPv6 transition scenarios.” Wireless personal communications 78, no. 4 (2014): 1849-1862.

33. Oliveira, Luís ML, Joel JPC Rodrigues, Amaro F. Sousa, and Jaime Lloret. ”Denial of service mitigation approach for IPv6-enabled smart object networks.” Concurrency and Computation: Practice and Experience 25, no. 1 (2013): 129-142.

34. Oliveira, Luís ML, Joel JPC Rodrigues, Amaro F. de Sousa, and Jaime Lloret. ”A network access control framework for 6LoWPAN networks.” Sensors 13, no. 1 (2013): 1210-1230.

35. Oliveira, Luís Miguel L., Joel JPC Rodrigues, Amaro F. de Sousa, and Victor M. Denisov. ”Network Admission Control Solution for 6LoWPAN Networks Based on Symmetric Key Mechanisms.” IEEE Transactions on Industrial Informatics 12, no. 6 (2016): 2186-2195.

Abstract

The IoT means a world-wide network of interconnected objects based on standard communication protocols. An object in this context is a quotidian physical device augmented with sensing/actuating, processing, storing and communication capabilities. These objects must be able to interact with the surrounding environment where they are placed and to cooperate with neighbouring objects in order to accomplish a common objective. The IoT objects have also the capabilities of converting the sensed data into automated instructions and communicating them to other objects through the communication networks, avoiding the human intervention in several tasks. Most of IoT deployments are based on small devices with restricted computational resources and energy constraints. For this reason, initially the scientific community did not consider the use of IP protocol suite in this scenarios because there was the perception that it was too heavy to the available resources on such devices. Meanwhile, the scientific community and the industry started to rethink about the use of IP protocol suite in all IoT devices and now it is considered as the solution to provide connectivity between the IoT devices, independently of the Layer 2 protocol in use, and to connect them to the Internet. Despite the use of IP suite protocol in all devices and the amount of solutions proposed, many open issues remain unsolved in order to reach a seamless integration between the IoT and the Internet and to provide the conditions to IoT service widespread. This thesis addressed the challenges associated with the interconnectivity between the Internet and the IoT devices and with the security aspects of the IoT. In the interconnectivity between the IoT devices and the Internet the problem is how to provide valuable information to the Internet connected devices, independently of the supported IP protocol version, without being necessary accessed directly to the IoT nodes. In order to solve this problem, solutions based on Representational state transfer (REST) web services and IPv4 to IPv6 dual stack transition mechanism were proposed and evaluated. The REST web service and the transition mechanism runs only at the border router without penalizing the IoT constrained devices. The mitigation of the effects of internal and external security attacks minimizing the overhead imposed on the IoT devices is the security challenge addressed in this thesis. Three different solutions were proposed. The first is a mechanism to prevent remotely initiated transport level Denial of Service attacks that avoids the use of inefficient and hard to manage traditional firewalls. It is based on filtering at the border router the traffic received from the Internet and destined to the IoT network according to the conditions announced by each IoT device. The second is a network access security framework that can be used to control the nodes that have access to the network, based on administrative approval, and to enforce security compliance to the authorized nodes. The third is a network admission control framework that prevents IoT unauthorized nodes to communicate with IoT authorized nodes or with the Internet, which drastically reduces the number of possible security attacks. The network admission control was also exploited as a management mechanism as it can be used to manage the network size in terms of number of nodes, making the network more manageable, increasing its reliability and extending its lifetime.

Keywords

Internet of Things; Wireless Sensor Networks; 6LoWPAN; IPv6; Network Admission Control; REST-Ful web services.

Contents

List of Figures

List of Figures

Routing and mobility approaches in IPv6 over LoWPAN mesh networks

Wireless Sensor Networks: a Survey on Environmental Monitoring

Ubiquitous monitoring solution for Wireless Sensor Networks with push notifications and end-to-end connectivity

Wireless Sensor Networks in IPv4/IPv6 Transition Scenarios

Denial of service mitigation approach for IPv6-enabled smart object networks

A Network Access Control Framework for 6LoWPAN Networks

Network Admission Control Solution for 6LoWPAN Networks Based on Symmetric Key Mechanisms

List of Tables

Denial of service mitigation approach for IPv6-enabled smart object networks

Acronyms

3-DES	Triple Data Encryption Standard
6CO	6LoWPAN Context Options
6LBR	6LoWPAN Border Router
6LN	6LoWPAN Node
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks
6LR	6LoWPAN Router
ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
AFI	Accept Data From the Internet
AODV	Ad hoc On-Demand Distance Vector
API	Application Programming Interface
ARO	Address Registration Option
ASIC	Application Specific Integrated Circuit
C2DM	Cloud to Device Management
CBC-MAC	Cipher Block Chaining Message Authentication Code
CCA	Clear Channel Assessment
CGA	Cryptographically generated addresses
CN	Correspondent Node
COTS	Commercial Off-the-Shelf
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier-Sense Multiple Access with Collision Avoidance
CTR	Counter Mode
CTS	Clear to Send
DAC	Duplicate Address Confirmation
DAD	Duplicate Address Detection
DAG	Directed Acyclic Graph
DAR	Duplicate Address Request
DBMS	Database Management System
DDoS	Distributed denial of service
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol v6
DODAG	Destination Oriented DAG
DoS	Denial of Service
DSR	Dynamic Source Routing Protocol
DSSS	Direct Sequence Spread Spectrum
ECC	Elliptic Curve Cryptography
ED	Energy Detection
EUI-64	64-bit Extended Unique Identifier
FA	Foreign Agent
FCF	IEEE 802.15.4 Frame Control Field
FCHK	IEEE 802.15.4 Frame Checksum
FFD	Full-Function Devices

FIB	Forwarding Information Base
FIFO	First In First Out
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GTS	Guaranteed Time Slot
HA	Home Agent
HC	Header Compression
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMPv6	Internet Control Message Protocol version 6
ID	Identification
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet protocol
IPsec	Internet Protocol Security
IPv4	Internet protocol v4
IPv6	Internet protocol version 6
JAX-RS	Java API for RESTful Web Services
JMS	Java Message Service
JSON	JavaScript Object Notation
JTAG	Joint Test Action Group
LBR	LLN Edge Router
LLN	Low Power Lossy Network
LoWPAN	Low power Wireless Personal Area Networks
LQI	Link Quality Indicator
LSEND	Lightweight Secure Neighbour Discovery for Low-power and Lossy
MAC	Medium Access Control sub-layer protocol
MANET	Mobile Ad-Hoc Network
MCU	Microcontroller Unit
MHz	Mega Hertz
MIPv6	Mobile IPv6
MN	Mobile Node
MTU	Maximum Transmission Unit
NA	Neighbour Advertisement
ND	Neighbour Discovery
NDP	Neighbour discovery Protocol
NEMO	Network Mobility
NS	Neighbour Solicitation
OLSR	Optimised Link State Routing Protocol
OS	Operating System
PAN	Personal Area Network
PC	Personal Computer
PHP	Hypertext Preprocessor
PHY	Physical layer protocol
PMIPv6	Proxy MIPv6

RA	Router Advertisement
RAM	Random-Access Memory
RC5	Rivest Cipher version 5
REST	Representational State Transfer
RFC	Request For Comments
RFD	Reduced-Function devices
RISC	Reduced Instruction Set Computer
ROLL	Routing Over Low power and Lossy networks
RPC	Remote Procedure Call
RPL	IPv6 Routing Protocol for Low-power and lossy networks
RS	Router Solicitation
RTS	Request to Send
SDK	System Development Kit
SEND	Secure Neighbor Discovery Protocol
SIP	Session Initiation Protocol
SLLA	Source Link-layer Address
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SSH	Secure Shell
TCP	Transmission Control Protocol
TCP/IP	Transport Control Protocol/Internet Protocol
TP	Transport Protocol
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USB	Universal Serial Bus
WPAN	Wireless PAN
WSAN	Wireless Sensor and Actuator Network
WSN	Wireless Sensor Networks
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol

Chapter 1

Introdution

This thesis addresses the subject of routing and mobility on IPv6 over LoWPAN wireless mesh networks. The focus and scope of this research are further described in this chapter, together with the problem definition and objectives, the thesis statement, the main contributions and the thesis organization.

1.1 Motivation

With the rapid evolution of Micro Electro Mechanical Systems (MEMS) and communication technologies, Internet of Things (IoT) has gained broad-spectrum attention due to its high potential impact on several aspects of everyday-life [1][2]. IoT is being widely applied in various areas such as military systems, environment monitoring, energy management, healthcare systems, building automation and transportation, changing business models. Semantically, IoT means a world-wide network of interconnected uniquely addressable objects based on standard communication protocols [3]. In this context, an object is a quotidian physical object augmented with sensing/actuating, processing, storing and communication capabilities. These objects must be able to interact with the surrounding environment where they are placed and to cooperate with neighbouring objects in order to accomplish a common objective. The IoT objects have also the capabilities of converting the sensed data into automated instructions and communicating them to other objects with actuating capabilities through the communication networks. These objects will in turn actuate other objects, avoiding the human intervention in several tasks. The origin of the term IoT and its original meaning are not clearly known. The IoT concept was probably devised by Kevin Ashton and was initially associated to the radio frequency identification (RFID) technology [4]. Sensors and actuators build the foundation of IoT where RFID was replaced by IEEE802.15.4 as base technology and the Sensor Wireless Networks (WSN) concept was appeared [5][6]. Recently, many other Layer 2 technologies are also being considered to support IoT, most of them standard, reinforcing the above described IoT vision. Despite the inclusion of technologies which are not aware of constraints such as energy consumption or amount of required hardware resources, most of IoT deployments are based on small devices with restricted computational resources and energy constraints, forming a low power and lossy network (LLN) [7]. At the first stages, the scientific community did not consider the use of IP protocol suite in LLNs, because there was the perception that it was too heavy to the available resources on such devices. Meanwhile, the scientific community and the industry started to rethink many misconceptions about the use of IP protocol suite in all LLN [8]. First, the IP protocol can be used to avoid interconnectivity problems related with incompatible low layer protocols, such as WirelessHart and ZigBee, perpetuating the end to end Internet nature [9]. Second, it provides an open and royalty free application developing process, where already available tools for commissioning, configuring, managing and debugging can be used or adapted. Finally, an all IP solution avoids the use of complex and hard to manage proxies and gateways necessary

to interconnect IP incompatible LLN nodes to the Internet [8]. Although the IPv6 protocol has enough address space to connect all LLN devices, it was not designed to be used in low-power and resource constrained nodes. To address these constraints, a 6LoWPAN Adaptation Layer was defined to be used between the data link and network layers and providing new routing approaches, header compression, fragmentation support above 1280 bytes and support to auto configuration and neighbour discovery mechanisms more adapted to LLNs characteristics [10].

One of the first proposed IoT architectures considers three layers: perception, network and application [11]. The perception layer runs on the sensor devices and is responsible for sensing and gathering the information from the surrounding environment. The network layer is responsible for the communications among sensor devices and with all other network elements like servers, gateways, etc. Finally, the application layer is responsible for extracting valuable information from the voluminous data retrieved by the LLN devices (employing intelligent computing technologies such as data mining and cloud computing) and providing an interface between users and the IoT system.

Although the three-layer architecture reflects the main idea of IoT, it is not sufficiently detailed to describe all the relevant aspects, the reason why more layered architectures were meanwhile proposed in the literature. One of these more recent proposals is the five-layer architecture, which includes the processing and business layers [12]. The five layers are perception, transport, processing, application, and business layers. The perception and application layers are the same as in the three-layer architecture. The transport layer is responsible for transferring the data (collected by the LLN devices) between the perception and the processing layers through a network infrastructure such as IEEE802.15.4, 5G, WiFi and Ethernet. The processing layer is responsible to store, analyse and process data that comes from the transport layer. This layer, also known as middleware, performs information processing and takes decisions automatically based on results. It then passes the results to the application layer. Technologies such as databases, cloud computing, and big data processing modules are used in this layer. The business layer is responsible for managing the complete IoT system including applications, business models and users' privacy. Despite the IoT being considered a new paradigm, it cannot be considered a greenfield technology because it uses some paradigms and protocols used in the Internet. However, existing networking architectures cannot provide smooth integration to the huge amount of devices since to reach the IoT paradigm different types of networks must be involved which can cause serious problems [13].

There is a huge amount of work already done but many open issues remain unsolved in order to reach a seamless integration between the IoT and the Internet and to provide the conditions to IoT service widespread. These open issues and requirements can be classified according to: interoperability, flexibility, scalability, energy efficiency, mobility management and security [13].

Three main types of interoperability open issues can be identified [13]: technical, semantic and pragmatic. The technical challenges are related with device capabilities and supported protocols. The semantic challenges are concerned with the capability of the IoT components to interpret and process the exchanged data. The pragmatic challenges are concerned with the capability of the system components to analyse the parties' intentions. Flexibility is required in order to provide service provisioning to the diversity of IoT applications according to their requirements. The scalability is a key requirement to the management systems since the IoT architectures need to scale up to accommodate a huge number of smart objects.

Most of the devices used in the perception layer are characterized by the limited processing, storage and energy resources and, therefore, resource and message intensive services and protocols should be avoided [3]. High energy efficiency is achieved with low duty cycle operations in order that the objects' radio interface remains in a sleeping mode most of the time [14]. Nevertheless, such approach must be used with care due to its impact on the performance of the whole IoT solution. Solutions to harvest energy from the surrounding environment might be preferable over low duty cycle approaches in order to minimize the impact of energy efficient solutions [15]. The support of mobility and its management is crucial to many LLN scenarios and both node and network mobility must be considered. While physical mobility is easy to understand because the node changes its position, there are also other subtler causes of topology changes in LLN due to wireless interference or fading for example [16].

Security is possibly the main issue on reaching IoT service widespread [17]. The security is fundamental to ensure data confidentiality, integrity and privacy during data transmission and storage and also to avoid service disruption. Therefore, the security guarantee is crucial as an enabling factor of most IoT applications. As an open system with Internet full integration, LLN exhibits a larger number of vulnerabilities and unique new security challenges, which make them even more vulnerable to security attacks than unconstrained networks (i.e. a network that does not fulfil the definition of constrained network defined in the Terminology for Constrained-Node Networks RFC 7228) [18]. Consequently, the application of traditional security techniques to LLN is not straightforward. Most of the new security challenges are related with the absence of an organized communication infrastructure and with the resource constrained nature of LLN devices where security mechanisms must be used with care due to the resources they need. Additionally, any message overhead caused by security mechanisms must also be minimized whenever possible. Different security mechanisms have been proposed, some of them defined to address particular well-known attacks. In order to make the IoT networks and services more manageable and secure, new security mechanisms and approaches must be considered [19].

Different criteria are being used to classify security attacks. First, they can be classified as external or internal, according to the ownership of the resources used to execute the attacks [20]. In external attacks, the attacker uses its own resources to perform the attack. In internal attacks, the attacker first compromises one legitimate node (usually, by the injection of malicious code or by accessing important data stored on them), and then, uses its resources to perform the attacks. Note that internal attacks are much harder to detect because the compromised node appears as legitimate to the other nodes and, therefore, has their trust. Second, security attacks can also be classified as passive or active based on the modification of the regular data streaming [21]. Passive attacks are based on message gathering without modification, while active attacks involve modification on the legitimate data stream and/or false data injection. Finally, security attacks can also be clustered in three main groups according to their security requirements: attacks against secrecy and authenticity, attacks against network availability, and stealthy attacks against service integrity. While eavesdropping, packet replay, tampering, and spoofing are examples of the first group, Denial of Service (DoS) is an example of an attack against availability [22]. DoS can be defined as any event that reduces or eliminates a network's capacity to perform its expected function. DoS is a common attack type because it may use only regular equipment and does not require high knowledge skills. Note that DoS is more damaging in LLNs due to the resource differences between the regular Internet devices and the IoT devices and therefore this is still an open problem to the network security community. First, the IoT devices cannot support the computational overhead necessary to implement many of the

typical defensive strategies. Second, small traffic rates are enough to drain IoT devices energy that makes the network inoperable.

1.2 Problem Definition and Research Objectives

The research objectives of this work have evolved during its realization. Initially, the research problem consisted in the proposal and validation of solutions addressing simultaneously the routing and mobility requirements of IPv6 over LLN wireless mesh networks. At the beginning, these were the main research issues to be solved in the IoT framework and security was still too soon to be addressed due to its dependability on the solutions for these previous issues. However, the RPL routing protocol, which was proposed in 2012, has addresses the most relevant routing problems in LLN while supporting routing paths diversity and micro mobility [23]. Since RPL has become a standard, the focus of the research took into account these developments and evolved to include research challenges associated with the interconnectivity between the Internet and IoT devices and with the security aspects of the IoT.

In the interoperability between Internet and IoT devices, the problem is how to let the Internet connected devices retrieve useful data from LLN nodes without accessing directly to them. The research objective is to develop and validate solutions that (i) minimize the amount of communications required in the LLN (extending in this way the lifetime of the resource constrained sensor devices), (ii) use well established Internet mechanisms and (iii) deal with both IPv4 and IPv6 Internet connected devices. To achieve these research objectives, special attention was dedicated to Web service technologies and IPv4 to IPv6 transition mechanisms as potential solutions for connectivity issues on the application and networks layers, respectively [24][25].

Web service technologies have become an industry de facto standard for service distribution between remote and heterogeneous systems [24]. Web services provide well-defined interfaces for distributed systems, which are independent of the hardware, operating system and programming languages used to develop the server and client side. Interoperability is mainly provided by Extensible Markup Language (XML) based open standards. The Simple Object Access Protocol (SOAP) messages are defined in XML, which are text-based and self-described, and are used to transfer information between remote located services without conversion issues [26]. The web services can still be used to provide access to the IoT services from mobile devices, such as smart phones. However, applying the current Web Service architecture model to mobile devices may result in unacceptable performance overhead, due to the encoding and decoding of verbose XML based SOAP messages. The RESTful web services attempts to emulate HTTP and similar protocols by constraining the interface to well-known and standard operations, such as GET, PUT, POST and DELETE. So, data and services are considered resources and are accessed using web links, i.e. Uniform Resource Identifiers (URIs) [27]. The REST is a client server architecture and was designed to use a stateless communication protocol, such as HTTP. Because this is a lightweight architecture, it performs well to provide services directly to the mobile devices or through cloud based services. The web services should be supported by the edge router used to connect the IoT network to the Internet since this device is usually main-powered and has more computational resources then regular IoT devices.

As already mentioned, the IPv6 is the most suitable network layer protocol to be used on IoT networks. However, despite most operating systems already support natively both IP protocol

versions (IPv4 and IPv6), there are still many deployed access networks providing only IPv4 support. Consequently, providing support to IPv4 to IPv6 transition mechanisms in the IoT edge router can be the drive to accelerate the IoT service based adoption without increasing the overhead in the IoT network.

The research problem related to the security issues on LLN networks addressed in this thesis is how to mitigate the effects of internal and external security attacks initiated inside and outside of the LLN. The research objective is to formulate and validate solutions suitable for use in multi-hop LLN networks that (i) provide node presence detection and authentication; (ii) prevent unauthorized nodes from using the network to communicate both with legitimate nodes and with the Internet; (iii) ensure that the LLN nodes are compliant with the established security posture; and (iv) filter on the edge router the messages received from the Internet and destined to the LLN according to predefined rules. In order to fulfil these research objectives, firewalls and network admission control solutions were carefully studied and analysed.

Traditional firewalls control access to resources by filtering network traffic and only allowing access that is specified by the security policy [28]. The network traffic is filtered according to a predefined set of rules statically set by the network administrator. The firewalls are not efficient when used to protect LLN networks due to dynamic nature of the LLN and to the heterogeneity of devices and their functions. Instead of a firewall based solution, the aim is to propose another filtering mechanism that avoids the disadvantages of firewalls. This new filtering mechanism should be supported only by the edge routers and the filtering rules should be dynamically built according to the information sent by the LLN internal nodes. Like in the traditional firewalls, unsupported traffic at the boundary must be filtered and supported traffic must be rate-shaped to ensure no more requests than the imposed limits from the Internet are forwarded to the LLN nodes.

Self-organization and self-configuration are key characteristics of LLNs because they minimize the network configuration efforts and simultaneously increase the network robustness, but they can also be exploited to perform security attacks [19]. A network admission control is a twofold solution. First, it can be used to manage the size of the IoT network in terms of number of nodes, making the network more manageable, increasing its reliability and extending its lifetime. Second, it can also be used for security since, if a malicious node is prevented from using the network, it cannot communicate with other network elements and, therefore, the number of possible internal and external security attacks is drastically reduced. A network admission control solution should comprise mechanisms to detect node presence, to perform node authentication and integrity check and also to ensure message integrity, source authentication and data freshness.

1.3 Thesis Statement

This thesis proposes mechanisms and strategies to promote the integration between IoT networks and Internet and to minimize the impact of internal and external initiated security attacks. Specifically, the thesis statement is:

The IoT vision considers the seamless connection of many LLNs to the Internet with each LLN composed by a potentially very large number of highly resource and energy constrained devices that generates huge amounts of heterogeneous data. The realization of the IoT vision is no

longer compatible with the end to end paradigm and mechanisms, traditionally located on end devices, must now be moved to intermediate network elements. Border routers are the ideal network elements to host such mechanisms since they are typically main-powered and less restricted resource devices. Using this approach, it is possible to reach some key IoT goals. One is to provide content access to Internet attached devices without direct connectivity from them to IoT devices, avoiding extra traffic exchange on LLNs. Another is to effectively mitigate internal and external initiated security attacks. External initiated attacks can be mitigated avoiding the inefficiency of traditional firewall approaches. Internal initiated attacks can be mitigated by network admission control without penalising too much the IoT devices and, additionally, imposing policy compliance and administrative control.

To support this thesis statement, the following research approach was adopted:

Considering the relevance of the LLN networks to the success of the IoT, specifically the wireless sensor networks, the standard protocols defined to support layer two and layer three were carefully investigated. Through these studies, the achievements in this area were identified, as well their limitations and open issues. A review of the wireless sensor network solutions applied to environmental monitoring scenarios was conducted in order to assess how this technology performs in harsh and unattended environments. Through these studies, state-of-art contributions were identified and used as a starting point for the proposal of novel approaches to promote seamless Internet integration and protection against internal and external security attacks.

Regarding the integration between the IoT networks and the Internet and based on the challenges and limitations identified in the previous studies on the proposed standard protocols, mainly for layer 2 and 3, a laboratory testbed using devices compliant with IEEE 802.15.4 running TinyOS and Contiki operating systems was implemented. In this testbed, the edge router only forwards packets between the two networks and therefore end to end connectivity was reached. This was the starting point to the next testbeds used to evaluate the proposed mechanisms.

Then, an approach based on RESTfull webservices, running on the border router, was considered (in this case, end-to-end connectivity is not allowed). The RESTfull web services are used to provide efficient data access and a standard application interface to make applications development easier and independent of the hardware and software used in the IoT devices. On this approach, three different ways for Internet connected mobile phones to retrieve data collected by the sensors were considered. In the first one, users request up-to-date and historical data to a RESTful web service that retrieves sensor data stored in a relational database. In the second one, users request real-time data directly to the border router and the border router, acting as a proxy, retrieves the requested data from the sensor device. Finally, in the third way, a push service notifies users when sensor data overcomes a given threshold. A laboratory testbed was used to assess and validate this solution.

IoT seamless integration into the Internet must take into account both versions of the IP protocol. In fact, despite the number of IPv6 compliant devices, for a considerable number of end devices only IPv4 Internet access is available. The RESTfull webservices may also be used to enable the interaction between the IoT and all Internet connected devices, whatever the the supported IP version is. To reach this goal, a mechanism that combines RESTfull Web services with IPv4 to IPv6 transition mechanisms was proposed and evaluated aiming, in this way, to increase the IoT services reachability dissemination. The transition mechanisms and the RESTfull Web services were implemented in the border router saving the IoT device resources'.

Connecting the IoT devices to the Internet exposes them to a myriad of security attacks even if end to end connectivity is not supported or constrained. The most relevant security threats were identified and studied and also the available security mechanism used to mitigate their effects. DoS attack effects in IoT networks were carefully analysed because they are among the most common types of security attacks. A mechanism located in the edge router to filter unwanted traffic was investigated. The evaluation of firewall mechanisms was the starting point but none of them is suitable for this purpose due to the following reasons. First, there is a diversity of devices inside of the same IoT network with different purposes and, therefore, too many different rules are required. Second, IoT networks are very dynamic, hence, ensuring the correct rules in place at each time instant is very challenging. Consequently, a new mechanism to filter unwanted traffic originated in the Internet and destined to the IoT nodes was investigated and designed where the IoT devices notify the edge router on the conditions used to filter the Internet received traffic. In this mechanism, IoT nodes use 6LoWPAN address registration messages to declare willingness to accept data from the Internet and if so, the data type and limiting rate.

Self-organization and self-configuration are key characteristics of IoT networks because they minimize the network configuration efforts and simultaneously increase the network robustness but they can also be exploited by neighbour devices to perform security attacks. By appropriate network admission control, a malicious node can be prevented from communicating with the other network devices, and, in this way, the number of possible security attacks is drastically reduced. Network admission control was also evaluated as a management mechanism as it can be used to manage the size of the network in terms of number of nodes, making the network more manageable, increasing its reliability and extending its lifetime. Based on this research, a new network admission control mechanism was proposed, based on the following nuclear functions: node provisioning, node presence detection, node authentication and authorization, propagation of the authorized node list, and data filtering. The use of symmetric key cryptography algorithms and the reuse of standard protocols required for normal operation, such as 6LoWPAN neighbour discovery was taken into account in order to reduce the overhead imposed by this mechanism. A laboratory testbed based on well-known operative systems and hardware was used to validate the proposed solution.

1.4 Main Contributions

This section briefly describes the main scientific contributions resulting from the research work presented in this thesis.

The first and second contributions of this thesis are a set of two surveys conducted on the initial stages of this work. The first contribution is a state-of-art survey on available solutions proposed to support routing and mobility over 6LoWPAN mesh networks. One of the main conclusions of this contribution is that 6LoWPAN should be considered as the convergence solution to connect different layer two constrained networks while enabling Internet connectivity. This survey was published on International Journal of Communication Systems [29]. The second contribution is a comprehensive review on the wireless sensor network solutions available to support environmental monitoring applications. This contribution identifies the challenges needed to be addressed in order to build a monitoring solution based on wireless sensor networks. This survey

was published on Journal of Communications [30].

The third contribution is a solution for WSN monitoring based on a REST Web Service. Real-time data sensed by WSN nodes is sent directly to a smartphone or stored in a database. The mobile application uses a well-defined REST Web Service to retrieve sensed data, avoiding the end to end connectivity between the Internet connected devices and the sensor nodes. A push notification system was introduced in order to alert mobile users when a sensor parameter reaches a given threshold. The proposed architecture and the mobile application were evaluated and validated using a laboratory testbed and are ready for use. This contribution was published in the form of a journal paper published in Mobile Information Systems international journal [31].

The fourth contribution is a solution that combines IPv4 to IPv6 transition mechanisms with REST Web Services in order to enable the interaction between mobile applications hosted in IPv4 or IPv6 compliant hosts and 6LoWPAN enabled networks. The Web Service and the IPv4 to IPv6 transition mechanisms run in the border router in order to save the resources on WSN nodes. This contribution was published as a paper published in Wireless Personal Communications international journal [32].

The fifth contribution proposes a new mechanism to avoid DoS attacks initiated from the Internet and destined to a 6LoWPAN enabled network. An adapted version of 6LoWPAN neighbour discovery messages is proposed and used by the constrained nodes to notify the border router with the type and the rate of traffic that should be received from the Internet nodes. This proposal is available in a paper published in Concurrency Computation Practice and Experience international journal [33].

The sixth contribution is the proposal of a network admission security framework that can be used both to control the nodes that have access to the network, based on administrative approval, and to enforce security compliance to the authorized nodes. In the proposal, the framework (i) controls, at the network access level, which nodes can be attached to the network and (ii) enforces the nodes' security compliance. The framework can be used both as a management tool and as a security mechanism. The proposed framework makes use of LSEND protocol (for secure neighbor discovery and key pairwise generation), RPL (for datagram routing), and Seluge (for security compliant code dissemination). Unlike previous proposals for access control mechanisms, this proposal includes an automatic remediation mechanism that enables nodes to become security compliant, if necessary, in order to have their access granted. This contribution is presented as a paper published in the Sensors international journal [34].

The seventh contribution is a network access control solution for 6LoWPAN networks that prevents unauthorized nodes from using the network to communicate both with the legitimate nodes and with the Internet, reducing in this way the security attacks that can be performed. The proposed solution includes node presence detection and authentication, administrative node authorization, and data filtering to discard frames from/to unauthorized nodes. It uses the standard 6LoWPAN neighbour discovery and RPL protocols, minimizing the number of additional required control messages. It includes cryptographic mechanisms, based on the AES symmetric key algorithm, to guarantee node authenticity and integrity, source authenticity, and data freshness to the mechanism control data frames. A laboratory testbed was used to validating the proposed network admission control solution. This solution was published in IEEE Transactions on Industrial Informatics journal [35].

1.5 Thesis Organization

This thesis is organized in nine chapters. Besides the current chapter, the Introduction, and the chapter with the conclusions and future work (Chapter 9), all other chapters are composed by papers published in international journals. This document is organized as follows.

Chapter 2 presents a survey entitled as "Routing and mobility approaches in IPv6 over LoWPAN mesh networks". This paper states the 6LoWPAN as the convergence solution to connect the different physical and link layer protocols and to connect the LLN devices to the Internet. It also identifies the mesh routing and mobility support to realize the IoT vision.

Chapter 3 presents a comprehensive review of the available solutions and projects on environmental monitoring based on wireless sensor networks technology and is entitled as "Wireless Sensor Networks: a Survey on Environmental Monitoring". It also states the most relevant challenges and open issues related to the use of wireless sensor networks on environmental monitoring.

Chapter 4, entitled as "Ubiquitous monitoring solution for Wireless Sensor Networks with push notifications and end-to-end connectivity", proposes a solution based on REST web services to provide IoT services to mobile applications. This solution also proves it is possible to provide valuable information to the Internet connected devices, such as smartphones, while avoiding direct access to the LLN nodes. The proposed solution also includes a push notification system to alert mobile users when a sensor parameter overcomes a predefined threshold.

Chapter 5, entitled as "Wireless sensor networks in IPv4/IPv6 transition scenarios", presents a solution that uses IPv4 to IPv6 transition mechanisms located in the border router to provide IoT services to IPv4 only compliant smartphones.

Chapter 6 presents a security mechanism to protect 6LoWPAN compliant networks against DoS attacks sourced at the Internet and destined to the LLN network and it is entitled as "Denial of service mitigation approach for IPv6-enabled smart object networks".

Chapter 7, entitled as "A network access control framework for 6LoWPAN networks", proposes a framework that can be used to control the nodes that have access to the network, based on administrative approval, and to enforce security compliance to the authorized nodes.

Chapter 8 presents a network admission control solution for 6LoWPAN networks that prevents unauthorized nodes from using the network to communicate both with the legitimate nodes and with the Internet, reducing in this way the security attacks that can be performed. This chapter is entitled as "Network Admission Control Solution for 6LoWPAN Networks Based on Symmetric Key Mechanisms".

Finally, Chapter 9 concludes the thesis, summarizing the main research findings and suggesting future research directions.

In Appendix, three contributions presented in international conferences are presented and described. In Appendix A, a demonstrator for power energy monitoring and actuating system and it was developed for home environments is presented. IPv6 protocol was used to provide connectivity between IEEE 802.15.4 and Power Line sensor devices.

In Appendix B, a solution to monitor light aircraft flight parameters and gliders pilot's physiologic parameters is presented. The proposed solution does not interfere with pilot's agility, is simple

to install, configure and operate. The solution is based on IPv6 sensor compatible with IEEE 802.15.4 layer 2 protocol.

Appendix C presents presents a testbed assess and validate end to end solutions to provide connectivity between 6LoWPAN nodes and IPv6 internet connected nodes. This testbed was constructed in the initial phase of this research work and was used evaluate and demonstrate several of mechanism and solutions proposed in this thesis.

1.6 References

1. Da Xu, Li, Wu He, and Shancang Li. ”Internet of things in industries: A survey.” IEEE Transactions on industrial informatics10, no. 4 (2014): 2233-2243.
2. Gubbi, Jayavardhana, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. ”Internet of Things (IoT): A vision, architectural elements, and future directions.” Future generation computer systems 29, no. 7 (2013): 1645-1660.
3. Al-Fuqaha, Ala, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. ”Internet of things: A survey on enabling technologies, protocols, and applications.” IEEE Communications Surveys and Tutorials 17, no. 4 (2015): 2347-2376.
4. Zhou, Mengchu, Giancarlo Fortino, Weiming Shen, Jin Mitsugi, James Jobin, and Rahul Bhattacharyya. ”Guest editorial special section on advances and applications of Internet of Things for smart automated systems.” IEEE Transactions on Automation Science and Engineering 13, no. 3 (2016): 1225-1229.
5. Whitmore, Andrew, Anurag Agarwal, and Li Da Xu. ”The Internet of Things—A survey of topics and trends.” Information Systems Frontiers 17, no. 2 (2015): 261-274.
6. Lu, Gang, Bhaskar Krishnamachari, and Cauligi S. Raghavendra. ”Performance evaluation of the IEEE 802.15. 4 MAC for low-rate low-power wireless networks.” In Performance, Computing, and Communications, 2004 IEEE International Conference on, pp. 701-706. IEEE, 2004.
7. Bormann, Carsten, Mehmet Ersue, and A. Keranen. Terminology for constrained-node networks. No. RFC 7228. 2014.
8. Hui, Jonathan W., and David E. Culler. ”Extending IP to low-power, wireless personal area networks.” IEEE Internet Computing 12, no. 4 (2008).
9. Gungor, Vehbi C., and Gerhard P. Hancke. ”Industrial wireless sensor networks: Challenges, design principles, and technical approaches.” IEEE Transactions on industrial electronics 56, no. 10 (2009): 4258-4265.
10. Shelby, Zach, and Carsten Bormann. 6LoWPAN: The wireless embedded Internet. Vol. 43. John Wiley Sons, 2011.
11. Khan, Rafiullah, Sarmad Ullah Khan, Rifaqat Zaheer, and Shahid Khan. ”Future internet: the internet of things architecture, possible applications and key challenges.” In Frontiers of Information Technology (FIT), 2012 10th International Conference on, pp. 257-260. IEEE, 2012.

12. Aazam, Mohammad, and Eui-Nam Huh. ”Fog computing and smart gateway based communication for cloud of things.” In Future Internet of Things and Cloud (FiCloud), 2014 International Conference on, pp. 464-470. IEEE, 2014.

13. Yaqoob, Ibrar, Ejaz Ahmed, Ibrahim Abaker Targio Hashem, Abdelmuttlib Ibrahim Abdalla Ahmed, Abdullah Gani, Muhammad Imran, and Mohsen Guizani. ”Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges.” IEEE Wireless Communications 24, no. 3 (2017): 10-16.

14. Rault, Tifenn, Abdelmadjid Bouabdallah, and Yacine Challal. ”Energy efficiency in wireless sensor networks: A top-down survey.” Computer Networks 67 (2014): 104-122.

15. Shaikh, Faisal Karim, and Sherali Zeadally. ”Energy harvesting in wireless sensor networks: A comprehensive review.” Renewable and Sustainable Energy Reviews 55 (2016): 1041-1054.

16. Islam, Md Motaharul, and Eui-Nam Huh. ”Sensor proxy mobile IPv6 (SPMIPv6) - A novel scheme for mobility supported IP-WSNs.” Sensors 11, no. 2 (2011): 1865-1887.

17. Sicari, Sabrina, Alessandra Rizzardi, Luigi Alfredo Grieco, and Alberto Coen-Porisini. ”Security, privacy and trust in Internet of Things: The road ahead.” Computer Networks 76 (2015): 146-164.

18. Ziegeldorf, Jan Henrik, Oscar Garcia Morchon, and Klaus Wehrle. ”Privacy in the Internet of Things: threats and challenges.” Security and Communication Networks 7, no. 12 (2014): 2728-2742.

19. Granjal, Jorge, Edmundo Monteiro, and Jorge Sá Silva. ”Security for the internet of things: a survey of existing protocols and open research issues.” IEEE Communications Surveys and Tutorials 17, no. 3 (2015): 1294-1312.

20. Singh, Shio Kumar, M. P. Singh, and D. K. Singh. ”A survey on network security and attack defense mechanism for wireless sensor networks.” International Journal of Computer Trends and Technology 1, no. 2 (2011): 9-17.

21. Abomhara, Mohamed, and Geir M. Køien. ”Security and privacy in the Internet of Things: Current status and open issues.” In Privacy and Security in Mobile Systems (PRISMS), 2014 International Conference on, pp. 1-8. IEEE, 2014.

22. Heer, Tobias, Oscar Garcia-Morchon, René Hummen, Sye Loong Keoh, Sandeep S. Kumar, and Klaus Wehrle. ”Security Challenges in the IP-based Internet of Things.” Wireless Personal Communications 61, no. 3 (2011): 527-542.

23. Gaddour, Olfa, and Anis Koubâa. ”RPL in a nutshell: A survey.” Computer Networks 56, no. 14 (2012): 3163-3178.

24. Pautasso, Cesare, Olaf Zimmermann, and Frank Leymann. ”Restful web services vs. big’ web services: making the right architectural decision.” In Proceedings of the 17th international conference on World Wide Web, pp. 805-814. ACM, 2008.

25. Nordmark, Erik, and Robert Gilligan. Basic transition mechanisms for IPv6 hosts and routers. No. RFC 4213. 2005.

26. Newcomer, Eric. Understanding Web Services: XML, Wsdl, Soap, and UDDI. Addison-Wesley Professional, 2002.

27. Fielding, Roy T., and Richard N. Taylor. Architectural styles and the design of network-based software architectures. Doctoral dissertation: University of California, Irvine, 2000.

28. Wool, Avishai. ”Trends in firewall configuration errors: Measuring the holes in swiss cheese.” IEEE Internet Computing 14, no. 4 (2010): 58-65.

29. Oliveira, Luís ML, Amaro F. De Sousa, and Joel JPC Rodrigues. ”Routing and mobility approaches in IPv6 over LoWPAN mesh networks.” International Journal of Communication Systems 24, no. 11 (2011): 1445-1466.

30. Oliveira, Luís ML, and Joel JPC Rodrigues. ”Wireless Sensor Networks: A Survey on Environmental Monitoring.” JCM 6, no. 2 (2011): 143-151.

31. Oliveira, Luis ML, Joel JPC Rodrigues, André GF Elias, and Bruno B. Zarpelão. ”Ubiquitous monitoring solution for Wireless Sensor Networks with push notifications and end-to-end connectivity.” Mobile information systems 10, no. 1 (2014): 19-35.

32. Oliveira, Luís ML, Joel JPC Rodrigues, André GF Elias, and Guangjie Han. ”Wireless sensor networks in IPv4/IPv6 transition scenarios.” Wireless personal communications 78, no. 4 (2014): 1849-1862.

33. Oliveira, Luís ML, Joel JPC Rodrigues, Amaro F. Sousa, and Jaime Lloret. ”Denial of service mitigation approach for IPv6-enabled smart object networks.” Concurrency and Computation: Practice and Experience 25, no. 1 (2013): 129-142.

34. Oliveira, Luís ML, Joel JPC Rodrigues, Amaro F. de Sousa, and Jaime Lloret. ”A network access control framework for 6LoWPAN networks.” Sensors 13, no. 1 (2013): 1210-1230.

35. Oliveira, Luís Miguel L., Joel JPC Rodrigues, Amaro F. de Sousa, and Victor M. Denisov. ”Network Admission Control Solution for 6LoWPAN Networks Based on Symmetric Key Mechanisms.” IEEE Transactions on Industrial Informatics 12, no. 6 (2016): 2186-2195.

Chapter 2

Routing and mobility approaches in IPv6 over LoWPAN mesh networks

This chapter consists of the following paper:

Routing and mobility approaches in IPv6 over LoWPAN mesh networks

L. Oliveira, A. de Sousa and J. Rodrigues
International Journal of Communication Systems, vol. 24, no. 11, pp. 1445-1466, 2011.

DOI: 10.1002/dac.1228

According to Journal Citation Reports published by Thomson Reuters, this journal scored ISI journal performance metrics as follows:

ISI Impact factor (2011): 0.406

Article Influence Score (2011): 0.164

Journal Ranking (2011): 199/245 (Engineering Electrical and Electronic)

Journal Ranking (2011): 63/79 (Telecommunications)

INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2011; **24**:1445–1466
Published online 1 February 2011 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/dac.1228

Routing and mobility approaches in IPv6 over LoWPAN mesh networks

Luís M. L. Oliveira[1, 2, 3], Amaro F. de Sousa[1, 4] and Joel J. P. C. Rodrigues[1, 2, *, †]

[1]*Instituto de Telecomunicações, Portugal*
[2]*Department of Informatics, University of Beira Interior, Covilhã, Portugal*
[3]*Polytechnic Institute of Tomar, Tomar, Portugal*
[4]*Department of Electronics, Telecommunications and Informatics, University of Aveiro, Aveiro, Portugal*

SUMMARY

It is foreseeable that any object in the near future will have an Internet connection—this is the Internet of Things vision. All these objects will be able to exchange and process information, most of them characterized by small size, power constrained, small computing and storage resources. In fact, connecting embedded low-power devices to the Internet is considered the biggest challenge and opportunity for the Internet. There is a strong trend of convergence towards an Internet-based solution and the 6LoWPAN may be the convergence solution to achieve the Internet of Things vision. Wireless mesh networks have attracted the interest of the scientific community in recent years. One of the key characteristics of wireless mesh networks is the ability to self-organize and self-configure. Mesh networking and mobility support are considered crucial to the Internet of Things success. This paper surveys the available solutions proposed to support routing and mobility over 6LoWPAN mesh networks. Copyright © 2011 John Wiley & Sons, Ltd.

Received 18 May 2010; Revised 11 September 2010; Accepted 13 November 2010

KEY WORDS: wireless sensor networks; low-power personal area networks; mesh networks; routing protocols; Mobile IP

1. INTRODUCTION

Low-power wireless personal area networks (LoWPAN) consist of small size sensing nodes compliant with standard IEEE 802.15.4 [1] for wireless communications support. The LoWPAN nodes are characterized by small size, power constrained, small computing and storage resources and reduced radio ranges and throughput. Wireless Sensor Network (WSN) is a subtype of LoWPAN, where the devices can interact with their environment by sensing and/or controlling some physical parameters. In most applications, these nodes have to collaborate to fulfill a common task [2, 3]. The low size, the low cost, and the wireless communication and sensing capabilities make these devices appropriate for monitoring purposes.

A WSN network may be composed by hundreds, or maybe thousands, of *ad hoc* devices working together to accomplish a common task. There is a tendency to implement these small devices in several quotidian objects, realizing a vision of ambient networks where many different devices will collect and process information from many different sources to both control physical processes and

*Correspondence to: Joel J. P. C. Rodrigues, Department of Informatics, University of Beira Interior, Covilhã, Portugal.
†E-mail: joeljr@ieee.org

interact with human users [2]. Self-organizing, self-optimizing and fault-tolerance are the main characteristics of such networks [3].

The standard IEEE 802.15.4 release, proposed in 2003, represented a millstone because it was the first standard for LoWPAN. Several solutions have been specified using IEEE 802.15.4 as link layer technology, some of them proprietary, such as ZigBee [4] and WirelessHART [5]. The ZigBee was created by ZigBee alliance [6] and defines the network, security and application layers. The ZigBee alliance also publishes application profiles that allow multiple vendors to create interoperable products. However, the published application profiles only solve a limited set of applications for wireless-embedded networking. Moreover, the ZigBee is not compatible with IP protocol and, therefore, complex gateway systems are required to connect the ZigBee networks to the Internet. Such gateways are hard to manage because updates are required whenever new functionalities are introduced. The WirelessHART is an open-standard wireless networking technology proposed by HART Communication Foundation [5] and used mainly in industrial environments. Like ZigBee, WirelessHART is a stand-alone standard and requires a gateway device to support communications with other networks with the same previous mentioned limitations.

A new paradigm was needed to enable low-power devices to participate in Internet. The Internet of Things paradigm [7] has emerged where all the embedded devices and networks are natively IP-enabled and Internet connected, independently of the used physical and media access control (MAC) layer protocol. The Internet of Things is considered the biggest challenge and opportunity for the Internet [7–9]. The growth of Internet of Things is hard to estimate for two main reasons: first, embedded systems are expected to have significant impact in several military and civil applications and, second, the growth is not directly dependent of human users.

In 2008, many industry leaders in promoting the use of Internet of Things formed the IP Smart Objects Alliance [10] and, in parallel, the IETF has opened the IPv6 over Low power WPAN charter [11]. IPv6 was considered more suitable than IPv4 for LoWPAN networks [12] because it provides both a much larger addressing space and better auto configuration mechanisms.

In large LoWPAN networks, packets have often to use multiple radio hops to reach their destination. The multi-hop forwarding is motivated by the fact that the sending node may not have radio range to reach the targeted destination node. Multi-hop forwarding increases the LoWPAN size, but does not necessarily provide multiple simultaneous paths to the same destination. The mesh network topologies can both provide multi-hop and path diversity [13, 14]. Hence, a routing protocol to support multi-hop mesh network is crucial [15], which must take into account the very demanding features of LoWPAN networks.

Mobility support is vital for the success of 6LoWPAN [16, 17]. Many applications require mobility support at node level while others require mobility support at the gateway router level. Therefore, the support of mobility must consider both node and network mobility. While physical mobility is easy to understand, there are also other more subtle causes of topology changes, in LoWPAN, caused by interference or fading for example.

This paper provides a survey on routing and mobility solutions to support IPv6 communications in all WSN nodes. The remainder of this paper is organized as follows. Section 2 analyses the standards IEEE 802.15.4 and IEEE 802.15.5. The two following sections focus on 6LoWPAN architecture (Section 3) and on 6LoWPAN routing approaches (Section 4). Node and network mobility support on 6LoWPAN is addressed in Section 5. Section 6 summarizes the current research challenges of effective routing and mobility on 6LoWPAN networks. Finally, Section 7 concludes the paper.

2. IEEE 802.15.4 AND IEEE 802.15.5 PROTOCOLS

IEEE 802.15.4 is a standard for communications on LoWPAN networks introduced by IEEE Computer Society to address the low-power and low-rate wireless personal area networks (PAN) requirements. The IEEE 802.15.4 [1] protocol defines the physical (PHY) and the MAC layers for such networks and is a *de facto* protocol for WSNs [3, 18, 19]. The IEEE 802.15.4 PHY defines

DOI: 10.1002/dac

three physical operation modes: (i) 20 kbps at 868 MHz, (ii) 40 kbps at 915 MHz and (iii) 250 kbps at 2.4 GHz (DSSS). A device in an 802.15.4 network can use either a 64-bit IEEE address or a 16-bit short address assigned during the association procedure. An 802.15.4 network can accommodate up to 64k (2^{16}) devices. The frame length is limited to 127 bytes because low-power wireless links are used in communications and the sensor devices have limited buffering capabilities.

The IEEE 802.15.4 defines two types of devices: full-function devices (FFD) and reduced-function devices (RFD). FFD devices support all network functionalities and, thus, can participate in peer-to-peer topologies to support multi-hop communications. RFD devices support a limited set of functionalities and are used to measure physical parameters and to execute non-complex tasks (they do not support multi-hop communications and can be used only in star topologies).

FFD and RFD devices organize themselves in PAN. A PAN is controlled by a PAN coordinator, which has the function of setting up and maintaining the PAN (obviously, only a FFD device can assume the role of PAN coordinator).

The IEEE 802.15.4 MAC provides two modes of operation: the asynchronous beaconless and the synchronous beacon-enabled mode. The beaconless mode requires that nodes listen for other nodes transmission all the time, which can drain battery power fast. The beacon-enabled mode is designed to support the transmission of beacon packets between transmitter and receiver providing synchronization among nodes. In the beacon-enabled mode, the PAN coordinator broadcasts a periodic beacon containing information about the PAN. Synchronization provided by the beacons allows devices to sleep between transmissions, which results in energy efficiency and extended network lifetimes.

In the beacon-enabled mode, the period between two consecutives beacons defines a superframe structure that is divided into 16 slots. Beacons always occupy the first slot, while the other slots are used for data communications. In these slots, slotted carrier sense multiple access with collision avoidance (CSMA/CA) is used for data transmission. In order to support low-latency applications, the PAN coordinator can reserve one or more slots, designated by guaranteed time slots, which are assigned to devices running such applications (in this case, these devices do not need to use contention-based medium access mechanisms). In the beaconless mode, there is no superframe structure and no guaranteed time slots. As a consequence, only random access methods, such as unslotted CSMA/CA, can be used to medium access.

A PAN can adopt one of the following two network topologies [1, 20]:

- *Star topology* (Figure 1): a master–slave network model is used where an FFD device assumes the PAN coordinator role and controls all networks operations; the other nodes can be either RFDs or FFDs and communicate only with the PAN coordinator (this topology is better suited for small networks).
- *Peer-to-peer topology* (Figure 2): FFD devices can communicate with other FFDs within its radio range and can use multi-hop communications to send messages to other FFDs outside of its radio range; RFDs can communicate only with FFDs.

In the beacon-enabled mode, a peer-to-peer topology also has a PAN coordinator, but additionally devices can communicate directly with each other. This allows the set up of more complex

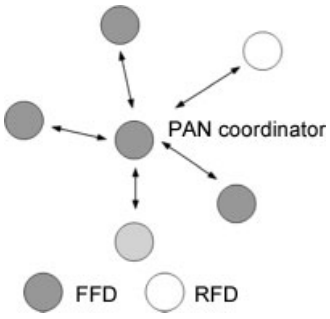


Figure 1. Illustration of a star topology.

DOI: 10.1002/dac

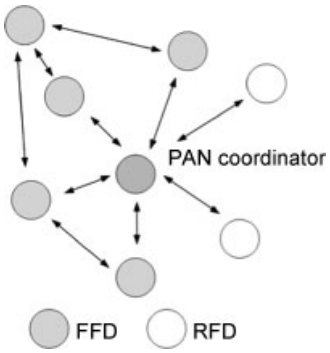


Figure 2. Illustration of a peer-to-peer topology.

topologies, such as mesh or cluster-tree. A mesh PAN network can be set up in either a full mesh or a partial mesh topology. In the full mesh topology, all node pairs are directly connected. In the partial mesh topology, some nodes are connected to all others, but other nodes are connected only to a limited number of neighbor nodes. An obvious advantage of mesh over star topology is its better reliability via route redundancy. Note that in the mesh topology, there is no central node that must be directly connected (i.e. within the radio range) to all other nodes. This fact enables many other potential advantages such as (i) larger network coverage without increasing transmitting power or receiving sensitivity, (ii) easier network set-up (in a star topology, the PAN coordinator must be as much geographical centered as possible) and (iii) better device battery life (for the same network coverage, the distance between communicating devices is smaller on average). Nevertheless, a very important issue is how to support beacon-enabled mode on mesh topologies [21]. Without such solutions, the potential better device battery life advantage cannot be obtained with the mesh topology.

The cluster-tree topology is a special case of a peer-to-peer topology, where nodes associated with a single PAN coordinator are arranged in a tree in accordance with their parent–child relationships. These relationships form a tree rooted at the PAN coordinator. Each cluster has a cluster head coordinator, acting as an intermediate aggregator and forwarding the data between different clusters.

As IEEE 802.15.4 does not define any layer 2 routing mechanism, the IEEE 802.15.5 [22], also known as mesh WPAN, was launched in November 2003 to develop the necessary mechanisms that must be present in physical and medium access control layers of WPANs to enable peer-to-peer topology. The work of the IEEE 802.15.5 group covers both high-rate and low-rate WPANs. The outcome of this working group is applicable both on IEEE 802.15.3 (a protocol addressing high rate WPANs) and IEEE 802.15.4 protocols.

In LoWPAN networks, the nodes are assumed to have severe resource constraints and, in such cases, a table-less addressing and routing scheme improves the network performance. Usually, such schemes rely on some sort of logical trees in which the position of devices are implicitly indicated by the addresses (a tree topology is formed when the active connections among nodes define a single path between any pair of them). IEEE 802.15.5 adopted a meshed tree-based routing algorithm. In the tree, a single node is set as the root node (designated as the Mesh Coordinator) as illustrated in Figure 3. The tree links are the links selected to be part of the routing tree and the mesh links are used as alternative paths to the tree. The meshed tree-based routing algorithm has two major components: the tree formation that defines the links belonging to the tree, and the distributed link state generation that enables to detect the useful mesh links.

The tree formation component is break up in two stages: the first stage is node association and the second stage is address assignment. At the beginning, the first active node becomes the Mesh Coordinator (this is the beginning of node association stage) and starts accepting association requests from other nodes to join the network. When a node A associates itself with a node B, node B becomes the parent node of node A and node A becomes one child of node B. After a node is successfully associated with a parent node, it also starts accepting association requests from other

DOI: 10.1002/dac

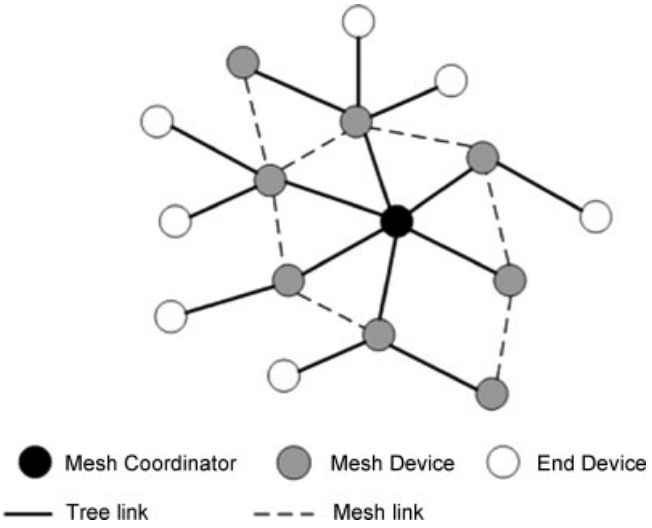


Figure 3. Illustration of a IEEE 802.15.5 mesh network topology.

nodes to join the network. Only FFD devices can accept new nodes, while RFD devices cannot. When a joining node receives multiple association responses, it should choose only one node as a parent. Each device records its parent and child devices in its neighbor list. At this stage, all communications are done using 64-bit IEEE addresses and devices communicate only with nodes that are present in their neighbor range. The tree is complete when no more nodes are waiting to join the network and address assignment stage begins.

In address assignment stage, all leaf nodes (devices with no child nodes) request one address to their parent nodes for its own use. In fact, they can request more than one address (some additional addresses can be reserved for further use). Meanwhile, parent nodes determine the number of requested addresses from their child nodes after all requests are received, then add their own required addresses and request the total number of addresses to their parent nodes. This process repeats until the Mesh Coordinator (the root of the tree) receives the information from all branches. Afterwards, the Mesh Coordinator assigns and disseminates the requested addresses (the address block assigned to each branch is specified by the beginning address and the ending address given). After the address assignment stage is complete, a tree is formed and each node has an address table with one address per child node and the address of its parent node (besides its own address). Depending on the Mesh Coordinator configuration, either the IEEE 64 bit or 16 bit addresses are used in the address assignment stage.

At this point, a tree of active connections covering all devices is set and data packets can now be routed between all devices based on these connections (the 'tree links' in Figure 3). When a node receives a frame: (i) it first checks if the frame destination address is its own; if yes, it stores locally the frame, if not, (ii) it checks if the frame destination address is in its child address table; if yes, it sends the frame to its destination address, if not, it sends the frame to the address of its parent node. This process is repeated at each intermediate node until the packet reaches its destination.

The routing algorithm as presented until now, suffers from two main problems [23]. First, the routes are not optimal because the frames can only be rooted up and down along the tree. Second, the tree structure is not robust, because there is only one path for each destination. To solve these two problems, IEEE 802.15.5 combines tree routing with local link state routing through the distributed link state generation.

Each node periodically broadcasts its own Hello messages and these messages specify a predefined number of hops. Each node also rebroadcasts the Hello messages of neighbors if the predefined number of hops has not been reached yet. After a while, the devices have the neighbor information up to the number of hops defined in the hello messages. This information is stored in its neighbor

DOI: 10.1002/dac

list for routing purposes. Now, a device searches its neighbor list for the destination address before using the tree routing. Depending on the distance between the two communication nodes and the network topology, devices may find that the destination is one of its neighbors, so a mesh link can be used minimizing the number of hops to reach the destination. Besides routing optimality, robustness is also enhanced since there are now multiple paths between many network devices. Nevertheless, the efficient usage of address space as well as tree link repair are among the biggest challenges for protocol designers.

3. 6LOWPAN ARCHITECTURE

Originally, the IP protocol was considered too heavy to be processed by LoWPAN nodes due to their scarce resources. Recently, the industry and the scientific communities start to rethink many misconceptions about the use of IP in all LoWPAN nodes [24–27]. Supporting IPv6 on sensor nodes simplifies the task of connecting LoWPAN devices to the Internet and creates the conditions to realize the paradigm of Internet of Things. Additionally, by using IPv6-based protocols, users can deploy tools already developed for commissioning, configuring, managing and debugging these networks [24]. Moreover, the application developing process is simplified and becomes open.

Traditionally, the routing process operates at link layer level [13, 26, 27]. With the IP support at all nodes, the routing process can now operate either at link layer and or at IP network layer. Therefore, it creates new opportunities to define routing protocols that fit to the LoWPAN purposes.

Nowadays, the IEEE 802.15.4 protocol is widely accepted as the PHY and MAC layer protocol for LoWPANs. Nevertheless, the network layer protocol must comply with the constraints imposed by the IEEE 802.15.4 protocol and the properties of the standard IPv6 protocol do not fully match with such constraints. The IETF created the 6LoWPAN working group to define the support of IPv6 over IEEE 802.15.4 LoWPAN networks. This support was defined by an additional adaptation layer introduced between data link and network layers, as specified in Figure 5.

Three different LoWPAN architecture types were defined: (i) *Ad hoc* LoWPAN, with no infrastructure, (ii) Simple LoWPAN, with one edge router and (iii) Extended LoWPAN with multiple edge routers. These three different 6LoWPAN network architecture types are illustrated in Figure 4.

The 6LoWPAN working group focused on the following items [27]: (i) to define limited extensions to IPv6 neighbor discovery (ND) protocol [28] tailored for low-power networks, (ii) to describe mechanisms allowing compression of 6LoWPAN headers, to reduce the header overhead, (iii) to define the '6LoWPAN Architecture' describing the design and implementation of 6LoWPAN networks, (iv) to define 6LoWPAN routing requirements describing 6LoWPAN-specific requirements on routing protocols used in 6LoWPANs, (v) to produce use cases for 6LoWPAN defining, for a small set of applications with sufficiently unique requirements, how 6LoWPANs can solve those requirements, and which protocols and configuration variants can be used for these scenarios and (vi) to define the threat model of 6LoWPANs documenting security mechanisms issues.

Two RFCs were released, the RFC 4919 [29] and the RFC 4944 [30]. The first document describes the assumptions, problem statement and goals of 6LoWPAN. The second describes (i) the frame format for transmission of IPv6 packets, (ii) the method for defining IPv6 link-local addresses and stateless auto configured addresses, (iii) an header compression scheme using shared context and (iv) the frame delivery process in a link-layer in IEEE 802.15.4 mesh network. The following issues are under open discussion: (i) compression format for IPv6 datagrams in 6LoWPAN networks [31], (ii) design and applications spaces for 6LoWPANs [32], (iii) 6LowPAN ND [33] and (iv) problem statement and requirements for 6LoWPAN routing [34].

3.1. Requirements

Low bandwidth, low-power resources and the maximum link-layer packet size of 127 bytes are the most relevant characteristics of the IEEE 802.15.4 standard [1]. Implementing standard IPv6

DOI: 10.1002/dac

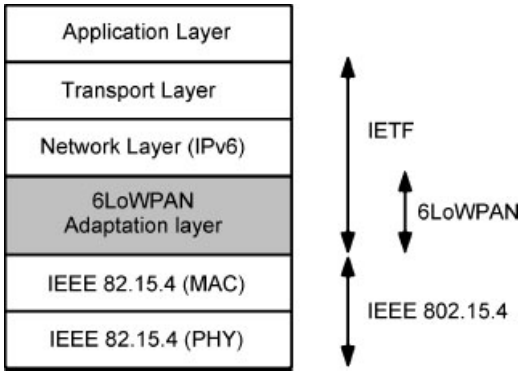


Figure 4. 6LoWPAN layered architecture.

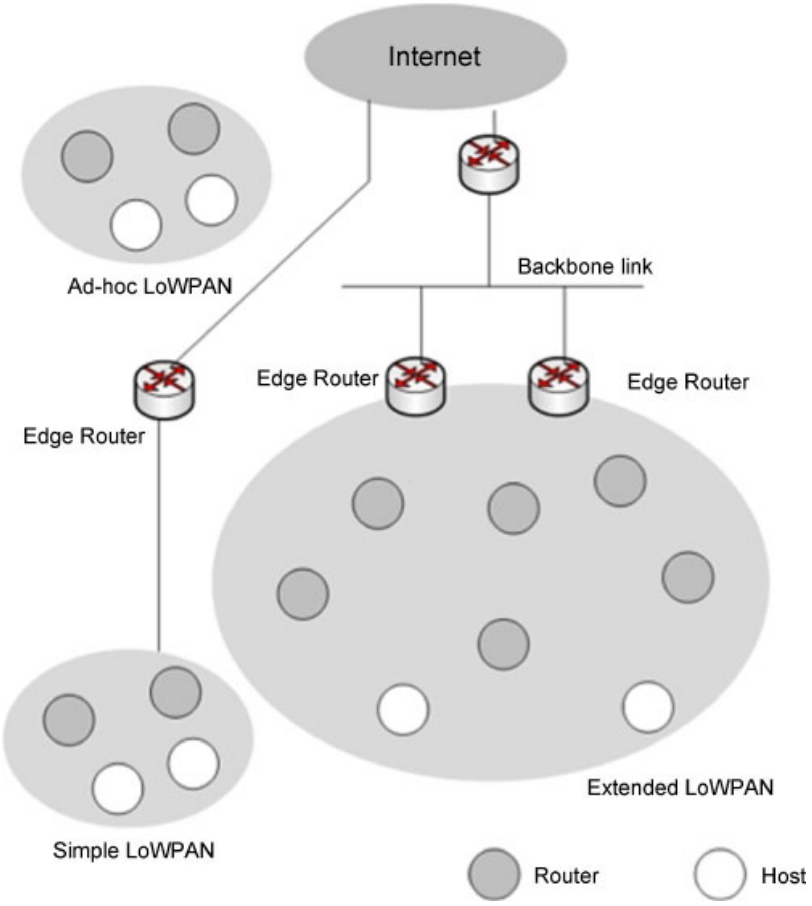


Figure 5. 6LoWPAN network architectures.

headers over LoWPAN would result in extremely small payloads for higher-level protocols (as illustrated in Figure 6): in the best case, the maximum size of an IP packet is 88 bytes; the IPv6 header has a minimum size of 40 bytes, which results in 48 bytes for upper-layer protocols like TCP or UDP; the length of the TCP header is another 20 bytes, which results in 28 bytes available for the application-layer protocol (in the TCP case).

The above description motivates the use of cross-layer compression to reduce the protocol header overhead. Besides this requirement, there are other important ones that must be dealt with.

DOI: 10.1002/dac

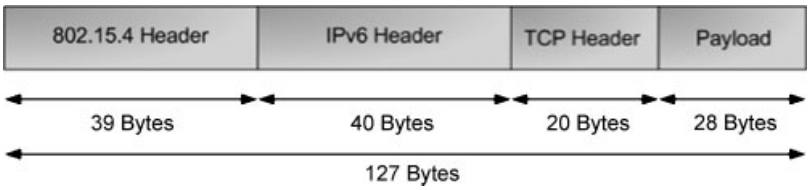


Figure 6. IPv6 over IEEE 802.15.4 headers size.

First, in order to guarantee that 6LoWPAN is compliant with the IPv6 maximum MTU of 1280 bytes, fragmentation and reassembly are required. Then, a stateless address auto-configuration is also necessary to reduce the configuration overhead. Finally, various security requirements are also required, in particular countermeasures against man-in-the-middle and denial of service attacks.

3.2. Adaptation layer

To address the above requirements, the 6LoWPAN protocol has introduced the adaptation layer, as previously explained. Rather than defining a single header (like IPv4 or ZigBee), the 6LoWPAN use stacked headers as the original IPv6 protocol does. To understand the reason for that, let us consider the example of a device sending small packets directly to another device. In this case, it does not need to use unnecessary header fields for mesh networking or fragmentation and it uses only the minimum necessary headers. The 6LoWPAN standard defines four header types: (i) the dispatch header, (ii) the IPv6 compression header, (iii) the fragmentation header and (iv) the mesh header. In the simplest case, only the dispatch and compression headers are used. At the beginning of each header, a header-type field identifies the header format.

The 6LoWPAN header compression is defined in RFC 4944 [30]. It defines a stateless compression scheme consisting of two parts: the header compression one (HC1) and the header compression two (HC2). HC1 allows compressing the IPv6 header with an original size of 40 bytes into only 3 bytes in the best case. Similarly, the HC2 describes a compression format to reduce the length of the TCP header (6LoWPAN supports also the compression of UDP header, which is denoted also as HC2).

6LoWPAN uses the fragmentation header to support the minimum value required by IPv6 for the underlying MTU (which is 1280 bytes). Whenever the payload is too large to fit into a single IEEE 802.15.4 frame, it is fragmented into several packets.

The IEEE 802.15.4 header contains only the source and the destination address of the next hop. If a packet is to be transmitted to a node that is not a neighbor of the source, a routing protocol is needed to implement this functionality (such as IEEE 802.15.5, for example). With IPv6, the source and final receiver addresses are included in the IPv6 hop-by-hop option header. Nevertheless, using the compression header this information is lost. The solution to this problem is the mesh header, which is designed to support layer 2 forwarding (as will be further explained later).

3.3. Addressing and neighbor discovery

The 6LoWPAN packet forwarding in a multi-hop environment can be implemented either in the link layer or in the network layer [15, 35] and both approaches are supported. Forwarding on the network layer is called route-over and forwarding below network layer is called mesh-under. 6LoWPAN routing approaches are addressed in the next section.

Addressing is mainly based on the stateless address configuration of IPv6. The RFC 4944 defines in detail how the interface address is determined. In general, it is based on the IEEE EUI-64 address of the IEEE 802.15.4 device. In mesh-under, a link-local address is sufficient for communications within the LoWPAN, but a routable address is required to communicate with other networks. In route-over, a link-local address is sufficient to communicate with nodes in direct radio

DOI: 10.1002/dac

communication, but a routable address is required to communicate with devices that are multiple radio hops away [13, 14].

The ND protocol is used by each node to discover the neighbor nodes and to maintain the reachability information in a similar way as in IPv6, including prefix discovery and default route configuration. This protocol also performs address resolution, neighbor unreachable detection and duplicated address detection. ND is currently defined only for operation on a single IP link. In a mesh-under configuration that emulates a single IP link over the entire LoWPAN, ND cannot be used unless some modifications are considered.

There are significant challenges to use the current ND specification within LoWPANs. First, ND uses link-local multicast for sending address resolution solicitations, router advertisements and duplicated address detection messages and, currently, LoWPAN does not support multicast communications due to energy conservation (to overcome this limitation, the simplest solution is to have all nodes processing all ND messages, which is an expensive operation, especially in large networks with multiple radio hops). Second, IPv6 ND was not designed for non-transitive wireless links. Third, some problems are experienced when the ND protocol runs in large multi-hop LoWPAN networks [36]. Finally, the ND protocol is too verbose and may generate an overhead in the number of transmitted messages. A new ND protocol is under discussion in IETF 6LoWPAN working group [33].

4. ROUTING APPROACHES IN 6LOWPAN MESH NETWORKS

Many protocols have been proposed for routing in LoWPAN [36, 37]. These routing mechanisms have taken into consideration the network purpose and the architecture requirements. Research efforts in LoWPAN have led to the development of several energy-aware routing protocols where the network lifetime maximization is the main concern [36]. Routing protocols in LoWPAN can be classified from the perspective of network structure in three classes: flat based routing, hierarchical based routing and location-based routing. In flat-based routing, all nodes have the same role or functionalities. In hierarchical-based routing, different nodes play different roles on the network: nodes with higher resources (energy, power computation and memory) can be used in multi-hop forwarding, whereas the other nodes can only be used for sensing operations. In location-based routing, sensor nodes are addressed by means of their locations, and route data using node positions. The node position can be estimated by the strength of received signals or using GPS (Global Positioning System).

WSN packets often need to travel through multiple radio hops to reach the destination. The multi-hop forwarding is motivated by the fact that the sending node may not have radio range to reach the destination node. To send a packet to another node, two main processes are involved—forwarding and routing. Both processes can be executed at layer 2 or at layer 3. Routing process usually uses a routing protocol to evaluate the best path to reach the destination. Each node maintains a routing information base that contains all the information needed to run the routing protocol. The routing information base is used to fill the forwarding information base (FIB). In the forwarding process, the router looks up the destination address of each data packet in its FIB to select the next network element to transmit the packet.

Routing protocols generally fall into two different classes: distance vector and link state. The distance vector uses the Bellman-Ford distributed algorithm to compute the paths. Each node sends to neighboring nodes its routing table, allowing neighbors to compute the best paths. Bellman-Ford distributed algorithm is simple, but can experience long convergence times, where incoherent routing information can cause routing loops and count to infinite situations [14]. Link state routing protocols were designed to solve the convergence time problems experienced by distance vector protocols. In link state, every node builds a map of the entire network and uses shortest-path algorithms, like Dijkstra's algorithm, to compute the best paths to each destination [14].

Two main approaches can be used to fill the FIB, (i) proactively or (ii) reactively. In the proactive approach, the routing table has information to reach all available destinations. In the reactive

DOI: 10.1002/dac

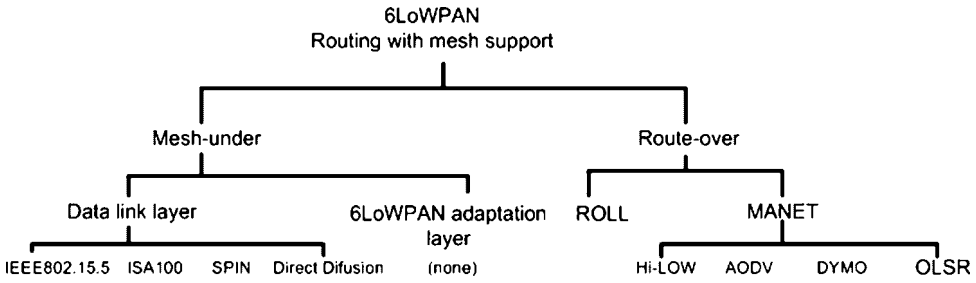


Figure 7. Routing protocol taxonomy.

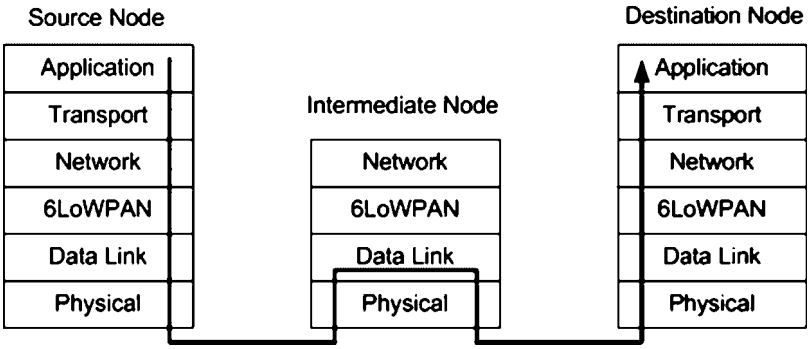


Figure 8. Link layer mesh-under routing.

approach, the routing table is filled on-demand. Proactive protocols may not fit well to LoWPAN because of high traffic overhead caused by the continuous updating of the routing tables but, on the other hand, reactive protocols may cause excessive delays.

When packets are received, the router searches the destination address in its FIB, selects the next hop and sends the packet encapsulated with new link layer addresses (it is the forwarding process). Usually LoWPAN nodes have only one interface and, therefore, the receiving and the sending interfaces are the same.

Routing and forwarding in a LoWPAN can be done in three different ways: (i) link layer mesh-under, (ii) 6LoWPAN mesh-under and (iii) route-over [35]. Link layer mesh and LoWPAN mesh are designated by mesh-under and are transparent to the IP network layer. IP routing is designated by route-over. Figure 7 shows the routing protocols and the taxonomy used in its analysis (this figure includes only routing protocols with multiple routing path support).

4.1. Mesh-under routing

In mesh-under, routing and forwarding are performed based on layer 2 addresses. The 6LoWPAN-working group has originally considered only the mesh-under approach to support routing. Mesh-under provides a virtual broadcast link to the IP protocol. In this case, the network layer can assume that all nodes within a subnet are directly reachable, hence, the IP model does not need to change. In the mesh-under approach, the routing and forwarding occur at data link layer (Figure 8) or at 6LoWPAN adaptation layer (Figure 9).

In the past few years, an intensive research addressing data link layer routing protocols was conducted. Flat and multipath routing protocols are the most relevant, because they are more suitable to be used in 6LoWPAN mesh networks. Four flat and multipath routing protocols have been proposed: ISA100 [38], IEEE 802.15.5 [22], SPIN family routing protocol [39] and Directed Diffusion [40]. Nevertheless, more research is still required to evaluate the performance of such protocols in 6LoWPAN networks.

DOI: 10.1002/dac

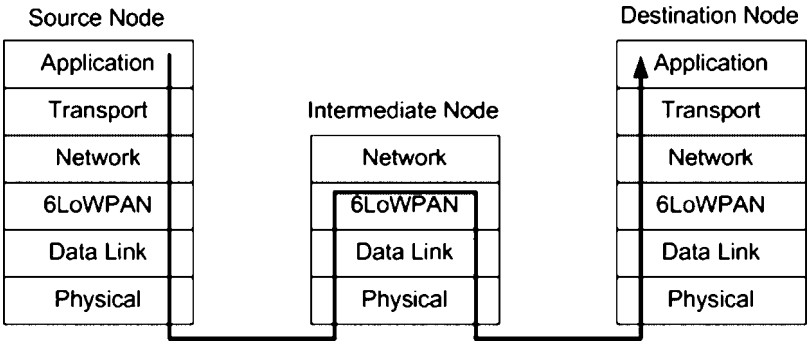


Figure 9. 6LoWPAN mesh-under routing.

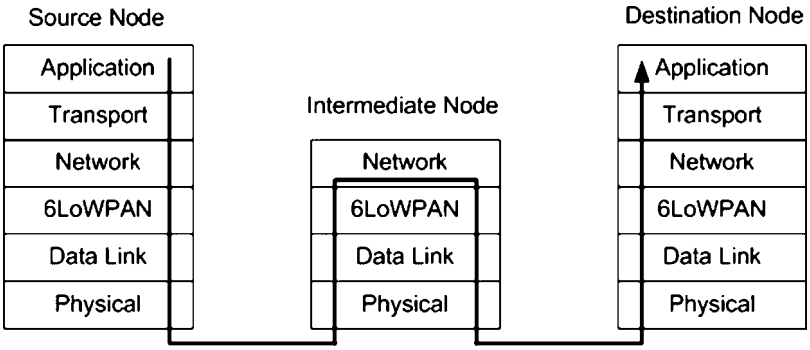


Figure 10. 6LoWPAN route-over routing.

In the case where the link-layer routing process operates at 6LoWPAN adaptation layer (Figure 9), there is a problem to solve: the link layer transports the source and the destination layer 2 addresses. These addresses change on every hop because forwarding process overwrites the link layer destination address by the next hop link layer address and the source address by the link layer address of the forwarder node. To forward the packet to its destination, the node needs to know the destination link layer address. Moreover, the destination nodes need to know the layer 2 source address to reassembly packets that were segmented in the source node. To solve this problem, the 6LoWPAN adaptation layer includes the mesh header to store the layer 2 source and the destination addresses (this header also includes the layer 2 equivalent of the IPv6 Hop Limit value).

In spite of the existence of the mesh header in the 6LoWPAN adaptation layer, as far as we know, there is not yet any proposal for a 6LoWPAN mesh-under protocol [17].

4.2. Route-over routing

In the route-over approach, the routing decision is done on the network layer (Figure 10). IP routing in LoWPAN networks has special characteristics when compared with regular IP networks, such as: (i) 6LoWPAN routers typically perform forwarding on a single interface, so the packets are received in one interface and forward to the next-hop using the same interface, (ii) all nodes of the same LoWPAN share the same IPv6 prefix, (iii) LoWPAN nodes have strict resource constraints, (iv) LoWPANs are stub networks and (v) the LoWPAN routing topology may change frequently.

The connection of 6LoWPAN networks to the Internet involves the use of two different routing protocols (Figure 11): the intra LoWPAN routing protocol used for routing inside the LoWPAN domain and the border routing protocol used to connect the LoWPAN to other networks, such as Internet. Only intra LoWPAN routing is considered in this survey. Routing metrics are used by

DOI: 10.1002/dac

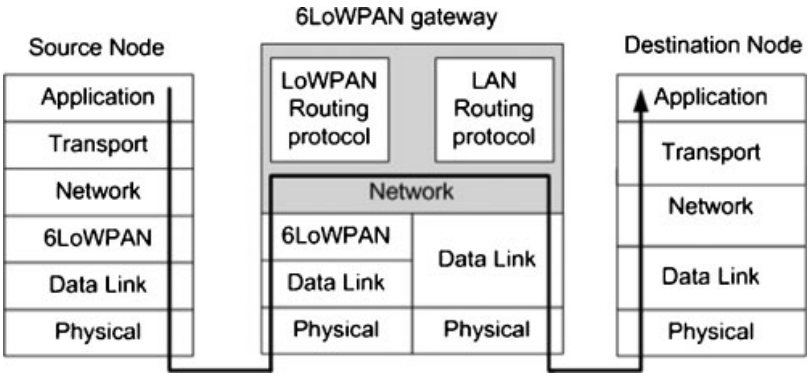


Figure 11. 6LoWPAN gateway.

routing protocol to select the best paths. Routing metrics used by 6LoWPAN routing protocols are dependent on the application where they are being used [37].

Two IETF working-groups have made contributions to solve the 6LoWPAN routing problem. One is Routing Over Low-power and Lossy networks (ROLL) [41] and the other is Mobile *Ad hoc* Network (MANET) [42].

The IETF MANET working-group was formed in 1997 to define solutions for wireless *ad hoc* IP-based networks. Mobile multi-hop *ad hoc* networks (MANETs) are characterized by the non-existence of any fixed network infrastructure and all network elements can be mobile. In a MANET, there is no distinction between a host and a router and all nodes can be sources as well as forwarders of traffic. Because of these characteristics, paths used to connect the nodes may be very unstable and go down at any time, making communications over *ad hoc* networks difficult [43]. The protocols designed for MANETs can be used in 6LoWPAN networks because both networks have similar requirements. However, the biggest challenge for applying a MANET routing protocol to 6LoWPAN is in reducing the overhead of signalling messages and simplifying the routing algorithm.

The standard MANET routing protocols establish only one path from a source to a destination [44]. Multi-path MANET protocols, which establish multiple disjoint paths during a single route discovery phase, have a number of benefits, such as lower overhead, lower packet loss rate and increased reliability compared with their single-path equivalent protocols [45].

Multi-path routing protocols proposed for MANETs can be classified as (i) delay-aware multi-path routing protocols, (ii) reliable multi-path routing protocols, (iii) minimum overhead multi-path routing protocols, (iv) energy efficient multi-path routing protocols and (v) hybrid multi-path routing protocols [46].

Several routing protocols have been proposed to MANETs. Among all algorithms, only a few were submitted for publication as experimental RFC: (i) *Ad Hoc* On Demand Distance Vector (AODV) [47] Routing, (ii) The Dynamic Source Routing Protocol (DSR) [48] for Mobile *Ad Hoc* Networks, (iii) Optimized Link State Routing Protocol (OLSR) version 1 [49] and (iv) Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) [50]. Dynamic MANET On-demand (DYMO) [51] and the Optimized Link State version 2 (OLSRv2) [52] routing protocols are under discussion in the IETF MANET working-group. The AODV routing protocol is a reactive protocol supporting both unicast and multicast routing. The DSR protocol is also a reactive protocol and can support multi-path routing with a few modifications. OLSR and TBRPF are both proactive link state routing protocols. Dynamic MANET On-demand is a reactive routing protocol and it is basically an enhancement of the AODV protocol.

Research is currently in progress to study their suitability for Wireless Mesh Network environments [46, 53, 54]. A number of multi-path routing protocols are available for MANET, most of them based on standard routing protocols [45, 53, 55]. Several studies have been made to assess the use of the AODV, OLSR and DSR standard MANET routing protocols in LoWPAN networks, some

DOI: 10.1002/dac

of them with encouraging results [15, 55]. Some studies refer also the DYMO working-progress routing protocol [51].

The ROLL working group within the IETF is currently developing a new routing protocol for IP LoWPAN networks. This protocol is being designed to meet the requirements identified by the working group for the use of LoWPAN on monitoring of buildings, home automation and other industrial applications of sensor networking. The ROLL working-group began with a requirements analysis for these application areas. Based on these requirements, a survey was made of existing IETF routing protocols. The survey has concluded that it is impossible to use the existing IETF routing protocols without modifications [56]. The ROLL working group has already defined the routing requirements for urban [57] and industrial low power and lossy networks [58] applications. Currently, it is focused on defining (i) a new routing protocol compliant with LoWPAN requirements, (ii) a set of metrics to be used on routing protocols [59], (iii) a basic architecture requirements [60] and (iv) a security framework [61, 62] to be used in LoWPAN networks.

The ROLL basic architecture is presented in Figure 12. In this architecture, routers with interfaces connected to the low-power and lossy networks (LLN) and other IP networks are designated by LLN edge routers (LBR). More than one LLN edge router can be used to connect one LLN to the backbone link. LLN routers and hosts form the low-power lossy network. Routers participate in the routing algorithm while hosts do not participate in the routing algorithm but, instead, choose a neighbor router to send all packets. The algorithm uses a graph structure between routers, hosts and edge routers. After topology setup, the routing protocol maintains upstream (from node to edge router) and downstream (from edge router to node) paths (Figure 13). Forwarding along these paths is performed using IPv6 forwarding. The edge routers are responsible for coordinating the multi-topology routing and traffic engineering. Node-to-node optimizations may be performed in a distributed mode (Figure 12).

After nodes establish the neighborhood relations, the first step is to build a routing topology. IPv6 nodes use already ICMPv6 ND messages to topology discovery; these messages can be

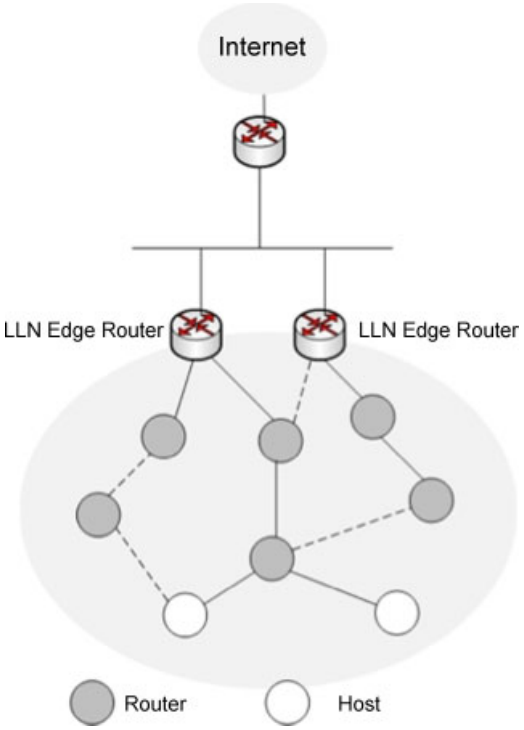


Figure 12. ROLL architecture.

DOI: 10.1002/dac

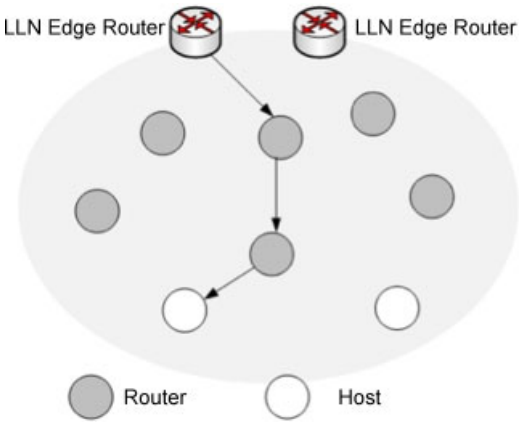


Figure 13. Downstream routing—path from LLN edge router to host.

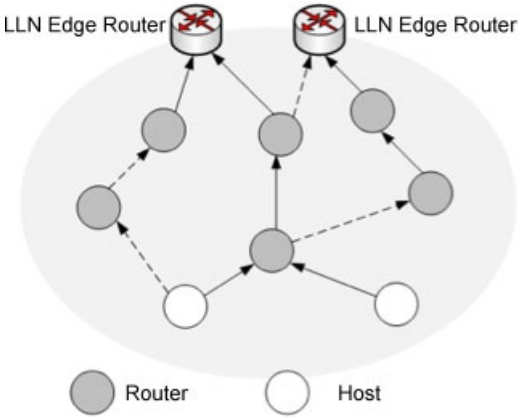


Figure 14. Multipath upstream routing—paths from host towards LLN edge router.

reused to build the routing topology. Routers use these messages to join the network by choosing a set of default next-hop routers towards edge router. A set of multiple distance vector paths from nodes to edge routers are automatically set up, which can be used to forward packets towards edge router. In this routing topology, nodes can change their position. To maintain downstream routing information, nodes disseminate path cost information upstream towards edge routers. ICMPv6 ND messages are used to maintain the routing topology.

The routing topology enables traffic forwarding between LLN nodes and edge routers (downstream) (Figure 13) and the route dissemination enables traffic forwarding from edge routers towards LLN nodes (Figure 14).

The ROLL routing protocol enables multipath routing support using multiple techniques. To achieve many of the ROLL requirements, additional features are being considered by IETF ROLL working-group, for example [41]: (i) better node-to-node communication support, (ii) network mobility integration with NEMO [63] and (iii) traffic engineering support on edge routers.

4.3. Mesh-under vs Route-over

The scientific community has not reached yet a consensus about the best routing approach since both mesh-under and route-over approaches have their benefits.

When originally defining 6LoWPAN, the working group assumed a mesh-under approach. Mesh-under provides a broadcast network abstraction to the IP layer and, as a consequence, the IP protocol does not need to change. In mesh-under approach, fragments can be delivered over

DOI: 10.1002/dac

multiple hops without requiring fragmentation and reassembly at each hop. Moreover, mesh-under also allows the use of multiple paths to deliver fragments for a given datagram. However, there are some significant drawbacks to this approach. Network diagnostic tools, such as *traceroute* and some SNMP-based diagnostics, cannot be used in mesh-under approach since every node in the mesh is one hop away, from the viewpoint of IP. Hence, *traceroute* is not able to show the path through the mesh. Additionally, it is not possible to use standard tools to query the routing tables because the underlying mesh is not utilizing an IP routing table. Therefore, new *traceroute*-like tools and routing table lookup tools must be developed, in order to diagnose network failures.

An alternative to mesh-under approach is to use either currently available IP routing protocols or new IP routing protocols designed to be more efficient for low-power and limited bandwidth networks. In route-over approach, each radio hop is an IP hop. IP routing model works by separating the routing engine and forwarding engine into distinct functions. The routing function is responsible for maintaining the routing tables and the forwarding function examines the routing table to find the best next hop node to forward the packet. Sometimes, the routing table is as simple as a single 'default route' entry with the neighbor address (this is sufficient for an IEEE 802.15.4 RFD or edge node with a single parent). This design enables the use of tools like *traceroute* as valuable tools for mesh network diagnostics. Nevertheless, two issues need to be carefully addressed [25, 33]. One is the ability for the routing protocol to be able to query the radio device for various parameters and characteristics, node resources, link performance and other information in order to be able to properly advertise and utilize route information. Another is to understand how one device per subnet might impact the IP model since an IP route between nodes would require each node to be in its own subnet. Even though IPv6 offers a large number of subnets, investigation is necessary. The main drawback of route-over approach is that it requires 6LoWPAN fragmentation and reassembly at every radio hop. As a consequence, the transmission of fragments of a given datagram to its final destination through multiple paths cannot be done. Note that fragmentation and reassemble occur at the 6LoWPAN adaptation layer and, therefore, only the first fragment carries the IP header. In [35], mesh-under and route-over are compared and some interesting conclusions are drawn. First, the route-over scheme is more reliable to deliver fragments; second, if selective retransmission is used in both routing approaches, then the route-over scheme performs better in terms of the total number of transmissions and, finally, mesh-under scheme outperforms route-over scheme in the case of total delay.

Choosing the most appropriate routing approach depends on several factors, such as the reliability when fragmentation is used. Moreover, both approaches may be used in a large network. The large network could be separated in several small networks. Mesh-under approach can be used inside the small networks and a route-over approach to provide global connectivity.

5. MOBILITY IN 6LOWPAN

There are several causes for topology changes in LoWPAN networks. In these networks, often devices are integrated in moving machines, carried by people or animals or incorporated on equipments. In other applications, the edge routers themselves may be mobile, changing their point of attachment on the Internet. The term node mobility refers to the cases where a single device changes its point of attachment. The term network mobility refers to the cases where the entire network, including the edge router, changes its point of attachment.

While the mobility caused by physical movements is easy to understand, there are other more subtle causes of topology changes in which the devices do not move. Besides physical movement, the causes of topology change can be characterized as radio channel changes, network performance, sleep schedules and node failure. Changes in the transmission channel, such as fading and interferences, cause changes in radio propagation. As a consequence, these changes may involve topology changes even without physical movement. Packet losses and delay on wireless network may be caused by poor signal strength, overload channel capacity, collisions or node congestion. Hence, high rate packet losses may also cause a node to change its point of attachment. LoWPAN

DOI: 10.1002/dac

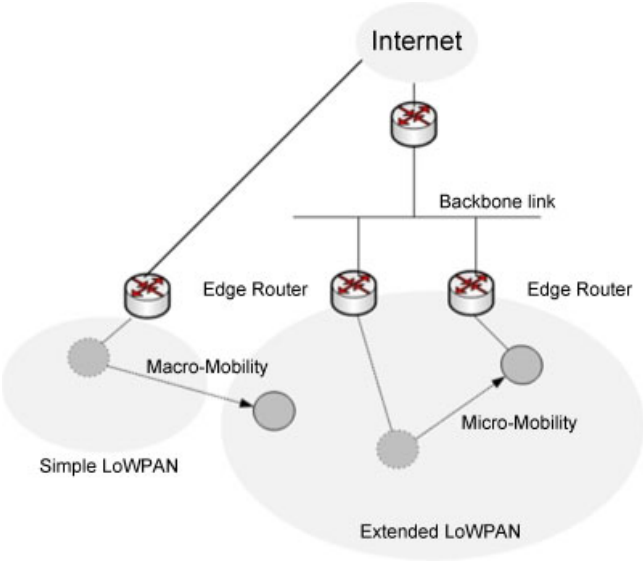


Figure 15. 6LoWPAN macro and micro-mobility.

nodes stay a long time in sleeping mode to preserve battery power. However, nodes must be synchronized so that they can communicate. If not, the node may change its neighbor relations, which is equivalent to change its point of attachment. LoWPAN nodes are often inaccessible due to their power constraints. Hence, the failure of a node causes a topology change for nodes using it as a neighbor.

Generically, mobility means the act of a node changing its network point of attachment. We can distinguish two kinds of mobility processes: roaming and handover [64]. Roaming is the process used to extend the connectivity service in a location that is different from the home location where the service was registered. Roaming ensures that the wireless device keeps connected to the network. Handover is the process that allows a mobile device to change from one point to another point of attachment, both belonging to the same network domain, and maintaining all ongoing communications. The handover can be both processed at layer two or layer three, according to the OSI model.

Mobility can also be classified as micro-mobility and macro-mobility (Figure 15). In micro-mobility, the device mobility occurs within a network domain, i.e. the mobile device network prefix does not change. The macro-mobility refers to mobility to another network domain with different network prefixes. In micro-mobility, only the handover mobility process is involved. In macro-mobility, both the handover and roaming mobility processes are involved. The macro-mobility is more difficult to support when compared with micro-mobility because it requires network prefix change.

When a node changes its point of attachment, there are sets of procedures that must be executed before being able to send and receive packets again. These procedures include [65]: (i) establishing a new link, (ii) fulfilling conditions for access control to the link, (iii) acquiring a topologically correct IP address. Depending on the type of mobility, some of these procedures may not be necessary. For example, when micro-mobility occurs, the link layer may be able to deal with mobility without the network layer participation. For 6LoWPAN, mesh-under link layer techniques may deal with micro-mobility inside a LoWPAN, although actually no well-known support mechanisms exist [17]. LoWPAN link layer technologies, such as IEEE 802.15.4, tend to leave mobility support to the network layer.

ND for 6LoWPAN [33] include support to deal with micro-mobility in extended LoWPAN networks. This is achieved by using an ND proxy technique and whiteboard synchronization between edge routers, allowing a node to keep the same IPv6 address regardless of its point of attachment within the extended LoWPAN.

DOI: 10.1002/dac

Several macro-mobility node-based solutions have been proposed. The most known are: (i) application-based mobility and (ii) Mobile IPv6 [65].

In many cases, it is the application layer who deals with the mobility effects. This solution is application dependent and does not provide a global mobility solution. Some application protocols have methods for dealing with mobility. The Session Initiation Protocol (SIP) [66] includes a SIP Uniform Resource Identifier (URI) used to globally identify a user and that can be used for tracking 6LoWPAN nodes. Moreover, SIP has Re-INVITE messages to notice the other nodes about the new IP address. However, the SIP header format is too verbose to be used over 6LoWPAN without modification [17]. A recently proposed draft defines a solution to use SIP in 6LowPAN networks [67].

Mobile IP enables a Mobile Node (MN) to maintain the same address in all communications with any Correspondent Node (CN) while moving from one network to another. Mobile IPv4 [68] defines two entities to provide mobility support: a Home Agent (HA) and a Foreign Agent (FA). The HA is statically assigned to the MN based on its permanent IPv4 Home Address. The FA is assigned to the MN based on its current location. The FA has an associated IP address called care-of address. When MN is outside its home network, packets destined to him are intercepted by its HA, and tunneled to the FA using the care-of address as the exit tunnel address. Then, the FA decapsulates the packets and forwards them directly to MN. In the opposite direction, MN sends all packets directly to the CN using its Home Address as source address (this is known as the triangular routing concept).

The existence of an FA is not strictly required and, as an alternative, MN may use a co-located care-of address that can be obtained via some dynamic IP address assignment protocol such as DHCP. In this case, the tunneling is done directly between the MN and the HA. Mobile IPv6 [65] has some improvements over mobile IPv4. First, due to the huge IPv6 address space and the IPv6 address auto configuration, the FA entity is no more necessary. Second, mobile IPv6 uses the Binding Mechanism on the destination options of IPv6 header to announce to the CN its current care-of address and, in this way, only the first packet from CN to MN goes through the HA (this avoids the triangle routing problem of mobile IPv4). Nevertheless, the use of MIPv6 as defined in [65] in 6LoWPAN networks is not straightforward due to the following problems:

- IPv6-in-IPv6 tunnels used to forward packets destined to MN when it is outside its home network increase the header overhead. Furthermore, the existing compression algorithms cannot compress the encapsulated IPv6 header and the transport header.
- IPsec is mandatory in MIPv6. The IPsec support in mobile nodes may be unreasonable for LoWPAN nodes.
- The MIPv6 signalization traffic increases the traffic overhead.
- MIPv6 route optimization adds complexity to mobile devices and requires memory to maintain the state of all active correspondent nodes.

There is a draft [69] that proposes message simplification and compression to support MIPv6 in 6LoWPAN networks. In this solution, the edge routers are responsible to compress and decompress the messages to be compliant with MIPv6 protocol. The draft expired in November 2009 and no follow-up version was proposed. Further work is needed to support MIPv6 in 6LoWPAN networks.

In the Proxy MIPv6 (PMIPv6) [70] solution, an IPv6 device can change its point of attachment without changing the IPv6 address or even implementing the MIPv6 protocol. PMIPv6 uses a local hierarchical structure of routers, usually edge routers, to handle mobility on behalf of mobile devices. The PMIPv6 solution is more appropriate to support node mobility in 6LoWPAN networks than the previous solutions. Nevertheless, there are two issues that still need to be solved: multi-hop communications cannot be used because PMIPv6 protocol only enables a node to communicate with its point of attachment and the 64 bit network prefix is mandatory for each mobile device.

Network mobility (NEMO) [63] is an extension of MIPv6 that enables mobile networks to attach to different points in the Internet. It also allows every node in the mobile network to be reachable while moving from one network to another. The mobile router, which connects the mobile network to the Internet, runs the NEMO basic support protocol with its home agent. The NEMO protocol is designed so that network mobility is transparent to the nodes inside the network. A number of

DOI: 10.1002/dac

different solutions were proposed to handle NEMO routing optimization [71]. NEMO seems to be a good solution when the entire LoWPAN domain, including the edge router, moves to a new point of attachment since it can run on edge routers which, usually, do not have the same limitations as LoWPAN nodes.

6. RESEARCH ISSUES SUMMARY

The term Internet of Things describes a vision in which networks and embedded devices are omnipresent in our lives and provide relevant content and information whatever the location of the user. IPv6 LoWPAN networks with mesh and mobility support will play a relevant role in realizing this vision. Although extensive efforts have been made, there are still many challenges that must be addressed in order to reach effective solutions. We summarize the identified challenges as follows:

Routing in IPv6 over LoWPAN mesh networks challenges: Three different approaches can be used to execute routing decisions: mesh-under, 6LoWPAN mesh-under and route-over. All approaches have benefits and there is no consensus on the best routing approach according to each network application scenario. Hence, it is important to define guidelines about the best suitable approach to the most used application scenarios.

In Section 4.1, several existing layer 2 routing protocols that can be used on mesh-under approach were identified. However, more research is required to evaluate the performance of such protocols in 6LoWPAN networks.

In the 6LoWPAN mesh-under routing approach, the routing decision occurs at the adaptation layer. 6LoWPAN mesh header was defined to support 6LoWPAN mesh-under routing. Nevertheless, actually there are no standard routing protocols that use 6LoWPAN mesh header.

Several multi-path standard routing protocols can be used on route-over approach, most of them adapted from MANET networks. Several efforts have been made to make those protocols suitable to LoWPAN network requirements. Reducing the number and the size of signaling messages and simplifying the routing algorithms remains a challenge. In parallel with these efforts, a new routing protocol is under discussion on the ROLL IETF working group. Efforts have been done to evaluate the new protocol in simulation environment for small outdoor and for large scale smart meter networks. Nevertheless, it is still necessary to prove its applicability on several network scenarios, particularly where mobility is a requirement.

Mobility in IPv6 over LoWPAN mesh networks challenges: Mobility can be classified as micro-mobility and macro-mobility. The design of a new protocol to deal with micro-mobility remains an open issue. Several solutions can be used to support node macro-mobility, all of them inspired by available solutions from standard IP networks. The adaptation of such solutions to 6LoWPAN network requirements and resources remain a challenge.

NEMO protocol can be used to support 6LoWPAN network mobility. A number of different solutions were proposed to handle NEMO routing optimization in regular IP networks. In spite of this, to apply these routing optimization techniques to LoWPAN networks it is necessary to reduce the protocol overheads, such as header overhead and signaling messages.

7. CONCLUSIONS

This paper has reviewed the state of art related with routing and mobility support in 6LoWPAN networks. Recently, the industry and the scientific community start to rethink many misconceptions about the use of IP in all LoWPAN nodes. The 6LoWPAN may be the convergence solution to connect the different physical and link layer protocols and enable to connect the LoWPAN devices to the Internet. Wireless Mesh support is considered the best-fitted routing topology for LoWPAN because of its ability to self-organize and self-configure. Mesh-under and router-over approaches can be used to support mesh routing on 6LoWPAN networks. Both have advantages and disadvantages. The standard routing protocols used on route-over approach were adapted from MANET routing protocols. Some adaptations are required to use these protocols in 6LoWPAN

DOI: 10.1002/dac

mesh networks. A new route-over routing protocol is under discussion on ROLL IETF working-group. Some solutions can be used to support LoWPAN node mobility and just one to support network mobility. For the time being, there is no single protocol, or even set of protocols, that can fulfill the routing and the mobility requirements to all LoWPAN scenarios. The design of a framework that addresses simultaneously routing and mobility requirements is crucial to achieve the Internet of Things vision.

ACKNOWLEDGEMENTS

Part of this work has been supported by *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal, in the framework of the Projects BodySens and Ecosense, and by the Euro-NF Network of Excellence of the Seventh Framework Program of EU, in the framework of the Project PADU.

REFERENCES

1. IEEE Std 802.15.4-2006. Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs). *IEEE Std. 802.15.4-2006*, 2006.
2. Karl H, Willig A. *Protocols and Architectures for Wireless Sensor Networks*. Wiley: New York, 2005.
3. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: a survey. *Computer Networks* 2002; **38**(4):393–422.
4. ZigBee Alliance, ZigBee Specification, October 2007.
5. Wireless HART homepage. Available from: http://www.hartcomm.org/ [February 2010].
6. ZigBee Alliance homepage. Available from: http://www.zigbee.org/ [January 2010].
7. Gershenfeld N, Krikorian R, Cohen D. The internet of things. *Scientific American* 2004; **291**(4):76–81.
8. Song J, Song H, Mok A, Chen D, Lucas M, Nixon M, Pratt W. Wireless HART: applying wireless technology in real-time industrial process control. *Proceedings of the 2008 IEEE Real-time and Embedded Technology and Applications Symposium*, St. Louis, United States, 2008; 377–386.
9. Commission of the European Communities. Internet of Things—An action plan for Europe, Communication from the commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the regions Webpage. Available from: http://www.ec.europa.eu/information_society/policy/rfid/documents/commiot2009.pdf [January 2010].
10. Dunkels A, Vasseur J. IP for smart objects alliance. *Internet Protocol for Smart Objects* (*IPSO*) *Alliance. White paper No. 2*, IPSO, September 2008.
11. IPv6 over Low power WPAN homepage, Available from: http://www.ietf.org/html.charters/6lowpan-charter.html, [February 2010].
12. Hui J, Culler D. Extending IP to low-power, Wireless personal area networks. *IEEE Internet Computing* 2008; **12**(4):37–45.
13. Al-Karaki N, Kamal A. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications* 2004; **11**(6):6–28.
14. Akkaya K, Younis M. A survey of routing protocols in wireless sensor networks. *Elsevier Ad Hoc Network Journal* 2005; **33**:325–349.
15. Kushalnagar N, Montenegro G, Schumacher C. IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals. *Internet Engineering Task Force*, Request for comments 4919, August 2007.
16. Bag G, Mukhtar H, Shams S, Kim K, Yoo S. Inter-PAN mobility support for 6LoWPAN. *Proceedings of 3rd International Conference on Convergence and Hybrid Information Technology* (*ICCIT '08*), Busan, Korea, 11–13 November 2008; 787–792.
17. Shelby Z, Bormann C. *6LoWPAN*: *The Wireless Embedded Internet*. Wiley: New York, 2009.
18. IPSO Alliance homepage. Available from: http://ipso-alliance.org [December 2009].
19. Internet of Things Strategic Research Roadmap. Available from: http://ec.europa.eu/information_society/policy/rfid/documents/in_cerp.pdf [January 2010].
20. Baronti P, Pillai P, Chook V, Chessa S, Gotta A, Hu Y. Wireless sensor networks: a survey on the state of the art and the 802.15.4 and ZigBee standards. *Computer Communications Wired/Wireless Internet Communications* 2007; **30**(7):1655–1695.
21. Muthukumaran P, Paz R, Spinar R, Pesch D. MeshMAC: enabling mesh networking over IEEE802.15.4 through distributed beacon scheduling. *Proceedings of Ad Hoc Networks*, *First International Conference* (*ADHOCNETS 2009*), ICST, 2009; 561–575.
22. The IEEE std. 802.15.5-2009: Part 15.5: Mesh Topology Capability in Wireless Personal Area Networks (WPANs). *IEEE Std. 802.15.5-2009*, 2009.
23. Zhang R, Park T, Lee M, Jung H, Ryu J. Testbed experimentation of a meshed tree routing with local link state for wireless PAN mesh. *Proceedings of IEEE International Conference on Communications* (*ICC'08*). IEEE: New York, 2008; 3060–3065.

DOI: 10.1002/dac

24. Mulligan G. The 6LoWPAN architecture. *Proceedings of the 4th Workshop on Embedded Networked Sensors* (*EmNets'07*). ACM: New York, 2007; 78–82.
25. Hui J, Culler D. IP is dead, Long live IP for wireless sensor networks. *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems* (*SenSys*). ACM: New York, 2008; 15–28.
26. Durvy M, Abeillé J, Wetterwald P, O'Flynn C, Leverett B, Gnoske E, Vidales M, Mulligan G, Tsiftes N, Finne N, Dunkels A. Making sensor networks IPv6 ready. *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*. ACM: New York, 2008; 421–422.
27. IETF 6LoWPAN working group homepage. Available from: http://tools.ietf.org/wg/6lowpan [January 2010].
28. Narten T, Nordmark E, Simpson W, Soliman H. Neighbor discovery for IP version 6 (IPv6). *Internet Engineering Task Force*, Request for comments 4861, September 2007.
29. Kushalnagar N, Montenegro G, Schumacher C. IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals. *Internet Engineering Task Force*, Request for comments 4919, August 2007.
30. Montenegro G, Kushalnagar N, Hui J, Culler D. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. *Internet Engineering Task Force*, Request for comments 4944, September 2007.
31. Hui J, Thubert P. Compression format for IPv6 datagrams in 6LoWPAN networks. *Internet Engineering Task Force*, draft draft-ietf-6lowpan-hc-06 working progress, October 2009.
32. Kim E, Kaspar D, Chevrollier N, Vasseur JP. Design and Application Spaces for 6LoWPANs. *Internet Engineering Task Force*, draft draft-ietf-6lowpan-usecases-05 working progress, November 2009.
33. Shelby Z, Thubert P, Hui J, Chakrabarti S, Bormann C, Nordmark E. 6LoWPAN neighbor discovery. *Internet Engineering Task Force*, IETF draft draft-ietf-6lowpan-nd-08 working progress, February 2010.
34. Kim E, Kaspar D, Gomez C, Bormann C. Problem statement and requirements for 6LoWPAN routing. *Internet Engineering Task Force*, draft draft-ietf-6lowpan-routing-requirements-04 working progress, July 2009.
35. Chowdhury A, Ikram H, Cha M, Redwan H, Shams H, Kim M, Yoo S. Route-over vs. mesh-under routing in 6LoWPAN. *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing* (*IWCMC '09*), Leipzig, Germany, 21–24 June 2009; 1208–1212.
36. Al-Karaki N. Analysis of routing security-energy trade-offs in wireless sensor networks. *Internet Journal of Secure Network* 2006; **1**(4):634–660.
37. Al-Karaki J, Kamal A. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications* 2004; **11**(6):6–28.
38. ISA 100, Wireless Systems for Automation homepage. Available from: http://www.isa100.org [October 2009].
39. Akkaya K, Younis H. A survey on routing protocols for wireless sensor networks. *Department of Computer Sciences and Electrical Engineering University of Maryland Annual ACM/IEEE* 2000; **30**(14–15):2826–2841.
40. Intanagonwiwat C, Govindan R, Estrin D. Directed diffusion: a scalable and robust communication paradigm for sensor networks. *Annual ACM/IEEE International Conference on Mobile Computing and Networking*, vol. 11(1). IEEE/ACM: New York, 2000; 2–16.
41. IETF ROLL Working Group homepage. Available from: http://tools.ietf.org/wg/roll [January 2010].
42. IETF MANET Working Group homepage. Available from: http://tools.ietf.org/wg/manet [January 2010].
43. Tarique M, Kemal E, Adibi S, Erfani S. Survey of multipath routing protocols for mobile ad hoc networks. *Journal of Network and Computer Applications* 2009; **32**(6):1125–1143.
44. Draves R, Padhye J, Zill B. Routing in multi-radio, multi-hop wireless mesh networks. *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*. ACM: New York, 2004; 114–128.
45. Mueller S, Tsang R, Ghosal D. *Multipath Routing in Mobile Ad Hoc Networks*: *Issues and Challenges*, Lecture Notes in Computer Science, vol. 2965. Springer: Berlin, 2004; 209–234.
46. Ramachandran K, Buddhikot M, Chandranmenon G, Miller S, Belding-Royer E, Almeroth K. On the design and implementation of infrastructure mesh networks. *Proceedings of the 1st IEEE Workshop on Wireless Mesh Networks* (*WiMesh '05*), Santa Clara, CA, U.S.A., 26 September 2005; 4–15.
47. Perkins C, Belding-Royer E, Das S. Ad hoc on-demand distance vector (AODV) routing. *Internet Engineering Task Force*, Request for Comments 3561, July 2003.
48. Johnson D, Maltz D, Hu Y, Jetcheva J. The dynamic source routing protocol for mobile ad hoc networks. *Internet Engineering Task Force*, Request for Comments 4728, February 2007.
49. Jacquet P. Optimized link state routing protocol. *Internet Engineering Task Force*, Request for Comments 3626, October 2003.
50. Bellur B, Lewis M, Ogier R, Templin F. Topology broadcast based on reverse-path forwarding (TBRPF). *Internet Engineering Task Force*, Request for Comments 3684, February 2004.
51. Chakeres I, Perkins C. Dynamic MANET on-demand (DYMO) routing. *Internet Engineering Task Force*, draft, draft-ietf-manet-dymo-21, July 2010.
52. Clausen T, Herberg U. Optimized link state routing protocol version 2. *Internet Engineering Task Force*, draft draft-ietf-manet-olsrv2-10, September 2009.
53. Jones H, Xu S, Blacktnore K. Link ratio for ad hoc networks in a Rayleigh fading channel. *Proceedings of the 3rd Workshop on the Internet*, *Telecommunications and Signal Processing* (*WITSP '04*), Adelaide, Australia, 20–22 December 2004; 252–255.
54. McDonald A, Znati T. A path availability model for wireless ad-hoc networks. *Proceedings of IEEE Wireless Communications and Networking Conference* (*WCNC '99*). IEEE: New York, 1999; 35–40.

55. Gomez C, Salvatella P, Alonso O, Paradells J. Adapting AODV for IEEE 802.15.4 mesh sensor networks: theoretical discussion and performance evaluation in a real environment. *Seventh IEEE International Symposium on a World of Wireless*, *Mobile and Multimedia Networks* (*WOWMOM 2006*). IEEE: New York, 2006; 159–170.
56. Levis P, Tavakoli A, Dawson-Haggerty S. Overview of existing routing protocols for low power and lossy networks. *Internet Engineering Task Force*, draft draft-levis-roll-protocols-survey-02, August 2008.
57. Dohler M. Routing requirements for urban low-power and lossy networks. *Internet Engineering Task Force*, Request for comments 5548, May 2009.
58. Dohler M, Watteyne T, Winter T, Barthel D. Routing requirements for urban low-power and lossy networks. *Internet Engineering Task Force*, Request for comments 5548, May 2009.
59. Kim M, Vasseur P, Chong H. Routing metrics used for path calculation in low power and lossy networks. *Internet Engineering Task Force*, draft draft-ietf-roll-routing-metrics-04, December 2009.
60. Vasseur P. Terminology in low power and lossy networks. *Internet Engineering Task Force*, draft draft-ietf-roll-terminology-02.txt, October 2009.
61. Tsao T, Alexander R, Dohler M, Daza V, Lozano A. A security framework for routing over low power and lossy networks. *Internet Engineering Task Force*, draft draft-tsao-roll-security-framework-01, September 2009.
62. Zahariadis T, Leligou H, Karkazis P, Trakadas P, Maniatis S. A trust framework for low power and lossy networks. *Internet Engineering Task Force*, draft draft-zahariad-roll-trust-framework-00, May 2009.
63. Devarapalli V, Wakikawa R, Petrescu A, Thubert P. Network mobility (NEMO) basic support protocol. *Internet Engineering Task Force*, Request for Comments 3963, January 2005.
64. Koodli S, Perkins C. *Mobile Inter-Networking with IPv6*: *Concepts*, *Principles*, *and Practices*. Wiley: New York, 2007.
65. Johnson D, Perkins C, Arkko J. Mobility Support in IPv6. *Internet Engineering Task Force*, Request for Comments 3775, June 2004.
66. Rosenberg J, Schulzrinne H, Camarillo G, Johnston A, Peterson J, Sparks R, Handley M, Schooler E. SIP: Session initiation protocol. *Internet Engineering Task Force*, Request for Comments 3261, June 2002.
67. Roychowdhury A, Gouran S. Motivation for SIP as an application protocol for 6lowpan devices. *Internet Engineering Task Force*, draft draft-roychowdhury-6lowappsip-00, October 2009.
68. Perkins C. IP mobility support. *Internet Engineering Task Force*, Request for Comments 3344, August 2002.
69. Silva R, Silva J. An adaptation model for mobile IPv6 support in lowPANs. *Internet Engineering Task Force*, draft draft-silva-6lowpan-mipv6-00, May 2009.
70. Gundavelli S, Leung K, Devarapalli V, Chowdhury K, Patil B. Proxy Mobile IPv6. *Internet Engineering Task Force*, Request for comments 5213, August 2008.
71. Shahriar M, Atiquzzaman M, Ivancic W. Route optimization in network mobility: Solutions, classification, comparison, and future research directions. *Communications Surveys and Tutorials*, *IEEE* 2010; **12**(1):24–38.

AUTHORS' BIOGRAPHIES

Luís Miguel L. Oliveira is a PhD student of Informatics Engineering at the University of Beira Interior under the supervision of Professor Joel Rodrigues and Professor Amaro de Sousa. He received his 5-year BS degree (licentiate) in Electronics from the University of Aveiro, Portugal, in 1998; and his MSc degree in Electronics and Telecommunications Engineering from the University of Aveiro, Portugal in 2004. He also teaches in the Informatics Engineering Department at the Superior School of Technology of the Polytechnic Institute of Tomar, Portugal. He is a PhD student member of the Instituto de Telecomunicações, Portugal. His current research areas are routing on wireless sensor mesh networks, Internet Protocol integration on wireless sensor networks and wireless sensor networks applications. He authors or co-authors more than ten international conference papers and also has three accepted journal publications. He has been acting as a reviewer for international journals and conferences.

Amaro F. de Sousa is an Assistant Professor since 2001 in the Department of Electronics, Telecommunications and Informatics of the University of Aveiro, Portugal, and Researcher since 1993 at the Instituto de Telecomunicações—pole of Aveiro, Portugal. He received his PhD degree in 2001 in Electrotecnics Engineering from the University of Aveiro, his MSc in 1991 in Telecommunications Engineering from the University College of North Wales, UK, and his 5-year BS degree in 1989 in Electronics and Telecommunications Engineering from University of Aveiro, Portugal. His main research interests are advanced services and protocols, traffic engineering of telecommunication networks and optimization algorithms for efficient network resource management. He has authored several papers in refereed international journals and conferences. He has been involved in different European Union funded and Portuguese funded projects over the last 15 years. He has been acting as a reviewer for many international journals and conferences.

Joel José P. C. Rodrigues is a Professor in the Department of Informatics of the University of Beira Interior, Covilhã, Portugal, and researcher at the Instituto de Telecomunicações, Portugal. He received a PhD degree in Informatics Engineering, an MSc degree from the University of Beira Interior, Portugal, and a 5-year BS degree (licentiate) in Informatics Engineering from the University of Coimbra, Portugal. He is the leader of the Next Generation Networks and Applications Group (NetGNA) at IT-UBI. His main research interests include sensor networks, eHealth and eLearning, delay-tolerant networks, high-speed networks, and mobile and ubiquitous computing. He is the Editor-in-Chief of the International Journal on E-Health and Medical Communications. He is the general Chair of the MAN 2009 and MAN 2010 (in conjunction with IEEE ICC 2009 and 2010), N&G 2010 (with IEEE AINA 2010), co-Chair of the Communications Software, Services and Multimedia Applications Symposium at IEEE Globecom 2010, co-Chair of the Selected Areas on Communications Symposium at IEEE ICC 2011, Chair of the Symposium on Ad-Hoc and Sensor Networks of the SoftCom Conference, TPC Chair of IEEE CAMAD 2010, and chaired many other technical committees. He is or was member of many international program committees (IEEE ICC, IEEE Globecom, IEEE WCNC, IEEE CCNC, IEEE ISCC, IEEE ICCCN, ICTTA, SoftCOM, etc.) and several editorial review boards (IEEE Communications Magazine, Journal of Communications Software and Systems, International Journal of Communications Systems, International Journal of Business Data Communications and Networking, etc.), and he has served as a guest editor for a number of journals including the Journal of Communications Software and Systems. He chaired many technical sessions and gave tutorials at major international conferences. He has authored or co-authored over 150 papers in refereed international journals and conferences, a book and a patent pending. He is a licensed Professional Engineer and a member of the ACM SIGCOMM, a member of the Internet Society, IARIA Fellow, and a Senior Member of the IEEE.

DOI: 10.1002/dac

Chapter 3

Wireless Sensor Networks: a Survey on Environmental Monitoring

This chapter consists of the following paper:

Wireless Sensor Networks: a Survey on Environmental Monitoring

L. Oliveira and J. Rodrigues
Journal of Communications, vol. 6, no. 2, pp. 143-151, 2011.

DOI: 10.4304/jcm.6.2.143-151

According to Journal Citation Reports published by Thomson Reuters, this journal scored ISI journal performance metrics as follows:

ISI Impact factor (2011): N/A

Article Influence Score (2011): N/A

JOURNAL OF COMMUNICATIONS, VOL. 6, NO. 2, APRIL 2011 143

Wireless Sensor Networks: a Survey on Environmental Monitoring

Luís M. L. Oliveira
Instituto de Telecomunicações, University of Beira Interior, Portugal
Polytechnic Institute of Tomar, Tomar, Portugal
Email: loliveira@it.ubi.pt

Joel J. P. C. Rodrigues
Instituto de Telecomunicações, University of Beira Interior, Portugal
Email: joeljr@ieee.org

***Abstract*— Traditionally, environmental monitoring is achieved by a small number of expensive and high precision sensing unities. Collected data are retrieved directly from the equipment at the end of the experiment and after the unit is recovered. The implementation of a wireless sensor network provides an alternative solution by deploying a larger number of disposable sensor nodes. Nodes are equipped with sensors with less precision, however, the network as a whole provides better spatial resolution of the area and the users can have access to the data immediately. This paper surveys a comprehensive review of the available solutions to support wireless sensor network environmental monitoring applications.**

***Index Terms*— Wireless Sensor Networks; Low-Power Personal Area Networks; Mesh Networks; IEEE 802.15.4; Environment Monitoring.**

I. INTRODUCTION

Environmental monitoring has a long history. In early times analog mechanisms were used to measure physical environmental parameters. Some of them with the ability to record the values on paper dish. The old mechanisms recorded data at specific intervals and required human intervention to download them.

Some years ago, digital data loggers have replaced the old mechanical. The digital data loggers are more easy to operate and to maintain and more cheaper than the old mechanisms. Digital data loggers may also be combined with long-range communication networks, such as GSM, to retrieve data from remote sites. However, digital data loggers have some drawbacks. The digital data loggers solution, usually provide monitoring at one point only and in many cases multiple points need to be monitored. There is not a standard to store data and to communicate with the data logger, so several different solutions are used.

Recent advances in micro-electro-mechanical systems and in low-power wireless network technology have created the technical conditions to build multi-functional tiny sensor devices, which can be used to observe and to react according to physical phenomena of their surrounding environment [1]. Wireless sensor nodes are low-power devices equipped with processor, storage, a power supply, a transceiver, one or more sensors and, in some cases, with an actuator. Several types of sensors can be attached to wireless sensor nodes, such as chemical, optical, thermal and biological. These wireless sensor devices are small and they are cheaper than the regular sensor devices.

The wireless sensor devices can automatically organize themselves to form an ad-hoc multi hop network. Wireless sensor networks (WSNs), may be comprised by hundreds or maybe thousands of ad-hoc sensor node devices, working together to accomplish a common task. Self-organizing, self-optimizing and fault-tolerant are the main characteristics of this type of network [2]. Widespread networks of inexpensive wireless sensor devices offer a substantial opportunity to monitor more accurately the surrounding physical phenomena's when compared to traditional sensing methods [3]. Wireless sensor network has it own design and resource constrains [4]. Design constrains are related with the purpose and the characteristics of the installation environment. The environment determines the size of the network, the deployment method and the network topology. Resources constrains are imposed by the limited amount of energy, small communication range, low throughput and reduced storage and computing resources. Research efforts have been done to address the above constrains by introducing new design methodologies and creating or improve existing protocols and applications [1,2].

This paper provides a review on wireless sensor networks solutions to environmental monitoring applications. The remainder of this paper is organized as follows. Section II gives an overview of sensor network platforms. Section III analyses the standard IEEE 802.15.4 [5] while Section IV overviews recent sensor architectures. WSN environmental monitoring projects are presented in Section V and challenges related with environment sensor networks are studied in Section VI. Section VI concludes the paper and addresses future research challenges related to WSN networks deployment.

Manuscript received August 15, 2010; revised November 15, 2010; accepted January 15, 2011.

doi:10.4304/jcm.6.2.143-151

II. Sensor Network Platforms

Sensor nodes are the elementary components of any WSN and they provide the following basic functionalities [1-2,7]: *i*) signal conditioning and data acquisition for different sensors; *ii*) temporary storage of the acquired data; *iii*) data processing; *iv*) analysis of the processed data for diagnosis and, potentially, alert generation; *v*) self-monitoring (e.g., supply voltage); *vi*) scheduling and execution of the measurement tasks; *vii*) management of the sensor node configuration; *viii*) reception, transmission, and forwarding of data packets; and *ix*) coordination and management of communications and networking.

To provide the above-described functionalities, as illustrated in Figure 1, a sensor node is composed by one or more sensors, a signal conditioning unit, an analog-to-digital conversion module (ADC), a central processing unit (CPU), memory, a radio transceiver and an energy power supply unit. Depending on the deployment environment, it can be necessary to protect the sensor hardware against mechanical and chemical aggressions with an appropriate package.

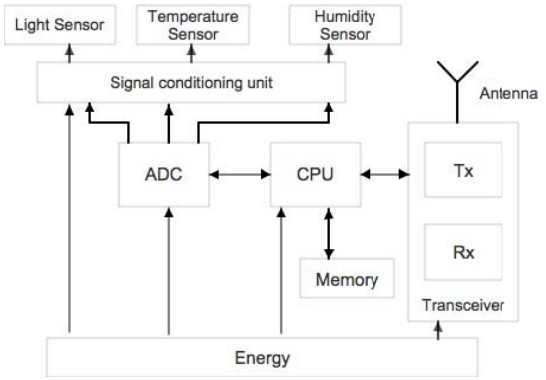


Figure 1 - Sensor node hardware architecture.

Sensor node hardware devices can be classified into three different main categories [8]:

• Adapted general-purposes computers. This sensor platform uses hardware similar to embedded personal computers hardware, personal assistants devices and low-power personal computer devices. Windows and Linux are the mainly used operating systems. High level programming languages can be used to develop software components. Usually supports simultaneous layer two low power protocols and layer two local area protocols. Processing capabilities, multiple layer two protocol support and versatility are the main advantages of this hardware platform. However, they consume a considerable amount of energy when compared with other hardware platforms. Adapted general-purposes computer platform are usually used as a gateway to connect the wireless sensor network to other networks.

• Embedded sensor modules. This sensor hardware platform uses commercial off-the-shelf (COTS) chips. These platforms are cheaper than the previous because COTS chips are produced in large scale. A microcontroller unit (MCU) is used as central processing unit. The C programming language is usually used to program the platform, enabling the development of thigh code that fits in their limited memory size.

• System on chip. This platform uses application specific integrated circuits (ASIC), which integrate all sensor hardware components. Because of this integration, systems on chip platforms are extreme low power, cheap and small size.

Hardware management, scheduling policies, multi-threading and multitasking are some of the low level services to be provided by an operating system (OS). Moreover, the operating system should also provide the support for dynamic loading and unloading of modules, provide proper concurrency mechanisms, Application Programming Interface (API) to access underlying hardware and enforce proper power management policies. The achievement of those services in WSN is a non-trivial problem, due to the hardware constrains [9]. A classification framework that compares the existing operating systems according to the core OS is proposed on [9]. The core OS features that constitute classification framework are architecture, reprogramming, execution model and scheduling. Other features such as power management, simulation support, and portability also has been considered. The proposed framework was used to compare and evaluate the existing operating systems. The operating systems were also evaluated according to WSN application. TinyOS [10] and Contiki [11] are the most used operating systems.

Several energy storage devices are available. Battery is the most common energy storage device. Fuel cells and ultracapacitors are presented as promising technologies. Energy harvesting techniques can be used to increase the sensor energy autonomy [12]. Energy harvesting schemes developed in the laboratory have generated 10 μW of power from mechanical vibrations [13]. This energy is enough for low-frequency digital signal processor. Advances in energy harvesting and improvements in node integration will make possible to produce a battery less infinite-lifetime sensor device. Wireless data transmission consumes more energy than data processing. So it is preferable to process the data at the sensor in order to minimize the data transmitted to the other nodes. The power consumed when the radio is in receive mode is almost equal to that consumed when it is transmitting [7]. So, the radio must be turned off when it is not required. Moreover, sensor nodes must take advantage of long periods of idle time between interesting events to save energy. In the inactivity periods, the sensor cans gracefully scaling back their energy consumption. So, it is crucial defining the network's performance requirements using metrics ranging from latency to accuracy and reliability. Then, the network performs just enough data computation, and data receptions and transmissions to meet the WSN application requirements. Turning off the sensor poses the problem of how neighboring nodes can be organize to wakeup at the same time to communicate. Several approaches were proposed to address this problem, such as [14] and [15].

III. IEEE 802.15.4 Overview

The standard IEEE 802.15.4 [5] released in 2003, represented a millstone because it was the first low-power layer two standard for low power wireless personal area network (LoWPAN). Several technologies have been specified using IEEE 802.15.4 as link layer technology, some of them proprietary, such as ZigBee [16] and WirelessHART [17]. The ZigBee was created by ZigBee alliance and defines the network, security and application layers. The ZigBee alliance also publishes application profiles that allow multiple vendors to create interoperate products. The WirelessHART is an open-standard wireless networking technology proposed by HART Communication Foundation and it is also based in IEEE 802.15.4. It is mainly used in industrial environments. WirelessHART, like ZigBee, is a stand-alone standard; consequently do not support communications with other networks without using a specific gateway device.

IEEE 802.15.4 physical layer provides an interface between the medium access control (MAC) sub-layer and the physical radio channel. Two services are provided, the physical data service and the physical management service. The physical layer is responsible for the following tasks: i) activation and deactivation of the radio transceiver, ii) energy detection (ED) sensed on the current channel, iii) clear channel assessment (CCA) for CSMA/CA, iv) channel frequency selection, v) link quality indication (LQI) for received packets and vi) data transmission and reception.

The physical layer is responsible to turn the radio transceiver into one of the three states, that is, transmitting, receiving, or sleeping (equivalent to turn off the radio transceiver) according to the information returned by MAC sub-layer.

Energy detection (ED) sensed on the current channel is executed by physical layer and is an estimate of the received signal power of an IEEE 802.15.4 channel. No attempt is made to identify or decode signals on the channel in this procedure. The result from energy detection can be used as part of a channel selection algorithm or for the purpose of clear channel assessment (CCA).

The physical layer performs CCA using energy detection, carrier sense or a combination of both. In energy detection mode, the medium is considered busy if any energy above a predefined energy threshold is detected. In carrier sense mode, the medium is considered busy if a signal compatible with IEEE 802.15.4 is detected. In the combined mode, both conditions above-mentioned must occur in order to conclude that the medium is busy.

Wireless links under IEEE 802.15.4 can operate in 27 different channels. So, the physical layer should be able to adjust its transceiver into a certain channel according with the information received from the MAC sub-layer.

Link quality indication (LQI) measurement is performed by the physical layer for each received packet. The physical layer uses energy detection function, a signal-to-noise ratio or a combination of these to measure the strength and/or quality of a link from which a packet is received.

Modulation and spreading techniques are used to transmit the data over radio channel. Data reception is also a physical layer function.

The IEEE 802.15.4 defines the following three physical operation modes: 20 kbps at 868 MHz, 40 kbps at 915 MHz, and 250 kbps at 2.4 GHz (DSSS).

A device in an IEEE 802.15.4 network can use either a 64-bit address or a 16-bit short IEEE address assigned during the association procedure. An 802.15.4 network can accommodate up to 64k (2^{16}) devices.

The frame length is limited to 127 bytes because low-power wireless links are used in communications and the sensors have limited buffering capabilities.

The IEEE 802.15.4 define the following two types of devices; full-function devices (FFD) and reduced-function devices (RFD). In FFD all network functionalities are implemented and therefore can be used in peer-to-peer topologies and multi-hop communications are supported. Reduced-function devices only support a limited set of functionalities and they are used to measure physical parameters and to execute uncomplicated tasks. An RDF device does not support multi-hop communications.

FFD and RFD devices organize themselves in personal area network (PAN). A PAN is controlled by a PAN coordinator, which has the function of setting up and maintaining the network. Only FFD devices can assume the role of PAN coordinator.

The MAC sub-layer provides an interface between the service specific convergence sub-layer and the physical layer. Like the physical layer, the MAC sub-layer also provides two services, namely, the MAC data service and the MAC management service. The MAC sub-layer is responsible for the following tasks: i) support PAN node association and disassociation, ii) transmit network beacons if the device is a PAN coordinator; iii) synchronize to the beacons, iv) use carrier sense multiple access with collision avoidance (CSMA/CA) mechanism for channel access, v) support the guaranteed time slot (GTS) mechanism and vi) provide a reliable link between two peer MAC entities.

To support self-configuration, IEEE 802.15.4 embeds association and disassociation functions in its MAC sub-layer. This not only enables a star to be setup automatically, but also allows the creation of self-configuring peer-to-peer network topologies.

A coordinator must determine if the beacon-enabled mode is required, in which a superframe structure is used. In the beacon-enabled mode, a coordinator sends out beacons periodically to synchronize the other PAN nodes. A device attached to a coordinator operating in a beacon-enabled mode must track the beacons to be synchronized with their PAN coordinator. This synchronization is important for data polling and for energy saving purposes.

The IEEE 802.15.4 MAC provides two modes of operation, the asynchronous beaconless and the synchronous beacon enabled mode. The beaconless mode requires nodes to listen for other nodes transmission all the time and as a consequence drains the battery power fast. The beacon-enabled mode is designed to support the transmission of beacon packets between transmitter and receiver providing synchronization among nodes. In the beacon-enabled mode, the PAN coordinator broadcasts a periodic beacon containing information about the PAN. Synchronization provided by the beacons allows devices to sleep between transmissions, which result in energy efficiency and extended battery lifetime. Supporting beacon-enabled mode in peer-to-peer topologies is currently considered a challenge.

The period between two consecutives beacons defines a superframe structure that is divided in to 16 slots. Beacon always occupies the first slot, while the others is used to data communication. In order to support low-latency applications, the PAN coordinator can reserve one or more slots, designated by guaranteed time slots, which are assigned to devices running such applications. These devices do not need to use contention mechanisms before transmit. The beaconless mode doesn't permit superframe structures, so guaranteed time slots cannot be reserved. As a consequence, only random access methods, such as unslotted CSMA/CA can be used to medium access. The IEEE 802.15.4 CSMA/CA does not include the request-to-send (RTS) and clear-to-send (CTS) mechanism, because low data rate is used.

The MAC sublayer employs various mechanisms to enhance the reliability of the link between two peers, among them there are the frame acknowledgment and retransmission, data verification by using a 16-bit CRC, as well as CSMA/CA.

A PAN can adopt one of the following two network topologies [18]: star topology and peer-to-peer topology.

In a star topology a master-slave network model is used (Figure 2). An FFD device assumes the PAN coordinator role and controls all the networks operations. Other nodes can be RFDs or FFD and communicates only with PAN coordinator. This topology is better suited for small networks. In this configuration the PAN coordinator

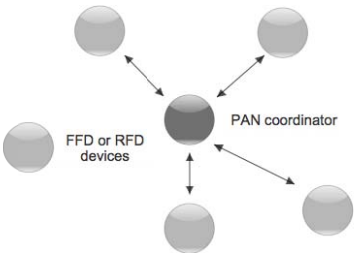


Figure 2 – Illustration of a star topology.

In peer-to-peer topology FFD devices can communicate with other FFDs within its radio range and can use multi-hop communications to send messages to other FFDs outside of its radio range. RFDs can communicate only with FFDs (Figure 3).

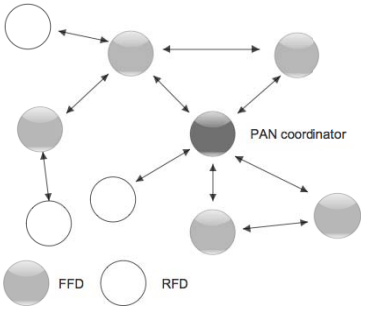


Figure 3 - Illustration of a peer-to-peer topology.

Peer-to-peer topology supports more complex topologies, such as mesh or hierarchical cluster. A mesh network topology is a PAN that uses one of two connection configurations: full mesh topology or partial mesh topology. In the full mesh topology, each node is connected directly to each of the others. In the partial mesh topology, some nodes are connected to all the others, but some of the nodes are connected only to limited number of nodes. When compared to star topologies, mesh networks have the capability to provide extension of network coverage without increasing transmit power or receive sensitivity, better reliability via route redundancy, easier network configuration and better device battery life, due to fewer retransmissions. As IEEE 802.15.4 does not define any path selection mechanism, the IEEE 802.15.5 [6], also known as mesh WPAN, was chartered in November 2003 to develop the necessary mechanisms that must be present in physical and medium access control layers of WPANs to enable mesh networking. The work of the IEEE 802.15.5 group covers both high-rate and low-rate WPANs. So, the outcome of this work group is applicable on IEEE 802.15.3 (high rate PAN) and on IEEE 802.15.4 protocols.

Many routing protocols have been specifically designed for WSNs [19] where energy awareness is an essential design issue. Routing protocols in WSN can be classified from the perspective of network structure in three different classes, flat based, hierarchical based and location based. In flat based routing, all network devices have the same roles in the routing topology. In hierarchical based routing, nodes can play different roles. Nodes with higher resources can be used in multi-hop forwarding and the other nodes can be uses in sensing functions. In location based routing, sensors are addressed according to their location, and data are routed using node positions. The routing mechanisms must take in consideration the network purpose and the architecture requirements.

IV. Overview of Recent Sensor Architectures

Reduced instruction set computer (RISC) microcontrollers with a small program and data memory size are used on low-end and low-cost sensors devices.

Additionally, an external flash memory can be used to provide secondary storage. Two approaches have been adopted for the design of sensing equipment [7]. The first approach uses a sensing board that can be attached to the main microcontroller board through an expansion bus. Usually, more than one can be attached. This is the most expandable approach and can be found on Iris platform [20]. A typical crossbow sensing board provides light, temperature, microphone and two-axis accelerometer device. Other boards only have I/O connectors and can be used to connect custom sensor to the main board. In the second approach, the main board also includes the sensing devices. The sensing devices are soldered or can be mounted if needed. The expandability is affected because the available sensing devices options are very limited. The second approach can be used to reduce the production costs. TelosB [21] vendor follow the second approach.

Currently, the most popular sensors platforms employ one of two type radios designed by Chipcon [22], the CC1000 and the CC2420. The CC1000 is the simpler and the cheaper alternative. It offers a basic medium access control protocol, operates in a license free band (315/433/868/915 MHz) and has a bandwidth in the range 20-50 Kbps. It has a simple byte oriented interface that allows software implementation of other MAC protocols. The CC2420 is compliant with IEEE 802.15.4 specification, operate at 2.4 GHz license free band and has 250Kbps bandwidth.

There are two popular microcontrollers used on WSN platforms, the ATMega 128L [23] and Texas Instruments MSP430 [24]. The ATMega 128L has 128KB of code memory and 4KB of data storage. The MSP430 has 48KB of code memory and 10KB for data storage.

An exhaustive list of sensor boards, vendors and their main characteristics are presented in [25].

Currently available sensor platforms mainly use two size AA battery cells. Standard batteries are cheaper and easy to replace. However they limit the platform size reduction.

TinyOS and Contiki are the most used open source and freeware WSN operating systems [9]. TinyOS is an event driven operating systemand it uses a C-like programming language (NesC), although incompatible with C standard, which has a very low memory footprint. Commands and event handlers may post a task, which is executed by the TinyOS first-in first-out (FIFO) scheduler. These tasks are non preemptive and run to completion. TinyOS supports power management functions. TinyOS is gaining its importance in the WSN applications and has been ported to different platforms. Although popular, TinyOS has some drawbacks, namely the lack of supporting fault tolerance, preemptive multitask, priority scheduling, dynamic programming, and real time grantees. Contiki OS merges the advantages of both events and threads execution models. It is primarily an event driven model but it also supports optionally preemptive multi-threading as an optional application level library. Events in Contiki OS are classified as synchronous and asynchronous. Synchronous events are scheduled immediately and asynchronous events are scheduled afterward. In Contiki, everything such as communications, device drivers and sensors data handling are supported as a service and each service has an interface and implementation. Contiki OS also has support for dynamic loading and replacement of individual programs and services in runtime. Applications are developed using C++ standard language. There are simulation tools to both operating systems. TOSSIM [26] simulates TinyOS applications for sensor network and Cooja [27] is a simulation environment for Contiki OS. Both operating systems have support for IPv4 and IPv6 protocols.

V. WSN ENVIRONMENTAL MONITORING

Environment monitoring is a natural candidate for applying wireless sensor networks, since the physical variables that must to be monitored, e.g., temperature. They are usually distributed over large regions.

Environmental monitoring applications can be broadly categorized into indoor and outdoor monitoring [28]. Indoor monitoring applications typically include buildings and offices monitoring. These applications involve sensing temperature, light, humidity, and air quality. Other important indoor applications may include fire and civil structures deformations detection. Outdoor monitoring applications include chemical hazardous detection, habitat monitoring, traffic monitoring, earthquake detection, volcano eruption, flooding detection and weather forecasting. Sensor nodes also have found their applicability in agriculture. Soil moisture and temperature monitoring is one of the most important application of WSNs in agriculture. Only outdoor environmental monitoring will be considered in this work.

When monitoring the environment, it is not sufficient to have only technological knowledge about WSN and their protocols. It is also necessary the knowledge about the ecosystem.

Several projects, with real implementations, had focused on environmental sensor networks; some of them are presented bellow.

GreatDuckIsland [29] was the first WSN implemented for habitat monitoring purposes. College of Atlantic and Berkeley University conducts field research on several remote islands. One of them, Great Duck Island (GDI) is located 15Km south of Mount Desert Island, Main. Studying the usage pattern of the nesting burrows when one or both parents alternate between incubation and feeding is the major objective of this project. A single hop hierarchical network comprises 32 nodes in the first phase and 120 in the last were set up at GDI. Berkeley Mica sensor nodes with TinyOS installed were used to measure temperature, humidity and atmosphere pressure and to detect the presence of the birds. Readings from sensor nodes are periodically sampled and relayed from the local sink node to base station on the island. The base station sends the data using a satellite link to a server connected to the Internet.

Sonoma Dust [30] is a WSN, constituted by 120 Mica2dot nodes that were installed on Sonoma County, California to monitor the redwood trees habitat conditions. Nodes with TinyOS were programmed to measure the environmental conditions (temperature, humidity and photo-synthetically active radiation) every 5 minutes and forwarded them through a multi-hop mesh network to a local base station. The data is sent from the base station to a computer located 70 Km away, through radio links. The nodes were programmed to run at a very low duty cycle to save energy.

A wireless sensor network was deployed to monitor eruptions at Tungurahua volcano, located in central Ecuador [31]. This single hop network is constituted by five sensor nodes where three of them are equipped with a specially constructed microphone to monitor infrasonic signals originated by volcanic eruptions. The data collected by the sensors are sent to a local sink and then relayed over radio links to a computer located 9 Km away. Mica2 nodes with TinyOS were used.

Measurement the microclimate in potato crops is the main goal of Lofar agro project [32]. The collected information will be used to improve the advice on how to combat phytophtora within a crop, based on the circumstances within each individual field. Phytophthora is a fungal disease in potatoes, their development and associated attack of the crop depends strongly on the climatologically conditions within the field. A total of 150 sensor nodes, similar to the Mica2 motes, were installed in a parcel for crop monitoring. Nodes are manually localized and their location registered on a map. Sensor nodes are equipped with sensors for registering the temperature and relative humidity. In addition to the sensor nodes, the field is equipped with a weather station to register the luminosity, air pressure, precipitation, wind strength, and direction. The sensor nodes use TinyOS operating system. The data collected by the sensor nodes is sended over a multi-hop routing protocol to the local sink node (field gateway) and further transferred via Wi-Fi to Lofar gateway. The Lofar gateway is connected via wire to the Internet and data is uploaded to a Lofar server and further distributed to a couple of other servers.

In SECOAS project [33] a sensor network was deployed at Scroby sands off the coast of Great Yarmouth and its purpose will be to monitor the impact of a newly developed wind farm on coastal processes in the area. New sensor hardware, based on MCU PIC 18F452 was developed in this project and a new operating system, designated by kOS (kind-of operating system) was proposed to run on it. The sensor nodes are equipped with sensors for registering the pressure, turbidity, temperature and salinity. Sensor nodes, base stations on the sea and land stations, form the hierarchical and single hop network. Nodes transmit their data to the sea base stations, which will then transmit the data to the land station. Base stations are sensor nodes equipped with additional functionalities, more power supplies and larger communication range. The data accessed from the land station via Internet.

Foxhouse [34] get real time information about the habitat of foxes in a fox house. A wireless sensor network in the Foxhouse case has 14 nodes organized in two clusters. The network uses FFD nodes to relay data and RFD nodes for sensing. The sink node is connected to a personal computer where data is stored. CiNet boards compliant with IEEE 802.15.4 and based on ATmega 128L MCU are used on sensing nodes. The sensing nodes are equipped with temperature, humidity and light sensors.

In Sensorscope project [35], two networks were deployed. The first network was installed in Wannengrat to study environmental processes involving snow. The second network was installed on a glacier in the canton Valais, Switzerland, to measure air temperature, air humidity, surface temperature, wind direction and speed, precipitation and solar radiation. Seven nodes were used in the first deployment and sixteen nodes in the second. The similar solutions were used on both deployments. A Shockfish TinyNode platform was chosen and it is composed by a Texas Instruments MSP430 MCU and a Semtech XE1205 radio transceiver, operating in the 868 MHz band. The sensing nodes and the sink node uses TinyOS operating system. A multi-hop network is used to support communications between the sink node and the sensing nodes. Sensing stations regularly transmit collected data (e.g., wind speed and direction) to a sink, which, in turn, uses a gateway to relay the data to a server. GPRS, Wi-Fi or Ethernet technologies can be used to connect the sink node to the data base server, which can be installed remotely. Data is published on a real-time Google Maps-based web interface and on Microsoft's SensorMap website.

VI. Challenges for Environmental Sensor Networks

The term Internet of Things [36] describes a vision in which networks and embedded devices are omnipresent in our lives and provide relevant content and information whatever the user location. Sensors and actuators will play a relevant role to accomplish this vision. Although, extensive efforts have been done to achieve the Internet of Things vision, there still some challenges that need to be addressed. The most relevant are presented bellow.

Power management. This is essential for long-term operation, especially when it is needed to monitoring remote and hostile environments. Harvesting schemes, cross-layer protocols and new power storage devices are presented as possible solutions to increase the sensors lifetime.

Scalability. A wireless sensor network can accommodate thousands nodes. Current real WSN for environment proposes the use of tens to hundreds nodes. So it is necessary to prove that the available theoretical solutions are suited to large real WSN.

Remote management. Systems installed on isolated locations cannot be visited regularly, so a remote access standard protocol is necessary to operate, to manage, to reprogramming and to configure the WSN, regardless of manufacturer.

Usability. The WSNs are to be deployed by users who buy them off the shelf. So, the WSN need to become easier to install, maintain and understand. It is necessary to propose new plug and play mechanisms and to develop more software modules with more user-friendly interface.

Standardization. The IEEE 802.15.4 represents a millstone in standardization efforts. Although, compatibility between of-the-shelf modules is in practice very low. It is important to specify standard interfaces to allow interoperability between different modules vendors in order to reduce the costs and to increase the available options.

Mesh routing support. The mesh networks topologies can both provide multi-hop and path diversity [40]. So, a routing protocol to support multi-hop mesh network is crucial [37], which must take into account the very limited features of the network.

Size. Reducing the size is essential for many applications. Battery size and radio power requirements play an important role in size reduction. The production of platforms compatible with the smart dust can be determinant in WSN environmental monitoring.

IP end-to-end connectivity. Originally it was not thought appropriate the use of IP protocol in WSN networks, because of the perception that is was to heavy weight to the WSN nodes resources. Recently, the industry and the scientific community start to rethink many misconceptions about the use of IP in all WSN nodes [38]. Supporting IPv6 on sensor nodes simplifies the task of connecting WSN devices to the Internet and creates the conditions to realize the paradigm of Internet of Things community. Additionally, by using IPv6 based protocols, users can deploy tools already developed for commissioning, configuring, managing and debugging these networks [37]. The application developing process is also simplified and open.

Price. Available sensor platforms on the market are expensive which precludes its use widely. Produce cheaper and disposable sensor platforms it is also a challenge.

Support other transducers types. Environmental monitoring usually uses limited type of transducers, such as temperature, light, humidity and atmospheric pressure. New environmental monitoring applications will be developed and new transducers will be necessary to measure new physical phenomena, for example image and video. Transmit images and video on resources and power constrained networks are a challenge [39].

The identified challenges must be addressed simultaneously by scientific community and by industry to create successful commercial solutions.

VII. Conclusions and Future Work

In this research work, a survey on environmental monitoring using wireless sensor networks and their technologies and standards was carried out. Some of the most relevant environmental monitoring projects with real deployments were analyzed and the conclusions used to identify the challenges that need to be addressed.

Wireless sensor networks continue to emerge as a technology that will transform the way we measure, understand and manage the natural environment. For the first time, data of different types and places can be merged together and accessed from anywhere. Some significant progress has been made over the last few years in order to bridge the gap between theoretical developments and real deployments, although available design methodologies and solutions are still relatively immature. As a consequence, widespread use of WSNs for environmental proposes is not yet a reality.

It is predictable that in the near future any object will have an Internet connection – this is the Internet of Things vision. In smart cities, the environmental data will provide usefully information to the citizens. For example, air quality, transportation information, emergency services, and so on. The citizens can access to this information via Internet.

Nowadays, the IP suite protocol support in environmental monitoring is inconsistent. It is necessary design new protocols and evaluates the existing ones. Assess the major benefits associated with the support of the IP protocol on all nodes, using simulation and testbeds is fundamental. This evaluation will be addressed as a future work.

Acknowledgments

Part of this work has been supported by the *Instituto de Telecomunicações*, Next generation Networks and Applications Group (NetGNA), Portugal, in framework of EcoSense Project.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey”, Computer Networks, Vol. 52, Issue 12, , pp. 2292-2330, August 2008

[2] IF. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” Computer Networks; vol. 38, issue 4, pp. 393–422, 2002.

[3] R. Cardell-Oliver, M. Kranz, K. Smettem, and K. Mayer, “A Reactive Soil Moisture Sensor Network: Design and Field Evaluation,” International Journal of Distributed Sensor Networks, vol. 1, pp. 149-162, 2005.

[4] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, “The hitchhiker's guide to successful wireless sensor network deployments,” Proceedings of the 6th ACM conference on Embedded network sensor systems, 2008, Raleigh, NC.

[5] IEEE Std 802.15.4-2006. Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs). IEEE Std 802.15.4-2006, 2006.

[6] The IEEE std. 802.15.5-2009: Part 15.5: Mesh Topology Capability in Wireless Personal Area Networks (WPANs) IEEE Std802.15.5-2009, 2009.

[7] P. Baronti, P. Pillai, V. Chook, S. Chessa, A. Gotta, and Y. Hu, “Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards,” Computer Communications, vol. 30, issue 7, Wired/Wireless Internet Communications, pp. 1655-1695, May 2007.

[8] F Zhao, and L. Guibas, Wireless Sensor Networks: An Information Processing Approach, Morgan Kaufmann: San Francisco, pp. 240–245, 2004.
[9] A. Reddy, P. Kumar, D. Janakiram, and G. Kumar, "Wireless sensor network operating systems: a survey," Int. J. Sen. Netw, vol. 5, issue 4, pp. 236-255, 2009.
[10] TinyOS, http://www.tinyos.net/, [Jully 2010].
[11] Contiki, http://www.sics.se/contiki/, [Jully 2010].
[12] F. Garcia-Hernandez, P. Ibarguengoytia-Gonzalez, and A. Perez-Diaz, "Wireless Sensor Networks and Applications: A Survey", IJCSNS International Journal of Computer Science and Network Security, vol. 7, issue 3, pp. 264-273, 2007.
[13] M. Pereyma, "Overview of the modern state of the vibration energy harvesting devices," Proc. IEEE MEMSTECH, 2007, pp. 107-112.
[14] B. Chen, K. Jamieson, H. Balakrishnan and R. Morris, "Span: an energy efficient coordination algorithm for topology maintenance in ad hoc wireless networks," ACM Wireless Networks Journal, vol. 8, issue 5, pp. 481–494, 2002.
[15] Y. Xu, J. Heidemann, and D. Estrin, "Geography informed energy conservation for ad hoc routing," Proceedings of the 7th International Conference on Mobile Computing and Networking (MobiCom 2001), Italy, July 2001, pp. 70–84.
[16] ZigBee Alliance, Network Layer Specification 1.0, Dec. 2004.
[17] WirelessHART Webpage. Available from: http://www.hartcomm.org/, [Jully 2010].
[18] L. Nardis, and M. Benedetto, "Overview of the IEEE 802.15.4/4a standards for low data rate Wireless Personal Data Networks", Positioning, Navigation and Communication, pp. 285-289, 2007.
[19] J. Al-Karaki, and A Kamal, "Routing techniques in wireless sensor networks: A survey", IEEE Wireless Commun. Mag., vol. 11, no. 6, 2004, pp. 6-28.
[20] Iris sensor platform, http://www.xbow.com/Products/productdetails.aspx?sid=264, [Jully 2010].
[21] TelosB sensor platform, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf, [Jully 2010].
[22] Chipcon webpage, http://www.chipcon.com, [Jully 2010].
[23] Atmel Corporation. 2004. 8-bit AVR microcontroller with 128K bytes in-system programmable flash, http://www.atmel.com/dyn/resources/prod_documents/2549S.pdf, [Jully 2010].
[24] Texas Instrument. MSP43OX4XX Family, User Guide, Dallas, Texas, 2003.
[25] I. Akyildiz, and M. Vuran, Wireless Sensor Networks, New York: John Wiley & Sons, 2010.
[26] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: accurate and scalable simulation of entire tinyos applications," in SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems. New York, NY, USA: ACM, 2003, pp. 126–137.
[27] F. O'sterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," In Proceedings of the First IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2006), Tampa, Florida, USA, Nov. 2006.
[28] T. Arampatzis, J. Lygeros, and S. Manesis, "A survey of applications of wireless sensors and wireless sensor networks," in Proc. 13th Mediterranean Conf. Control Automation, Cyprus, Turkey, Jun. 2005, pp. 719–724.
[29] R. Szewczyk, A. Mainwaring, J. Polastre, and D. Culler. "An analysis of a large scale habitat monitoring application," In Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys), Baltimore, November 2004.
[30] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buon- adonna, D. Gay, and W. Hong, "A Macroscope in the Redwoods," In: Proceedings of the 3rd ACM International Conference on Embedded Networked Sensor Systems (SENSYS), ACM Press, San Diego, CA, USA, pp. 51–63, November 2005.
[31] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees, "Deploying a Wireless Sensor Network on an Active Volcano," IEEE Internet Computing, pp. 18-25, March/April, 2006
[32] Lofar project, http://www.lofar.org/p/Agriculture.htm, [July 2010].
[33] M. Britlon and L. Sacks, "The SECOAS project: Development of a self organizing. Wireless sensor network for environmental monitoring", 2nd International Work5hop on Sensor and Actor Network Protocols and Applications, August 2004.
[34] I. Hakala, M. Tikkakoski, and I. Kivela , "Wireless sensor network in environmental monitoring - case foxhouse," in Proceedings of the Second International Conference on Sensor Technologies and Applications (SENSORCOMM 2008), Cap Esterel, France, August 25-31 2008.
[35] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "SensorScope: Out-of-the-Box Environmental Monitoring," Proceedings of the 7th international conference on Information processing in sensor networks, pp.332-343, April 22-24, 2008
[36] N. GershenfeldN, R. Krikorian, and D. Cohen, "The Internet of Things," Scientific American 2004; vol. 291, issue, pp. 76-81.
[37] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," Request for comments 4919, IETF, August 2007.
[38] J. Hui, and D. Culler, "Extending IP to Low-Power, Wireless Personal Area Networks," IEEE Internet Computing 2008, vol. 12, issue 4, pp. 37-45.
[39] L. Zhou, N. Xiong, L. Shu, A. Vasilakos and S.-S. Yeo, "Context-Aware Multimedia Service in Heterogeneous Networks," IEEE Intelligent Systems, vol. 25, no. 2, Mar./Apr. 2010, pp. 40-47.
[40] L. Zhou, X. Wang, W. Tu, G. Mutean, and B. Geller, "Distributed Scheduling Scheme for Video Streaming over Multi-Channel Multi-Radio Multi-Hop Wireless Networks," IEEE Journal on Selected Areas in Communications, vol. 28, no. 3, pp. 409-419, Apr. 2010.

Luís Miguel L. Oliveira is a PhD student of Informatics Engineering at the University of Beira Interior under supervision by Professor Joel Rodrigues and Professor Amaro de Sousa. He received his 5-year BS degree (licentiate) in Electronics from University of Aveiro, Portugal, in 1998; and the MSc degree in Electronics and Telecommunications Engineering from the University of Aveiro, Portugal in 2004. He also teaches in the Informatics Engineering Department at the Superior School of Technology

of the Polytechnic Institute of Tomar, Portugal. He is a PhD student member of the Institute of Telecommunications, Portugal. His current research areas are routing on wireless sensor mesh networks, Internet Protocol integration on wireless sensor networks and wireless sensor networks applications. He authors or co-authors more than ten international conference papers, participates on several Technical Program Committees, and also has two accepted journal publications.

Joel J. P. C. Rodrigues is a professor in the Department of Informatics of the University of Beira Interior, Covilhã, Portugal, and researcher at the *Instituto de Telecomunicações*, Portugal. He received a PhD degree in informatics engineering, an MSc degree from the University of Beira Interior, and a five-year BSc degree (licentiate) in informatics engineering from the University of Coimbra, Portugal. His main research interests include high-speed networks, delay-tolerant networks, sensor networks, e-health, e-learning, and mobile and ubiquitous computing. He is the leader of NetGNA Research Group from IT (http://netgna.it.ubi.pt), the Vice-chair of the IEEE ComSoc Technical Committee on Communications Software and the Secretary of the IEEE ComSoc Technical Committee on eHealth.

He is the editor-in-chief of the International Journal on E-Health and Medical Communications and the general chair and TPC Chair of many international conferences. He is a member of many international TPCs and several editorial review boards. He has authored or coauthored over 150 papers in refereed international journals and conferences, a book, and a patent.
Prof. Rodrigues is a licensed professional engineer (senior member), and he is member of ACM SIGCOMM, a member of the Internet Society, an IARIA fellow, and a senior member of IEEE.

Chapter 4

Ubiquitous monitoring solution for Wireless Sensor Networks with push notifications and end-to-end connectivity

This chapter consists of the following paper:

Ubiquitous monitoring solution for Wireless Sensor Networks with push notifications and end-to-end connectivity

L. Oliveira, J. Rodrigues, A. Elias and B. Zarpelão
Mobile Information Systems, vol. 10, no. 1, pp. 19-35, 2014.

DOI: 10.3233/MIS-130170

According to Journal Citation Reports published by Thomson Reuters, this journal scored ISI journal performance metrics as follows:

ISI Impact factor (2014): 0.949

Article Influence Score (2014): 0.192

Journal Ranking (2014): 76/139 (Computer science, information systems)

Journal Ranking (2014): 42/77 (Telecommunications)

Mobile Information Systems 10 (2014) 19–35
DOI 10.3233/MIS-130170
IOS Press

Ubiquitous monitoring solution for Wireless Sensor Networks with push notifications and end-to-end connectivity

Luis M.L. Oliveira[a], Joel J.P.C. Rodrigues[a,*], André G.F. Elias[a] and Bruno B. Zarpelão[b]
[a]*Instituto de Telecomunicações, University of Beira Interior, Covilhã, Portugal*
[b]*Pontifícia Universidade Católica do Paraná (PUC-PR), Londrina, Brazil*

Abstract. Wireless Sensor Networks (WSNs) belongs to a new trend in technology in which tiny and resource constrained devices are wirelessly interconnected and are able to interact with the surrounding environment by collecting data such as temperature and humidity. Recently, due to the huge growth in the use of mobile devices with Internet connection, smartphones are becoming the center of future ubiquitous wireless networks. Interconnecting WSNs with smartphones and the Internet is a big challenge and new architectures are required due to the heterogeneity of these devices. Taking into account that people are using smartphones with Internet connection, there is a good opportunity to propose a new architecture for wireless sensors monitoring using push notifications and smartphones. Then, this paper proposes a ubiquitous approach for WSN monitoring based on a REST Web Service, a relational database, and an Android mobile application. Real-time data sensed by WSNs are sent directly to a smartphone or stored in a database and requested by the mobile application using a well-defined RESTful interface. A push notification system was created in order to alert mobile users when a sensor parameter overcomes a given threshold. The proposed architecture and mobile application were evaluated and validated using a laboratory WSN testbed and are ready for use.

Keywords: Wireless Sensor Networks, Internet of Things, ubiquitous computing, network monitoring, mobile computing, RESTful Web Services, push notifications, Android applications

1. Introduction

Wireless sensors are tiny devices that are able to measure several environmental and crucial data. Recently, these devices have been used in areas such as environmental monitoring, home automation, and war scenarios. In this context, a new emerging technology called wireless sensor networks (WSNs) has become a trend in technological research [12,20]. This technology combines hundreds or even thousands of tiny and resource constrained sensor devices that communicate wirelessly in order to accomplish a common task. These devices are spatially distributed in the environment in order to collect data about surrounding environmental variables [18,28]. Each device has several sensor modules capable of measuring parameters such as temperature, humidity, and luminosity.

The main challenges regarding wireless sensor networks are power consumption of sensor devices and their connection to the Internet [8]. Power consumption is highly affected by the communication

*Corresponding author: Joel J.P.C. Rodrigues, Instituto de Telecomunicações, University of Beira Interior, Covilhã, Portugal. E-mail: joeljr@ieee.org.

with nodes. A solution to this problem is reducing the communication among nodes using better routing algorithms and shutting down (hibernate state) the nodes when they are not required [19,24].

Connecting these limited devices to the Internet is a big challenge because the use of the TCP/IP stack, as initially conceived, is too heavy in the context of WSNs. However, the use of IPv6 over Low-power Personal Area Networks (6LoWPANs) allows the transmission of IPv6 packets over IEEE 802.15.4 wireless links and enables WSNs to communicate with the Internet more efficiently [32]. Basically, 6LoWPAN adds an adaptation layer below network layer that fragments the packets and compresses the IPv6 transport layer headers. These adjustments allow the IPv6 to be used in low-power networks such as WSNs. The connectivity between 6LoWPAN-enabled WSNs and IPv6 networks is not straightforward because devices with IPv6 support are not able to handle the 6LoWPAN compression and fragmentation [2]. One solution to this issue is using an intermediary element such as a gateway that allows data exchange between WSNs and IPv6 hosts through 6LoWPAN [7,26].

The introduction of 6LoWPAN turn IPv6 suitable for resource-constrained devices enabling their connectivity to the Internet. Smartphones are becoming current personal computers, but it is a big challenge to establish end-to-end connectivity (between an end-user device and a sensor node) with all sorts of smart objects as proposed by the Internet of Things vision [16]. To solve this issue, an end-to-end connectivity solution is proposed in order to interconnect smartphones and sensor devices allowing real-time mobile monitoring.

The integration between mobile devices, specifically smartphones, and future wireless networks is crucial for the Internet growth. In the context of WSNs, the use of smartphones to control and monitor these kinds of networks represents a new trend in ubiquitous computing research [33]. The growing diversity of mobile operating systems and hardware platforms is developing new and heterogeneous network scenarios. Therefore, the construction of new models and architectures to enable the interaction between these devices in a platform independent way is essential [13].

Due to the heterogeneity of recent mobile devices and platforms, the construction of a Web service to interconnect these devices in a platform independent way require open technologies and protocols. The interaction between smart phones and WSN devices is possible by using Web technologies such as Web services and IP-enabled devices. The Representational State Transfer (REST) architecture is commonly used in the construction of Web services since it is based on HTTP that is supported by all smartphones [30]. The REST architecture is based on client-server communication, where clients request available resources from the Web service. A REST resource is defined as a set of data that is available through a well-defined RESTful interface [11]. Mobile clients request these resources using well-known HTTP methods such as GET and POST. To exchange information between the REST Web service and the mobile device, XML, and JSON media types are commonly used. Furthermore, since a WSN involves large amounts of data, the exchange of information needs to be optimized.

Wireless sensor networks are strictly related with monitoring solutions and the information collected by the sensors is more important than the sensor itself. In order to increase the efficiency of wireless sensor network monitoring, the user should be alerted when there are significant changes in sensed data. In a mobile environment, it is becoming natural for us to receive alerts or notifications from several services such as eMail and newsletters. The integration of push-notifications in ubiquitous wireless sensor networks makes sense since it is crucial for one to be alerted when a value collected by a sensor exceeds a threshold. Recent mobile operating systems are also capable to receive and presente this type of notifications seamlessly. Pushing notifications to the mobile device represents significant energy saving when compared with always-on solutions based on polling requests [29].

The main contributions of this paper are the following: (1) the introduction of a system architecture specifically designed to collect, store, and present real-time wireless sensor networks data and historical

measures in mobile environments; (2) a push notification system that allows mobile users to receive an alert over the Internet when a sensor value overcomes a given threshold; (3) an Android mobile application that presents the latest data collected by the WSN and historical measures in scalar and graphical ways; (4) and the testbed architecture definition to evaluate, demonstrate, and validate the proposed approach.

In the proposed architecture, users have three different ways of accessing sensors data. In the first one, users request up-to-date and historical data to a RESTful service that retrieves sensor data stored in a relational database. A format is also defined for the messages exchanged between the mobile devices and the RESTful service. Adoption of REST, XML, and JSON allow heterogeneous mobile clients to exchange information with the server independently of their platforms. In the second way of accessing sensors data, users request real-time data directly to the WSN gateway. Finally, in the third way, a push service notifies users when sensor data overcomes a given threshold. It is another possibility to users that do not want to keep requesting data to the RESTful service or WSN gateway, allowing client devices to optimize power consumption.

This paper is organized as follows. Section 2 reviews the available related literature about the topic. Section 3 introduces and describes the overall system architecture and the interaction between its modules. Section 4 addresses the implementation of the proposed architecture while Section 5 presents the design and construction of the mobile application. Section 6 evaluates and demonstrates the proposed solution considering its architecture and mobile application with performance measurements considering different scenarios. Finally, conclusion and further works are addressed in Section 7.

2. Related work

Wireless sensor networks (WSNs) monitoring applications demand new functionalities and design patterns to address recent challenges in efficiency, interoperability and user interaction. The increasing heterogeneity of mobile devices, operating systems, and communication interfaces requires modern architectures and mobile applications that access the information in a platform independent way. This section presents some available solutions regarding WSN monitoring. Some projects address the used communication protocols and their architectures while others focus on the application layer.

Munawar et al. [3] proposed an Open Sensor Platform to interconnect mobile devices and sensors. The solution uses available commercial hardware and software tools such as a proprietary data acquisition device. This device receives sensed data and sends the information to a host PC using a proprietary PC application. The information is then requested by the mobile device through a Symbian OS application. Since the presented solution does not use the Internet to retrieve data from the sensors, the mobility of the user is highly reduced. Also, using proprietary hardware and software is a limitation due to interoperability and costs. The use of Symbian OS is also a limitation, since it is becoming obsolete when comparing to modern mobile operating systems such as Android and iOS from Apple.

Herrera et al. [17] present an approach to wireless sensor networks monitoring using Zigbee and the iPhone platform. The focus of this solution is to collect environmental and weather data from remote stations and present the information on the mobile device. The remote stations are equipped with a microcontroller that periodically gathers data from the sensors such as wind speed and rainfall. The sensed data is then parsed into a readable format and sent to a proprietary gateway coordinator. This coordinator interconnects the Zigbee personal area network and other network with access to database servers. The data is stored in a MySQL database through PHP scripting and a web page. The mobile application sends requests to the web page in order to retrieve last sensed data. The use of PHP and

HTML instead of a structured Web Service is a limitation of this approach in terms of performance and scalability.

Hornsby et al. [1] proposed an architecture that is based on the XMPP-protocol and wireless sensor networks that support a push based notification functionality. The proposed architecture uses Atom feeds [22] to communicate with the wireless sensor network in a Web Service-like way and an UPnP gateway was used to interconnect the XMPP-based WSN and UPnP media-oriented networks. Therefore, messages are exchanged over IP protocol using the XML media type. In order to interact with the WSN, a solution was implemented in Internet tablet devices. This architecture has some drawbacks related to the use of XMPP-protocol that is not yet supported by many mobile operating systems. This protocol is also restricted to XML media type that is a limitation in modern mobile environments where the use of JSON is becoming a standard.

Cardei et al. [21] presented a RESTful architecture for healthcare patient monitoring using heterogeneous wireless sensor networks. This innovative approach uses an Android smartphone as a gateway in order to interconnect the WSN and the Internet. The heterogeneous WSN used in this system interconnects environmental, medical, and smartphone internal sensors. The sensed data is processed on the smartphone and sent to the Internet Web Service using the cellular network. This approach represents a limitation when using the mobile device as a gateway due to the limited resources of these devices, such as processing power and battery consumption.

A monitoring platform, called WSN Monitor, was proposed by Vajsar and Rucka [27]. The platform is based on client-server architecture and is focused on monitoring and managing wireless sensor networks. The collected data is stored in a MySQL database and the server processes requests from the client application using the data available in the database. The client application was developed using the proprietary Adobe Flex framework [4] that has some limitations in terms of performance and support when compared with native mobile applications.

Moreira et al. work [25] focuses on the design and construction of a mobile monitoring application for wireless sensor networks. The proposed architecture is based in REST interfaces and XML messaging. The mobile application was built on the top of the Android platform but no performance tests were conducted to validate the architecture and the mobile application. The proposed approach also features an alert functionality that alerts the user when a sensor threshold value exceeds a defined limit. The alerts are sent via SMS or e-mail and not as native push-notifications that are standard across the mobile operating system.

Tudose et al. [10] proposed a solution for home automation using a 6LoWPAN wireless sensor network and a mobile monitoring application. The architecture is based on a wireless sensor and actuator network with energy harvesting capabilities that minimize node power consumption. In order to interconnect the WSAN and the Internet, a gateway was developed. Periodically, sensed values are transmitted wirelessly to the gateway over a UDP over IPv6 connection. The gateway receives and stores the collected data sensed by WSN devices. The Android mobile application sends REST-like requests to the gateway that responds with data in JSON format. The inexistence of a structured Web Service on the proposed architecture has limitations in terms of scalability and performance. Moreover, since no database was constructed in order to store collected data, the access to historical measures is limited.

There are also commercial solutions that aim to gather, store, and display sensor data. Cosm [9], formerly named Pachube, is a solution based on a Web portal that allows users to visualize and manipulate data collected by sensors. Moreover, Cosm offers a notification system that sends HTTP POST requests to a URL selected by the user.

In the past few years, several solutions were proposed to control and monitor wireless sensor networks. This paper proposes a reliable architecture to collect, store, and present data gathered by wireless sensors

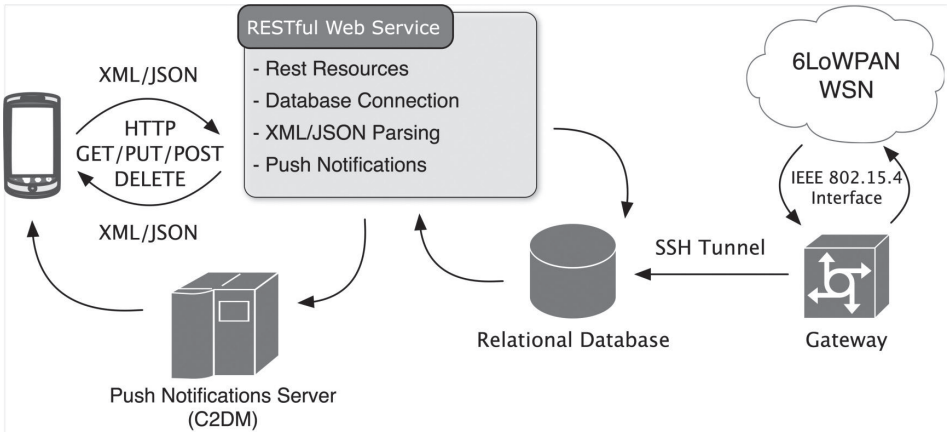


Fig. 1. Illustration of the system architecture diagram.

in a ubiquitous environment. The proposed architecture is designed for being a platform independent on a client and server sides since it is based on REST interfaces and XML/JSON messages. Besides, the native push-notification functionality presented sends a notification to the mobile user when a sensor reading overcomes a given threshold.

3. System architecture

The proposed four-tier architecture is based on the following main components: a Web service, a gateway, a relational database, and a mobile client application. The entire system architecture is illustrated in Fig. 1. The communication between these four components allows mobile client devices to present data collected by the sensors in a ubiquitous environment. The architecture was designed to enable the mobile application to present real-time data as well as stored historical measures. A push notification system integrated into the Web service and the mobile application is used to alert users if a sensor reading overcomes a given threshold.

Sensor devices send collected data to the gateway in pre-defined (and customizable) time intervals according to the variation of the physical sensed phenomenon. The software in the gateway parses the raw data and stores it in the relational database. Then, the mobile client application sends requests to the Web service that runs a query in the database and returns the result to the mobile application through a specified file format. Besides that, the mobile application can also request real-time data from the gateway. When a sensor reading overcomes a given threshold, the Web Service sends a push notification to the mobile application in order to alert the user that an event occurred. The push notification system is built for the Android operating system and is integrated in Google Cloud to Device Message servers that enable the mobile device to receive push notifications in a mobile environment.

The Web service ensures the communication between the relational database and the mobile application, using HTTP over an IPv4-based network. In order to communicate with the server, an Internet connection is required at the mobile device. This interconnection between Internet-based devices is crucial in the near future and is one of the biggest challenges of the Internet as defended by the Internet of Things vision. Designing and building the Web Service on top of industry open standards was a main requirement. Simplicity, scalability, and interoperability were also key requirements in the Web

service development process. The interoperability ensures the ubiquitous access to the database from multi-platform client applications and also the use of different file formats in the information exchange.

Nowadays, two main approaches are used in the construction of Web Services, namely Simple Object Access Protocol (SOAP) and REST. Both approaches have advantages and disadvantages depending on the deployment environment.

The REST approach uses a standard Uniform Resource Identifier (URI) that requests a unique resource in the Web service interface. The approach is simple and can be executed on any client or server that provides HTTP/HTTPS support. REST allows many different data formats while SOAP only supports XML.

SOAP uses XML in the definition of the envelope, that defines what is in the message and how to process it, in the encoding rules for data types, and finally, in the definition of the procedure calls and responses. The envelope is sent via HTTP and a Remote Procedure Call (RPC) is executed and the envelope is returned with the corresponding response in a XML formatted document. In comparison, SOAP presents some additional overhead that is not found in the REST approach.

In REST, the use of multiple data types offers some benefits since it adds support for all sorts of platforms. JSON has been widely used in REST architectures when less overhead is needed to exchange large amounts of data and also performs a faster parsing.

Each technology approach has its own characteristics and these both Web services solutions have issues related with security and transport layers. Summarizing, REST presents some advantages in scenarios with limited bandwidth, resources, and client platforms. Then, taking into account the relative performance of both approaches, the Representational State Transfer (REST) was chosen for the proposed solution.

The RESTful architecture is being widely adopted by major technology companies. Most of these companies rely on REST for sharing information and expose Web services and applications. REST architecture is based on client-server communication where clients initiate requests to servers that process these requests and return the appropriate results. These results are defined as resources and represent the information exposed by the service. RESTful architectures are based on HTTP to communicate over the network. HTTP is the Web protocol and it has a set of tools that simplify communications such as Uniform Resource Identifiers (URI), request and response headers, and Internet media types. These functionalities allow the mobile client application to use the HTTP methods GET, POST, PUT, and DELETE to communicate with the Web service and also to exchange information in several file formats such as XML and JSON.

In a mobile environment, 3G Internet connections have several limitations such as bandwidth, speed, and cost. Therefore, it is crucial to minimize data traffic in mobile applications and optimize communication protocols. The use of XML and JSON in Web services is the standard and these file structures are widely supported in both client and server architectures. A uniform file structure was defined for both XML and JSON media types in order to be parsed independently by the mobile application. The *result* tag or name was chosen to define the returned result set by the Web service and the *row* tag is used to define each individual row from the result set.

The historical information presented in the mobile application is stored in a relational database. The database was designed specifically to store all the information about a WSN and also user credentials to enable access control and manage user permissions. The scalability and flexibility were key requirements in the design of the database in order to allow its implementation across multiple server platforms and architectures. The structure of the database is a result of several iterations, from the analysis of different WSNs and the study of available solutions. Based on this analysis, several fields of the database were

defined to store all the relevant data needed to remotely access and monitor a WSN. Some of these fields are the following: MAC address, IP address, GPS coordinates; name, value, unit, and timestamp for each mote parameter; mote manufacturer and country of origin; information about the localization and the environment of the WSN deployment; and information about user credentials.

In order to get the collected data from the 6LoWPAN WSNs and store it in the database, a gateway is needed. The gateway can have more than one IPv6 interface and, at least, one 6LoWPAN interface to allow the communications between the regular IPv6 node and the WSN. The requests destined to WSN nodes are forwarded to one of the IPv6 interfaces and then sent to the 6LoWPAN adaptation layer. The 6LoWPAN adaptation layer is responsible for the packet fragmentation and reassembly in order to support the IPv6 minimum MTU, and for IP and UDP header compression. The gateway communicates with the 6LoWPAN WSN through IEEE 802.15.4. Sensed data received by the gateway is stored in the database through a JDBC connection over an SSH tunnel.

The proposed architecture joins generic messages in XML and JSON, and well-defined REST interfaces to build a communication protocol that enables clients and servers to exchange messages in a platform-independent way. A push notification system, in the context of WSNs, is also proposed and it allows mobile users to receive alerts without requesting them to the server continuously. Besides, the user can also access real-time data instead of getting latest collected values from the database. For real-time monitoring, the mobile application sends requests over the Internet to a HTTP service running on the gateway computer that queries the WSN and responds directly to the smartphone. The real-time HTTP service acts as an intermediary between the 6LoWPAN wireless sensor network and the IPv4-enabled smartphone.

4. Construction of the proposed model

The proposed model architecture was constructed in a real environment with all the needed components for a full WSNs monitoring solution. In this section, the development process of the server-side components is described, including a database, a Web service, a push notifications system, and an end-to-end connectivity between mobile devices and sensor nodes.

4.1. Database design

Based on the requirements analysis, the MySQL Database Management System (DBMS) was chosen for data storage [23]. MySQL provides scalability, flexibility, and is open source. It has support for almost all the operating systems and also provides drivers, plugins, and connectors for the majority of platforms and programming languages.

In the context of WSNs, the performance of the database is an important issue due to the large amount of data collected by sensors. The number of records and queries increases exponentially depending on the frequency of sensor readings. Tables and relations were defined according to the requirements analysis based on several WSNs and available monitoring solutions. Reducing redundant and null values was a main requirement in order to optimize performance and consistency.

The database structure is based on eleven related tables that can be divided into three groups according to the type of stored data. The entity-relationship diagram of the database is illustrated on Fig. 2. The tables *group*, *user*, and *credential* are grouped together because of the relation to user credentials and their permissions. The *user* table stores information about user credentials, such as username and password.

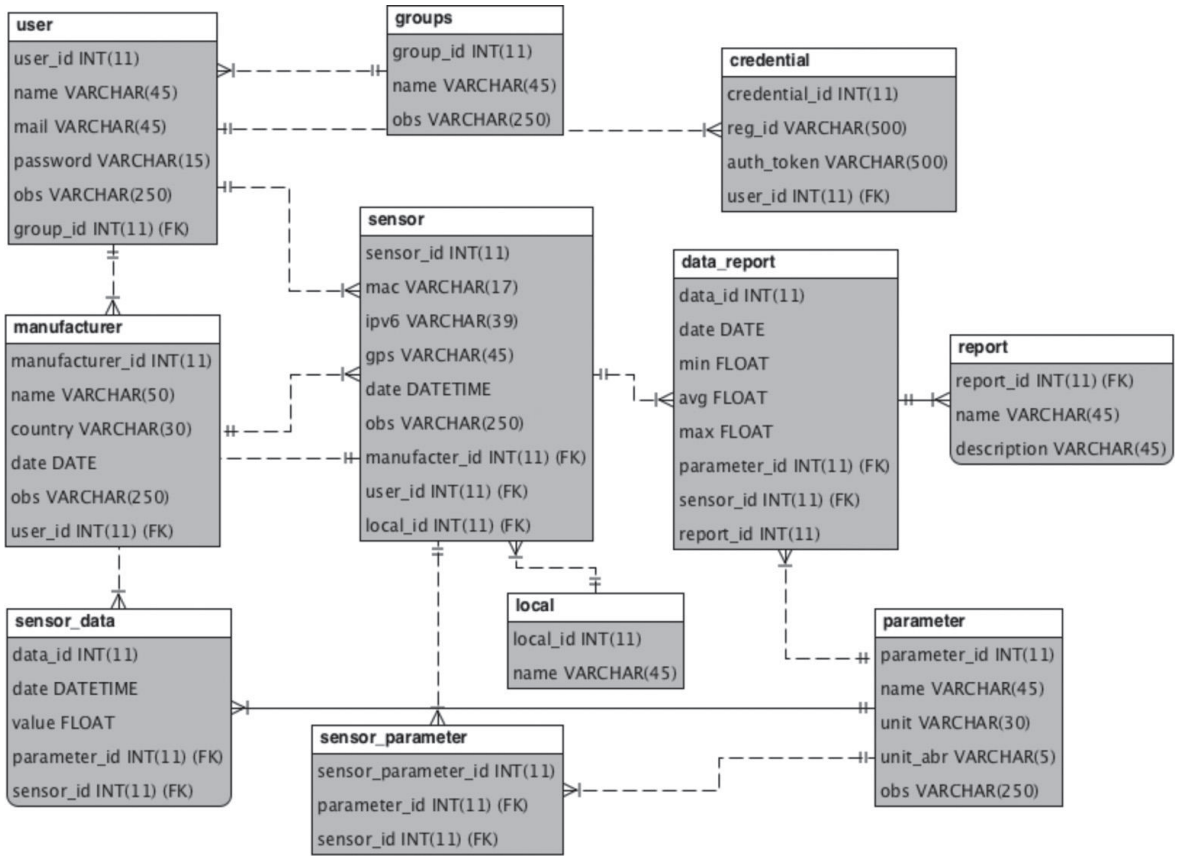


Fig. 2. Database Entity-Relationship diagram.

The *group* table stores information about groups of users that have different access permissions to available WSNs while the table *credential* stores the registration ID and the authentication token that enables the mobile application to receive push notifications. The following tables represent another group: *manufacturer, local, sensor, sensor_ parameter*, and *parameter*. These tables store information about sensor nodes, their location, specifications, and the parameters available in each mote. The *manufacturer* table holds information related to each mote specifications as well as the mote manufacturer.

The *local* table stores information about the geographical location of the WSN. Similarly, *sensor* and *parameter* tables store data related to each individual mote such as the mote's IP address, GPS coordinates, and the type of sensor parameters available at each mote. The third group of tables is formed by the tables *sensor_data, report*, and *data_report* that store the collected data by the sensors for each parameter as well as the maximum, minimum, and average values for each sensor parameter grouped by day, month, or year. The data stored in the table *data_report* are added through the MySQL event scheduler that calculates the minimum, maximum, and average values for each sensor parameter. These events are scheduled to work after each day, month, and year.

4.2. RESTful web service

To enable the information exchange between the database and the mobile devices, a RESTful Web service was created. The Web service presents a modular architecture and generic deployment in order

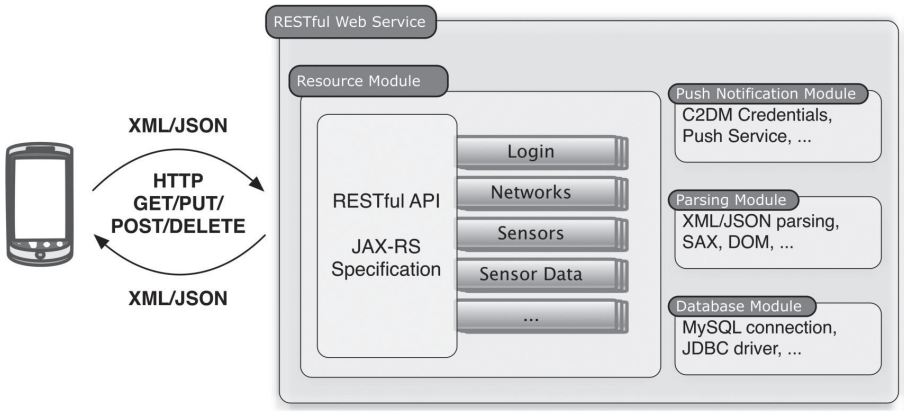


Fig. 3. RESTful Web service architecture.

to be scalable and accessed through several mobile platforms. A RESTful Web service can be defined as a set of resources, available through HTTP interfaces accessed using well-defined HTTP methods, such as GET and POST. Therefore, any client device with an Internet connection and HTTP support can send requests to the Web service in a complete platform independent way. In the construction of the Web service the Jersey open-source framework [15] was used. Jersey is the reference implementation for the JAX-RS specification [14] provided by the Java EE 6. It implements the annotations presented on the specification providing a Java API for RESTful Web services development. The modular architecture of the Web service provides scalability to the entire model and it can be further expanded by adding new modules and functionalities without changing the existing ones.

The Web service considers a four-tier approach with the following modules: the database module, the resource module, the parsing module, and the notifications module. The Web service architecture may be seen in Fig. 3. The database module manages all the connections to the database using the Java Database Connectivity (JDBC) API. The RESTful resources are available in the resource module and a unique Uniform Resource Identifier (URI) identifies each resource. A representation of the current state of each resource and a data format known as media type are used to exchange information in the RESTful environment. For example, the following uniform resource identifier (URI), "http://[server-ip]/rest/sensors" is used to request a list of the available sensors in a chosen WSN. Several unique resources were defined in the Web service, such as, the networks and sensors resources, the historical data resource, and the sensor value resource. When the Web service receives a request of an available resource, an SQL query is sent to the database and, then, the parsing module of the Web service converts the result set into the requested media type and returns the resource representation to the client. The parsing module is responsible for the construction and deconstruction of all the documents and objects exchanged between the Web service and the mobile application. When the mobile application sends an HTTP request to the Web service, the requested media type is specified in the payload to inform the Web service and the parsing module about the data type to return. This parsing module is able to create and parse XML and JSON documents based on the requested media type. For example, the XML structure is defined based on the result set returned by the database, using the names of the tables and columns to define each XML tag.

The <result> tag defines the result set returned by the server and the <row> tag is used to define each individual row of the result set. All the other tags derived from the name of each attribute in the database.

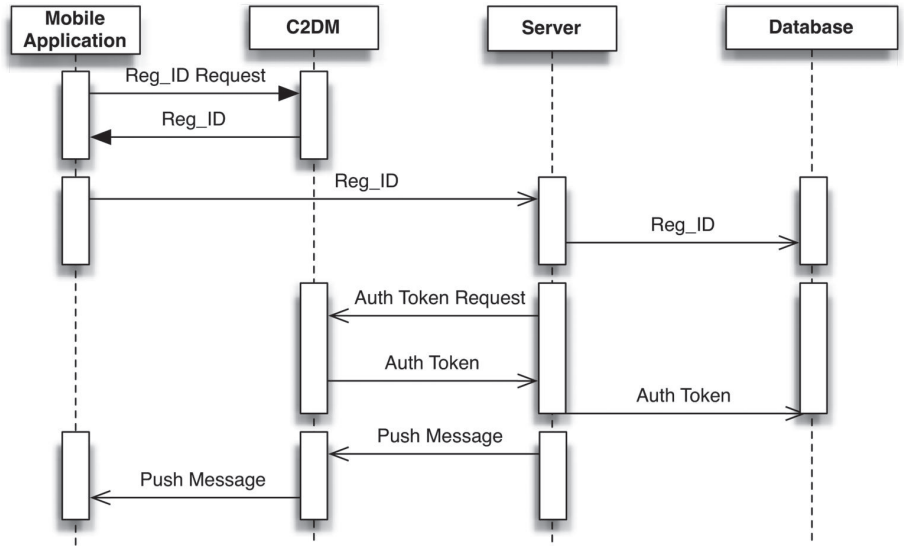


Fig. 4. Sequence diagram of the push notification system.

Thus, an example of an XML file with information about the last temperature reading on a given sensor node would be the following:

```
<?xml version=”1.0”?>
<result>
    <row>
        <value>27.14<value/>
    <row/>
<result/>
```

4.3. Push notifications

One of the main features of the proposed WSN monitoring architecture is the push notification system. This technology is capable to send messages to the mobile device without constantly poll the server for updates. Polling the server for updates has several well-known issues including the adjustment of the requests frequency, energy consumption, and high data traffic. These limitations have more impact when working with mobile devices due to limited battery life and network coverage.

The push notification system is focused on the Android operating system because it is an open platform and integrates very well the RESTful Web service and the mobile application. A new technology called Cloud to Device Messaging (C2DM), developed by Google was used in the construction of the push notification system [5]. The C2DM technology is part of the Android platform and provides libraries and APIs for developing push-enabled applications.

The integration of push notifications in the WSN monitoring model enables the mobile Android application to receive messages and alert the user when a sensor reading overcomes a predefined threshold. The push messaging technology follows a publish/subscribe model where mobile users register at the server in order to receive push messages on the mobile device. Figure 4 presents a sequence diagram of the push notification system.

To use C2DM, the mobile application must register at the Google authentication servers using a Google account. Then, a unique registration ID is generated by the authentication server and sent to the mobile device. This registration ID is forwarded to the Web service and stored in the database. The Web service also requests an authentication token that is used to send notifications to registered mobile devices. Both device registration and server authentication must be completed before sending push messages. Since the push notification system is integrated in the RESTful Web service, it polls the database continuously and when a sensor reading overcomes a predefined threshold, a push message is sent to Google C2DM servers and then to mobile clients. By doing this, the computational costs of polling are on the server side instead of stressing the mobile devices, resulting in significant energy and bandwidth savings. The push notification module was developed in Java language in order to be platform independent as well as the Web server. Therefore, several classes were added to the Web service to support three main functionalities: receiving the registration ID from mobile devices, server authentication and sending push messages.

4.4. End-to-end connectivity

In order to access real-time data from the wireless sensor networks, a multiplatform software application was developed and deployed in the gateway enabling the mobile users to request sensor measures. When the mobile client application requests data from the 6LoWPAN WSN, the software application deployed at the gateway handles the HTTP request and retrieves data directly from the WSN using the UDP transport protocol. The requested data is transmitted through IEEE 802.15.4. Then, the gateway application converts the collected data to XML format and forwards it to the smartphone over HTTP using an IEEE 802.11g wireless network with Internet connection. On the client side, the Android application parses the received data and presents it to the user. With this solution, the access to the database is not needed and the mobile application could request, on demand, data directly from the WSN bypassing the database. This performs the information exchange between the WSN and the smartphone even faster because there is no need to query the database for information. On the other hand, it is not possible to request historical sensors data.

5. Android application

The Android OS is an open mobile operating system, developed and supported by Google. It was built from scratch, specifically for mobile devices and is based on Linux kernel. The Android platform is supported by a wide range of mobile devices, from smartphones to tablets. The Android System Development Kit (SDK) provides libraries and APIs that enable developers to create Android applications and take advantage of hardware capabilities available on the devices using Java programming language. Through the APIs, developers can use functionalities such as text messaging or accelerometers in order to build richer and immersive applications. Since Android is an open platform, it integrates well with emerging technologies and Web services.

5.1. Android User Interface

The user interface was designed following the Android User Interface Guidelines [6] in order to be consistent with the operating system interface and other Android applications. A user-friendly and organized interface was a main requirement in the design process. The application follows two Android

Fig. 5. Login screen.

Fig. 6. Data visualization screen.

navigation guidelines: tabbed navigation and hierarchical navigation. Three fixed tabs at the top of the screen represent the tabbed navigation and within each tab the relationship between different screens is hierarchical.

The initial screen of the application is the login screen, as may be seen in Fig. 5. In this screen, the user must authenticate with username and password in order to access the application's main functionalities. If the user authentication is successful, a new screen is presented with the tab bar at the top where the user could choose between three tabs: Sensors tab, History tab, and Settings tab. By default, the Sensors tab is selected and presents a list view with the available WSNs for the current user. On this screen, if the user selects one of the available WSNs, a new list view is loaded with the mote names that belong to the selected network. When a unique mote is selected, the main data visualization screen is presented under the tab bar as shown in Fig. 6. This screen displays the latest sensor readings of the selected mote in both numerical and graphical modes. Following Fig. 6, "1" indicates the selected Sensors tab, while "2" points out the numerical data presentation where four sensor parameters are presented as well as each parameter unit. If more than four sensor readings are available for each mote, the user can select the four sensor readings to present simultaneously on the screen by using the button indicated by "3". At the bottom of the screen, sensed data is presented graphically as indicated by "5". Using the button pointed out by "4", the user may choose the parameter to visualize in the graphical representation. These parameter readings are presented as a line graph that is updated as new values are received by the mobile device. The graph line represents sensed values over time and the y-axis scale is constantly adjusted in order to center vertically the line graph.

If the user switches to the History tab, a new screen is shown, which presents an interface that allows the user to choose a time interval as indicated by Fig. 7. When the user defines a time interval, a

Fig. 7. Historic data screen.

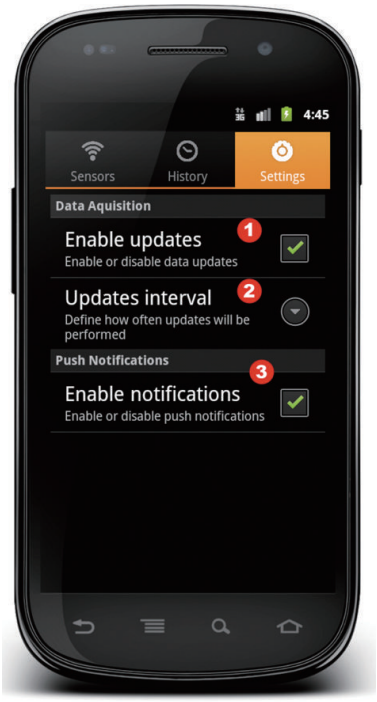


Fig. 8. Settings screen.

full-screen line graph is displayed in landscape orientation with all the sensor readings for the current parameter in that time interval. The chosen time interval may be days, hours or only a few minutes. Following Fig. 7, "1" indicates the selected History tab. Labels "2" and "3" represent the buttons that are used to define the limits of the time interval while "4" points out the button to draw the line graph.

The default application settings may be changed in the Settings tab as shown in Fig. 8. The settings menu provides the following application options: enable or disable data updates; define the updates frequency; and enable or disable push notifications. Following Fig. 8, label "1" points out the option used to enable or disable data updates. This option is enabled by default and is used to reduce battery consumption and data traffic. The frequency of data updates can be defined from 1 second to 1 minute by using the option indicated by "2". When the data updates are disabled, this option is blocked automatically. Label "3" points out the option that allows the user to enable or disable push notifications.

6. Performance evaluation and demonstration

In order to evaluate and demonstrate the architecture and mobile application, a 6LoWPAN wireless sensor network laboratory testbed was constructed. In the design of the 6LoWPAN WSN, several Telos B motes running the TinyOS operating system were used [31]. This network may be seen in Fig. 9. These motes communicate through IEEE 802.15.4 and the 6LoWPAN protocol stack is provided by the TinyOS Blip 1.0 implementation. The motes are capable of sensing air temperature and humidity, luminosity and battery voltage readings. A 6LoWPAN gateway is used to provide IPv6 end-to-end connectivity between the sensor network and the Internet. The 6LoWPAN gateway runs on Ubuntu 10.0.4 and it has multiple communication interfaces technologies, including IEEE 802.15.4, Ethernet and IEEE 802.11a/b/g. To

Table 1
Relation between the downloaded data, battery consumption, and missed updates with changing polling rates and push notifications

Polling rate	3 sec	10 sec	30 sec	Push	Idle
Downloaded data/hour	501 KB	149 KB	49 KB	301 KB	_
Battery consumption/hour	19%	13%	9%	4%	2%
Missed updates	0 %	50.4%	83.6%	_	_

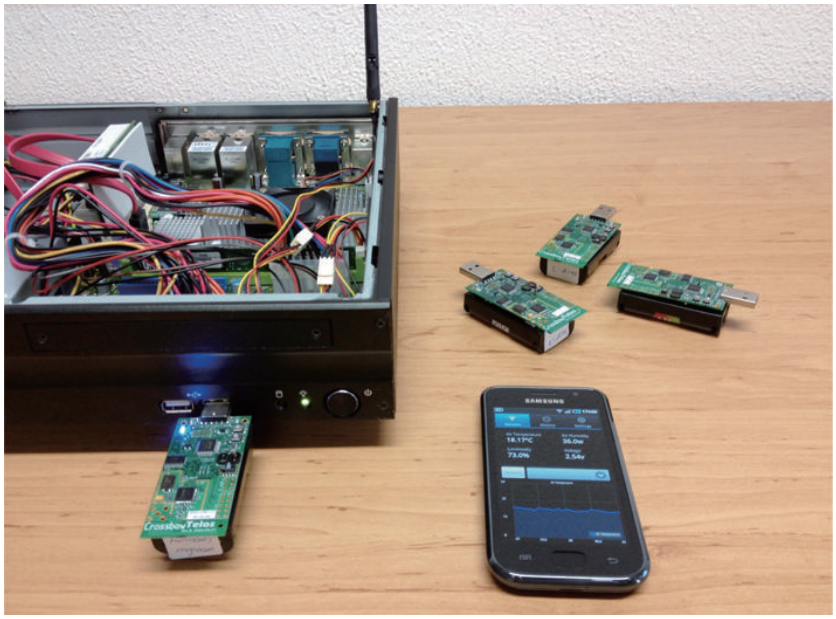


Fig. 9. 6LoWPAN wireless sensor network laboratory testbed.

implement the IEEE 802.15.4 interface in the gateway device, a TelosB mote connected to an USB port was used. An Intel desktop board D945GCLF with a 1.6 Ghz Intel Atom processor has been used to be the motherboard of the gateway. The application IP-driver compliant with RFC 4944, provided by TinyOS 2.1, acts as the 6LoWPAN adaptation layer in the gateway. The 6LoWPAN gateway is also responsible for sending ICMPv6 router advertisement messages to announce the IPv6 prefix and the default gateway address to all sensor nodes.

The smartphone used to evaluate the proposed architecture and mobile application was the Samsung Galaxy S, running Android 2.3 with a 1.0 GHz CPU and a Li-Ion 1500 mAh battery. All the tests were performed over an IEEE 802.11 g connection with Internet access. During the experiments, the phone only used essential core services, the Wi-Fi adapter and the constructed application in foreground.

Measuring energy consumption on mobile devices is not easy. There are several factors that influence the energy consumption in a mobile device, specifically a smartphone. The energy consumption is different from device to device and it also depends of the operating system version and network specifications. For each experiment, the Android battery manager was used to check the current battery level that runs from 100% when fully charged to 0%.

Table 1 shows the relation between the amount of downloaded data, the battery consumption and the missed updates for a varying polling rate with a fixed update rate. In Table 1, “Polling Rate” refers to the time interval between two requests from the mobile device to the RESTful service. A polling rate

equals to "push" means that the mobile device did not request data to the RESTful service, but waited for notifications from the push service. "Downloaded data/hour" refers to the amount of data the RESTful service transferred to the mobile device in one hour. "Battery consumption/hour" refers to the amount of energy consumed in the mobile device battery in one hour. "Missed updates" refers to the updates that were sent by the WSN gateway to the relational database but were not retrieved by the mobile device. The update rate of the wireless sensor network values was fixed in 5 seconds for testing purposes while the polling rate of the Android application varies between 3, 10 and 30 seconds. Furthermore, the same experiment was conducted with push notifications and also with the smartphone in idle for comparison purposes. All the experiments were performed during 1 (one) hour of monitoring for each polling rate.

As expected, the lower polling rate presents the lower battery consumption of the smartphone, but more updates missed by the monitoring application. The experiments also shown that if the polling rate is lower than the update rate of the WSN, none of the updates is missed but the amount of downloaded data is very high resulting in higher energy consumption. On the other hand, if the polling rate is too high, the amount of downloaded data and the battery consumption are reduced but the percentage of missed updates is also high. When the push notification system is used, the energy costs are significantly reduced. If push notifications are enabled, the mobile application is not constantly sending requests to the server and checking if there are any updates resulting in significant energy saving. As a result, ubiquitous wireless sensor networks monitoring is much more energy efficient when push technologies are used on mobile client applications.

The end-to-end connectivity between the smartphone and the 6LoWPAN WSN was also experiment in detail. Real-time temperature readings were collected and presented successfully in the mobile application.

7. Conclusion and future work

This paper proposed a ubiquitous wireless sensor networks monitoring solution allowing users to receive latest sensor readings as well as historical measures on their smartphones. The architecture was designed to be modular and was constructed based on open standards to ensure scalability and reusability. Since it is based on REST interfaces and XML/JSON messaging, the architecture is platform independent and supported in the majority of current mobile devices.

A push notification system was constructed specifically for mobile devices and is able to send push messages to smartphones if a sensor reading overcomes a given threshold. The smartphone application is also able to access real-time data over the Internet through a gateway software application. The proposed architecture was evaluated and demonstrated using a real wireless sensor testbed and an Android mobile application. The experiments showed that the solution work as planned and the push notification system has a significant impact on smartphone's energy savings.

As future work, the proposed solution may be deployed in a real environment, deploying the wireless sensor network testbed outside the laboratory. In an outdoor environment, factors such as energy management, security and weather conditions should be considered. Furthermore, the mobile application could be extended to other mobile platforms such as the iPhone and Windows Phone. With respect to storage of sensor data, NoSQL solutions may be adopted, such as document-oriented databases. Furthermore, the development of algorithms to search and locate sensor resources in the proposed REST environment may be considered.

Acknowledgments

This work has been partially supported by the *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal, by National Funding from the FCT – *Fundação para a Ciência e Tecnologia* through the Pest-OE/EEI/LA0008/2013, and by the AAL4ALL (Ambient Assisted Living for All), project co-financed by COMPETE under FEDER via QREN Programme.

References

[1] A. Hornsby, P. Belimpasakis and I. Defee, XMPP-based wireless sensor network and its integration into the extended home environment, in ISCE, IEEE 13th International Symposium on Consumer Electronics, May 25–28, 2009.
[2] A. Ludovici, A. Calveras and J. Casademont, Forwarding Techniques for IP Fragmented Packets in a Real 6LoWPAN Network, *Sensors* **11**(1) (2011), 992–1008.
[3] A. Munawar, A. Masood and F. Bangash, Open sensor platform: Integration of sensors and mobile phones, in IBCAST, International Bhurban Conference on Applied Sciences and Technology, January 9–12, 2012.
[4] Adobe Flex Framework, http://www.adobe.com/devnet/flex.html.
[5] Android Cloud To Device Messaging. http://developers.google.com/android/c2dm/. Accessed Jan 2012.
[6] Android User Interface Guidelines, http://http://developer.android.com/guide/practices/ui_guidelines/index.html. Accessed Jan 2012.
[7] B. da Silva Campos, J.J.P.C. Rodrigues, L.M.L. Oliveira, L.D.P. Mendes, E.F. Nakamura and C.M.S. Figueiredo, *Design and construction of a wireless sensor and actuator network gateway based on 6LoWPAN*, in EUROCON, International Conference on Computer as a Tool, April 27–29, 2011.
[8] C. Alcaraz, P. Najera, J. Lopez and R. Roman, Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration? 1er International Workshop on the Security of The Internet of Things, 2010.
[9] Cosm, https://cosm.com/. Accessed Mar 2013.
[10] D.S. Tudose, A. Voinescu, M. Petrareanu, A. Bucur, D. Loghin, A. Bostan and M. Tapus, *Home automation design using 6LoWPAN wireless sensor networks*, in DCOSS, International Conference on Distributed Computing in Sensor Systems and Workshops, June 27–29, 2011.
[11] F. Belqasmi, R. Glitho and F. Chunyan, RESTful web services for service provisioning in next-generation networks: a survey, *IEEE Communications Magazine* **49**(12) (2011), 66–73.
[12] I.F. Akyildiz, S. Weilian, Y. Sankarasubramaniam and E. Cayirci, A survey on sensor networks, *IEEE Communications Magazine* **40** (2002), 102–114.
[13] J. M. Corchado, J. Bajo, D.I. Tapia and A. Abraham, Using Heterogeneous Wireless Sensor Networks in a Telemonitoring System for Healthcare, *IEEE Transactions on Information Technology in Biomedicine* **14**(2) (2010), 234–240.
[14] Java API for RESTful Services. http://jax-rs-spec.java.net/. Accessed Nov 2011.
[15] Jersey Open-source Framework. http://jersey.java.net/. Accessed Nov 2011.
[16] L. Atzori, A. Iera and G. Morabito, The Internet of Things: A survey, *Computer Networks* **54**(15) (Elsevier, 2010), 2787–2805.
[17] L. Herrera, B. Mink and S. Sukittanon, Integrated personal mobile devices to wireless weather sensing network, Proceedings of the IEEE SoutheastCon, March 18–21, 2010.
[18] L.M.L. Oliveira and J.J.P.C. Rodrigues, Wireless Sensor Networks: a Survey on Environmental Monitoring, *Journal of Communications* **6**(2) (2011), 143–151.
[19] L.M.L. Oliveira, A.F. de Sousa and J.J.P.C. Rodrigues, Routing and mobility approaches in IPv6 over LoWPAN mesh networks, *International Journal of Communication Systems* **24**(11) (Wiley, 2011), 1445–1466.
[20] L. Mottola and G.P. Picco, *Programming Wireless Sensor Networks: Fundamental Concepts and State of the Art* **43**(3), ACM Computing Surveys, 2011.
[21] M. Cardei, A. Marcus, I. Cardei and T. Tavtilov, *Web-based heterogeneous WSN integration using pervasive communication*, in IPCCC, IEEE 30th International Performance Computing and Communications Conference, November 17–19, 2011.
[22] M. Nottingham and R. Sayre, *The Atom Syndication Format*, RFC 4287, December 2005.
[23] MySQL Database Management System. http://dev.mysql.com/doc/. Accessed Out 2011.
[24] N. Aslam, W. Phillips, W. Robertson and S. Sivakumar, A multi-criterion optimization technique for energy efficient cluster formation in wireless sensor networks, *Information Fusion* **12**(3) (Springer, 2011), 202–212.
[25] N. Moreira, M. Venda, C. Silva, L. Marcelino and A. Pereira, @Sensor – Mobile application to monitor a WSN, in CISTI, 6th Iberian Conference on Information Systems and Technologies, June 15–18, 2011.

[26] P. Sanyal, S. Das, S.S. Bhunia, S. Roy and N. Mukherjee, An experience of implementing IPv6 based data retrieval system for Wireless Sensor Networks, on RACSS, International Conference on Recent Advances in Computing and Software Systems, April 25–27, 2012.
[27] P. Vajsar and L. Rucka, Monitoring and management system for wireless sensor networks, 34th International Conference on Telecommunications and Signal Processing, August 18–20, 2011.
[28] P. Wang, Z. Sun, M.C. Vuran, M.A. Al-Rodhaan, A.M. Al-Dhelaan and I.F. Akyildiz, On network connectivity of wireless sensor networks for sandstorm monitoring, *Computer Networks* **55**(5) (Elsevier, 2011), 1150–1157.
[29] R. Kemp, N. Palmer, T. Kielmann and H. Bal, Energy Efficient Information Monitoring Applications on Smartphones through Communication Offloading, *Mobile Computing, Applications, and Services* **95**(2) (Springer, 2012), 60–79.
[30] R.T. Fielding, REST: architectural styles and the design of network-based software architectures, Doctoral dissertation, University of California, Irvine, 2000.
[31] Tiny OS Documentation Wiki, http://docs.tinyos.net/tinywiki/index.php/. Accessed Feb 2012.
[32] V. Kumar and S. Tiwari, Routing in IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN): A Survey, Journal of Computer Networks and Communications, 2012.
[33] Z. Li, Y. Liu, M. Li, J. Wang and Z. Cao, Exploiting Ubiquitous Data Collection for Mobile Users in Wireless Sensor Networks, *IEEE Transactions on Parallel and Distributed Systems*, Issue 99, 2012.

Luís M. Oliveira (loliveira@ipt.pt) is a PhD student of Informatics Engineering at the University of Beira Interior, Covilhã, Portugal, under supervision of Professors Joel Rodrigues and Amaro de Sousa. He received his 5-year BS degree (licentiate) in Electronics in 1998 and the MSc degree in Electronics and Telecommunications Engineering in 2004, both from the University of Aveiro. He also teaches at the Polytechnic Institute of Tomar, Portugal. He is a PhD student member of the Institute of Telecommunications, Portugal. His research interests include routing on wireless sensor mesh networks. He has authored or co-authored over ten papers in international refereed journals and conferences.

Joel J. P. C. Rodrigues (joeljr@ieee.org) is a professor at the University of Beira Interior (UBI), Covilhã, Portugal, and researcher at the *Instituto de Telecomunicações*, Portugal. He received a PhD degree in informatics engineering, an MSc degree from the University of Beira Interior, and a five-year BSc degree (licentiate) in informatics engineering from the University of Coimbra, Portugal. He is the Director of the Master degree in Informatics Engineering at UBI. He is the leader of NetGNA Research Group (http://netgna.it.ubi.pt), the Vice-chair of the IEEE ComSoc Technical Committee on Communications Software, the Vice-Chair of the IEEE ComSoc Technical Committee on eHealth, and Member Representative of the IEEE Communications Society on the IEEE Biometrics Council. He is the editor-in-chief of the International Journal on E-Health and Medical Communications, the editor-in-chief of the Recent Patents on Telecommunications, and editorial board member of several international journals. He has been general chair and TPC Chair of many international conferences. He is a member of many international TPCs and participated in several international conferences organization. He has authored or coauthored over 250 papers in refereed international journals and conferences, a book, and 2 patents. He had been awarded the Outstanding Leadership Award of IEEE GLOBECOM 2010 as CSSMA Symposium Co-Chair and several best papers awards. Prof. Rodrigues is a licensed professional engineer (as senior member), member of the Internet Society, an IARIA fellow, and a senior member of ACM and IEEE.

André Gaudêncio F. Elias is an MSc student of Informatics Engineering at the University of Beira Interior under the supervision of Prof. Joel Rodrigues and Prof. Bruno Zarpelão. He received his 3-year BS degree in Informatics Engineering from the University of Beira Interior, in 2010. In 2011, he studied in the University of Campinas, Brazil, during one MSc semester. He is an MSc student member of the Next Generation Networks and Applications Group (NetGNA) at Instituto de Telecomunicações, Portugal. His main research areas include wireless sensor networks, and mobile and ubiquitous computing.

Bruno B. Zarpelão received his B.S. degree in Computer Science from State University of Londrina, Brazil, and the Ph.D. degree in Electrical Engineering from University of Campinas, Brazil. He is currently a professor at the *Pontifícia Universidade Católica do Paraná* (PUC-PR), Londrina, Brazil and an associate researcher at the Next Generation Networks and Applications Group (NetGNA), University of Beira Interior, Covilhã, Portugal. His research interests include Smart Cities, eGov, Open Access MAN, Communication Network Management and Information Security.

Chapter 5

Wireless Sensor Networks in IPv4/IPv6 Transition Scenarios

This chapter consists of the following paper:

Wireless Sensor Networks in IPv4/IPv6 Transition Scenarios

L. Oliveira, J. Rodrigues, A. Elias and G. Han
Wireless Personal Communications, vol. 78, no. 4, pp. 1849-1862, 2014.

DOI: 10.1007/s11277-014-2048-9

According to Journal Citation Reports published by Thomson Reuters, this journal scored ISI journal performance metrics as follows:

ISI Impact factor (2014): 0.653

Article Influence Score (2014): 0.127

Journal Ranking (2014): 60/77 (Telecommunications)

Wireless Pers Commun (2014) 78:1849–1862
DOI 10.1007/s11277-014-2048-9

Wireless Sensor Networks in IPv4/IPv6 Transition Scenarios

Luís M. L. Oliveira · Joel J. P. C. Rodrigues · André G. F. Elias · Guangjie Han

Published online: 4 September 2014

Abstract The wireless sensor networks (WSNs) concept was appeared in the middle of 90s and have been a subject under intensive research in the past few years. Several factors have contributed to this, but the potential for application of WSNs in almost every aspect of day-to-day life is the predominant one. This type of networks has been developed using proprietary solutions instead of standard solutions. More recently, the importance of standards motivated the use of IETF standards in WSNs, making the Internet integration easier. However, more efforts are necessary in order to provide a full integration. The WSNs use mainly IPv6 protocol, but the IPv4 is the predominant one in the Internet. As a consequence, IPv4 to IPv6 transition mechanisms must be provided to allow the interaction between all Internet connected devices independently of the supported IP version. It is also critical to provide a standard application interface to make easier the application development and independently of the hardware platform used. The RESTfull Web services can provide this standard interface. So, combine RESTfull Web services with IPv4 to IPv6 transition mechanisms can increase the WSN services dissemination. The transition mechanisms and the REST Web services are supported in the gateway in order to save the wireless sensor device resources'. The smartphone with Internet connectivity can also be the drive to the WSNs growth, because user-friendly applications can be used to retrieve and collect sensed data. This paper proposes

L. M. L. Oliveira · J. J. P. C. Rodrigues (✉) · A. G. F. Elias
Instituto de Telecomunicações, University of Beira Interior, Covilhã, Portugal
e-mail: joeljr@ieee.org

L. M. L. Oliveira
e-mail: loliveira@it.ubi.pt

A. G. F. Elias
e-mail: andre.elias@it.ubi.pt

J. J. P. C. Rodrigues
University ITMO, St. Petersburg, Russia

G. Han
Department of Information and Communication Systems, Hohai University, Changzhou, China
e-mail: hanguangjie@gmail.com

a solution based on REST web services to permit the interaction between a mobile application and the IPv6 compliant WSN.

Keywords Ubiquitous computing · Mobile computing · Internet of things · 6LoWPAN · Wireless sensor network

1 Introduction

In the last century two important technological revolutions happened. First, the use of computers was widespread and their use is fundamental in all quotidian life aspects. Second, the Internet interconnected the computers, changing how people work, think and interact with each other's. Nowadays, there is a tendency to embed small devices with computational and communication capabilities in quotidian objects, in order to collect and process information from different sources to both control physical processes and interact with human users [1]. Connecting these objects to the Internet will be one of the biggest digital revolutions in twenty-first century [2,3]. Besides the sensing capabilities, the embed devices are characterized by small size, small computing and storage resources, power-constrains, and reduced radio ranges and throughput. Because their sensing capabilities, the network constituted by these devices is designated by wireless sensor network (WSN). A WSN can comprise hundreds or maybe thousands of sensor devices. Self-organizing, fault-tolerance and self-optimizing are their main characteristics. There are several layer-two technologies that can be used on WSN, most of them uses IEEE 802.15.4 [4] standard in the link layer, such as: ZigBee and WirelessHART. However, some of the used technologies are incompatible with IP protocols and therefore complex gateways must be used to connect these networks to the Internet. Such gateways are complex and require new programming whenever new functionalities are needed. Initially, the scientific community considered inappropriate the use IP suite protocol in small power and resource-constrained networks, because it was considered to heavy. Recently, the scientific community and the industry started to rethink many misconceptions about the use of IP in all nodes [5,6]. Two main benefits can be obtained if IP protocols are used to connect the WSN devices. First, the application developing process is simplified and open, because tools already developed for commissioning, managing and debugging can be used or in the worst case adapted. Second, connecting the WSN is easier because the protocols in use are compatible and connectivity with other Internet is assured independently of the used physical and MAC layers protocols. When compared with IPv4, the IPv6 protocol is more suitable to be used on WSN, because has enough address space to assign a global address to each device and also provides better auto-configuration mechanisms. However, the IPv6 protocol, has not been designed to be used in low power and resource constrained devices. In order to circumvent this problem, the 6LoWPAN [7] adaptation layer was defined to address these constrains and it is located between data link and network layer. The 6LoWPAN also provides adapted mechanisms for neighbour discovery and the support for new routing approaches.

Three main connectivity models, based on IP suite support, can be used to integrate the WSN in the Internet [8]. In the first model, all nodes support IP suite protocols however, the WSN is isolated from the Internet, for example, due to security reasons. In the second model, a proxy is used to mediate the connections between the Internet devices and the WSN nodes. In the third model, a global IPv6 address is assigned to each WSN device and end-to-end connectivity to the Internet is allowed. Nowadays, both IPv6 and IPv4 protocols are in use, however the IPv4 is the predominant protocol in the Internet. The migration of

IPv4-based infrastructures to those supporting IPv6 is one of the biggest challenges in the deployment of IPv6 [9]. It is commonly accepted that the transition from IPv4 to IPv6 will be slow and smooth and consequently both protocols will coexist during some time. In order to ensure a successful transition, the IETF IPng transition-working group has been working on several transition approaches, mechanisms and tools. Nowadays most of the Internet service providers don't provide IPv6 connectivity to their costumers. As consequence, IPv4 to IPv6 transition mechanism must be used in order to permit the interaction between IPv4 devices and the WSN, that mainly uses IPv6 protocol. In fact, it doesn't make sense to use IPv4 in WSN, because the transition to IPv6 is underway and as above described the IPv6 protocol is more suitable to be used in this type of networks.

Two different approaches can be used to integrate the WSN into Internet; the first one uses gateways that act as proxies between the Internet devices and the WSN nodes [10]. The second one, rely on end-to-end connectivity. The interoperability between the WSN devices and Internet is the main concern of both approaches. The network layer interoperability is already achieved using IP protocol in sensor nodes. The next step is to provide application level interoperability [11]. The same methodology used in the network layer can also be used to provide the same level of interoperability at the application layer, i.e. choose one of the most used application layer in the Internet, for example the HTTP, and make it viable to be used on WSN devices [12].

The dissemination of small mobile devices with Internet connectivity, such as smartphone and tablets, combined with user-friendly applications can be used to retrieve and collect sensed data from anywhere at any time independently of the WSN protocols in use and can also be the drive to the WSN applications growth.

This paper presents an IPv4/IPv6 transition mechanism for WSNs and it is demonstrated and validated on a solution based on a REST Web service [13] to allow the interaction between mobile applications and IPv6 compliant WSN. IPv4 to IPv6 dual stack transition mechanism installed in the WSN gateway is used to accept requests from both IPv4 and IPv6 mobile devices. The main contributions of this paper are the following:

- A new mechanism that provides simultaneously the capacity to communicate with both IPv4 and IPv6 devices and a standard application interface based on a RESTfull Web services.
- The proposed mechanism is supported in the gateway in order to save the wireless sensor device' resources. Moreover, no changes are required in the WSN devices.
- Security mechanisms that can be used to protect the WSNs against denial of service were added to the proposed approach. With this mechanism, only valid requests from the Internet and destined to available wireless sensor nodes will be forwarded.

The remainder of this paper is organized as follows. Section 2 analyses the related technologies, while Sect. 3 presents the proposal of the overall model architecture. Section 4 demonstrates the architecture and the mobile application for Android operating system. Finally, Sect. 5 concludes the paper and pinpoints future research topics.

2 Related Technologies

2.1 6LoWPAN

IEEE 802.15.4 [4] is a standard for communications on LoWPAN networks introduced by IEEE to address the low-power and low-rate wireless personal area networks requirements.

The IEEE 802.15.4 protocol defines the physical and the media access control layers for such networks and is *a de facto* protocol for WSNs. The IEEE 802.15.4 [4] physical layer defines three operation modes: (i) 20 kbps at 868 MHz, (ii) 40 kbps at 915 MHz and (iii) 250 kbps at 2.4 GHz (DSSS). The frame length is limited to 127 bytes, because low-power wireless links are used in communications and the sensor devices have limited buffering capabilities.

The network layer protocol must comply with the constraints imposed by the lower layer protocol in use. In fact, the requirements of the IPv6 protocol don't fully match with the IEEE 802.15.4 constraints. For example the minimum IPv6 MTU is 1,500 bytes and the IEEE802.15.4 MTU is 127 bytes. Besides to this incompatibility, using standard IPv6 headers would result in extremely small payload for high protocols. To address these issues, the IETF 6LoWPAN-working group were created to define the support of IPv6 over IEEE 802.15.4. The 6LoWPAN [7] working group were mainly focused on the following items: (i) to define limited extensions to IPv6 neighbour discovery protocol more adapted for WSN; (ii) to describe mechanisms to compress 6LoWPAN headers and (iii) to define 6LoWPAN routing approaches and protocols adapted to WSN characteristics. To support IPv6 over IEEE 802.15.4 an additional adaptation layer was introduced between data link and network layers. Instead of defining a single header, like IPv4, the 6LoWPAN use stacked headers as the original IPv6 protocol does. In this case, it does not need to use unnecessary header fields for mesh networking or fragmentation and it uses only the minimum necessary headers. The 6LoWPAN standard defines four header types: (i) the dispatch header, (ii) the IPv6 header compression header, (iii) the fragmentation header and (iv) the mesh header. In the simplest case, only the dispatch and compression headers are used. At the beginning of each header, a header type field identifies the header format.

2.2 Internet WSN Connectivity Models

Three main models can be used to connect the low power and resource-constrained networks to the Internet [8]. In this first deployment model, the low power and resource-constrained network are not connected to the Internet. In fact, there are several scenarios that do not require any connectivity with the Internet, for example the smart grid applications. Smart grid networks are used to monitoring the power generation networks, the automation and control devices, smart metering and building and home energy management. These networks can also use the IP protocol suite in all nodes but, due security and privacy reasons these networks are disconnected from the public Internet. In this case, supporting IP suite in all devices continues to be advantageous, because it improves the application development and the network management.

In the second connectivity model, the gateway acts as a proxy device (Fig. 1). Internet user will have access to the information provided by WSN devices, such as environmental data, but true end-to-end connectivity is not supported. The proxy can act as a server that collects and stores data retrieved from the WSN node devices. This connectivity model can be used to preserve scarce resources on WSN nodes, such as, the energy because only a small amount of the requests sent by the Internet users are forwarded to the WSN. The IPv4 to IPv6 transition mechanisms can also be located in the gateway, because is main powered and has more hardware resources when compared with WSN nodes. Moreover, the proxy mechanism can also be used to provide security enforcement and traffic filtering [14].

In the third model, the WSN is considered as an extension to the Internet and end-to-end connectivity is supported (Fig. 2). This connectivity model requires simpler gateway devices, however security enforcement is harder to implement when compared with the second connectivity model. In this connectivity model the Internet devices and the WSN

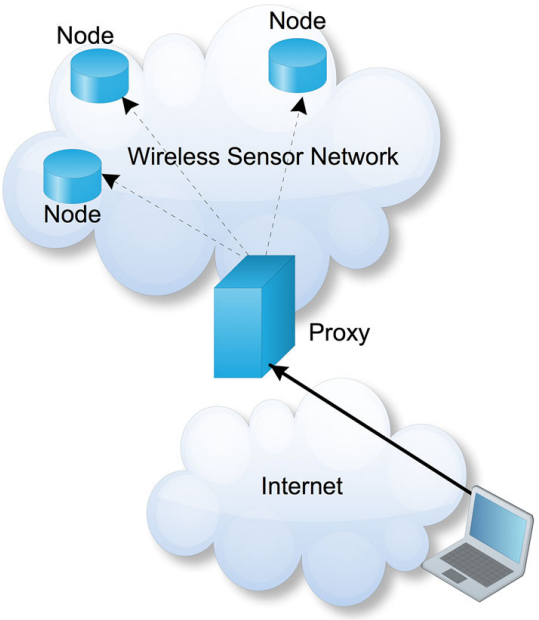


Fig. 1 Connecting the smart objects using a proxy device

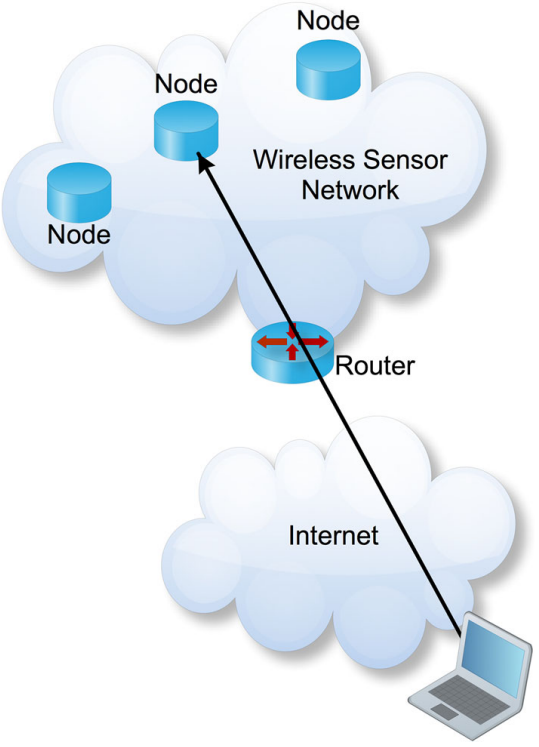


Fig. 2 Illustration of extended Internet connectivity

should use the same version of network and upper layer protocols. In fact, it is not possible to support transition mechanisms in all WSN nodes because the scarce resources. As a consequence it only be massively deployed in a near future when the transition from the IPv4 to IPv6 is concluded (Fig. 3).

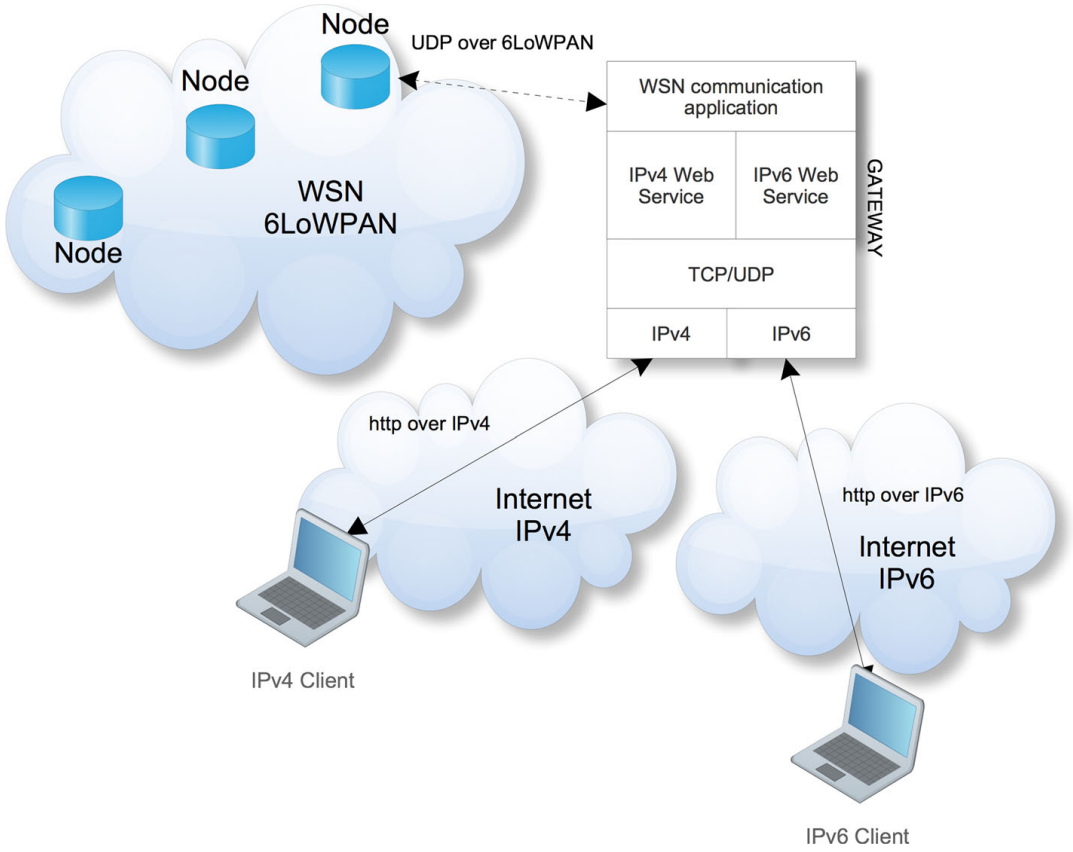


Fig. 3 System architecture illustration

2.3 IPv4 to IPv6 Transition Mechanisms

The IPv4 to IPv6 transition mechanisms can be classified in dual stack, tunneling and translators [9]. In dual stack mechanisms, both IPv4 and IPv6 protocols stacks are supported in all network devices. The application layer decides which IP stack should be used according to DNS resolution, the application layer and to the remote system IP version. This is the most straight forwarded way to implement transition scenarios, however this mechanism requires a double effort to run both protocol stacks and doesn't solve the inter-working between IPv4 and IPv6 hosts. Tunneling mechanisms assume that end systems have the same IP version, but intermediate networks supports only other IP version. Tunnels are implemented through encapsulation; the tunnel entry node puts the original IP packet in the payload of an IP packet of the other version and sends it to the tunnel exit address. The exit node removes the outer IP header. There are manual and automatic tunnels. In the manual tunnels, the tunnel end-points must be manually defined. In the automatic tunnels, the end-points addresses are embedded on the node's addresses. The use of tunneling leads to high protocol overhead due, to multiple protocol headers in same packet. This overhead is even more perceptible on WSN, which have scarce resources. This makes tunneling less efficient and results in performance deterioration. Translation mechanisms are required when the end systems support different IP versions. The translation is done either at the IP layer involving translation of IP addresses and mapping of IP header fields, or at higher layers (e.g., transport and application layers).

Support IPv4 to IPv6 transition scenarios requires additional storage space and CPU and consequently more energy. This overhead can be critical and therefore the transition mech-

anisms should be performed not on WSN nodes but on resource and energy unconstrained devices, such as gateways. In fact, the dual stack mechanism requires more space to store both IP stacks. The tunnelling mechanism increases the packet size and the CPU workload and consequently reduces the node's lifetime due to energy overhead [15]. Finally, the translation mechanisms require more CPU cycles to perform the translation and memory space to store the translation mechanism.

2.4 Web Services

Besides to the standardization on layers two and three, currently most of WSNs are based on specialized software and hardware platforms and custom high-level protocols and APIs [11]. These non-standard high-level protocols and APIs are limiting the interaction between the WSN nodes and the Internet devices. The Web of things proposes the integration of the embedded systems into the Web in order to provide a standard interaction at the upper layer protocol level. Each embedded device should be available using web protocols, such as hypertext transfer protocol [13].

The Web architecture relies on client-server architecture and therefore the clients have to actively pull the content instead of getting it pushed to them. But, in a typical physical monitoring application, it is necessary an asynchronous behavior. First, to access the updated readings in real-time the sensors asynchronously must send updates. Second, it is necessary support time-consuming operations, which do not return results immediately but, when they are available. Finally, rather than pooled periodically, the WSN nodes should communicate only when new data is available [16,17].

Initially the Web has a synchronous communication model only, a client opens a connection to a server, sends a request and the server responds to the request. Technologies such as persistent connections and Ajax [18] can be used to create faster and more interactive web applications. Although, both Ajax and persistent connections are statefull technologies and therefore the server must store the state of many persistent connections.

The Web was initially developed to provide the interaction between Web browsers and Web servers. However, there are many situations when it is desirable to have interaction between generic applications, located on the client site and web servers. A Web service is a method of communication between two applications or electronic devices over the Web. There are two types of Web services: simple object access protocol (SOAP) and representational state transfer (REST) [16]. The SOAP defines a standard communication protocol specification for XML based message exchange. SOAP can use different transport protocols, such as HTTP, JMS and SMTP. Most of SOAP implementations use HTTP because it can be tunnel across firewalls and proxies without any modifications. The REST describes a set of architectural principles by which data can be transmitted over a normalized interface (such as HTTP). REST does not contains an extra messaging layer and efforts on design rules for creating stateless services. A client can access the resource using the unique URI [19]. A representation of the resource represented by the URI is returned. The URI of the resource serves as the resource identifier and GET, PUT, DELETE, POST and HEAD are the standard HTTP operations to be performed on that resource. When compared with SOAP, the REST is more suitable when WSN networks and mobile devices are involved. The RESTful Web services are completely stateless and its implementation is simpler when compared with SOAP. The REST is particular useful for energy and bandwidth constrained devices, such as, mobile devices, for which the overhead of additional parameters like headers and other SOAP elements are critical. Moreover, REST services provide a good caching mechanism, based on the HTTP GET method, to the data that is not altered frequently [16,17].

3 System Architecture

The WSN growth depends on the capacity to provide new services that can be easily used by a large number of users [8,20,21]. This paper proposes a new solution that allows the access to the data collect by 6LoWPAN compatible sensor devices, to both IPv4 and IPv6 capable smartphones. Smartphones with IPv4 or IPv6 Internet connection; a 6LoWPAN gateway and WSN compatible with 6LoWPAN protocol are the main blocks of the proposed architecture (Fig. 3). A software solution that combines REST-like Web services and a Java application was constructed and deployed in the gateway. The Java application is responsible for retrieving sensed data from the WSN through an IEEE 802.15.4 interface. All the communications between the gateway and the sensor devices use the UDP transport protocol over 6LoWPAN. In fact, UDP is simpler than TCP and also result in benefits in terms of payload and compression [8]. The constructed Java application is integrated with the REST-like Web services [22] that enable the communication between the gateway and the Internet capable mobile devices [23].

The REST architecture [22] was chosen for the development of the Web service because it allows remote access from multi-platform client applications over GET and POST HTTP methods. The REST architecture is based on client-server communication, where clients request available resources from the server in defined media-types such as XML and JSON [24]. The Web service was constructed in two modules: the WSN Connection module and the Resources module. The WSN Connection module integrates the developed Java application into the Web services [25] in order to forward the requests from the mobile devices to the WSN. The Resources module receives and processes the HTTP requests and returns the result data set to the mobile devices.

The gateway supports dual stack mechanism in all physical interfaces, so it is possible to receive requests and send responses encapsulated in both IPv4 and IPv6. Two versions of the Web Service are available, the first supports IPv4 and the second supports IPv6. When data is requested over IPv4, the request is handled by the IPv4 Web service and forwarded to the WSN over IPv6. Then, the WSN responds over IPv6 and the Web service returns the data encapsulated in IPv4 to the mobile device. In order to avoid IPv4/IPv6 address translation, a list of the available sensors is presented to the user by the mobile application, as well as information about the location and supported functionality. So, the sensors are not identified by its IP address, but by his name, location or supported features. A register mechanism based on 6LoWPAN was developed in order to maintain the list of available sensor nodes and their capabilities (such as the type of transducers) [10]. The administrator is responsible to assign the name and location to each registered sensor device.

In the construction of the Web services, the open source JAX-RS specification was used because it is based on the Java language and integrates well with the developed Java application use to retrieve information from the WSN [17]. The XML media-type was used to exchange information between the Web service and the mobile devices because is widely supported in recent mobile operating systems [20,25–27]. When the gateway software request data from the WSN, network motes collect data using the integrated transducers and send it to the gateway. Then, the received data is parsed to the XML media-type and forwarded to the mobile device. After receiving the collected data, the mobile application parses the XML file and presents the data to the user. Two optimizations were introduced in order to save bandwidth and energy and to avoid denial of service attacks. First, only valid requests are forwarded to the WSN network. A request is considered valid if the sensor is available and if it supports the requested data. For example, it doesn't make sense forward a temperature measure request if the target sensor not has a temperature transducer. Second, the gateway

caches the retrieved from the sensor devices during a configurable time in order to avoid repeated requests of the same data. HTTPS protocol can be used to provide security to the data exchanged between the gateway and the mobile application.

4 Performance Evaluation, Demonstration, and Validation

A laboratory testbed was implemented in order to validate the operation of the proposed solution. The testbed scenario comprises four TelosB wireless sensor nodes, a gateway and smartphone with android operating system (Fig. 4). This section presents the testbed deployment details and the obtained results.

4.1 Wireless Sensor Networks

Four TelosB motes, with 6LoWPAN [4,7–14,16–20,22,23,28] support and equipped with air temperature, light, battery voltage and humidity transducers were used to deploy a single hop WSN. ICMPv6 router advertisements sent by the gateway is used announce the IPv6 prefix fec0::/64, used by all WSN in IPv6 address auto-configuration. A modified version of ICMPv6 neighbor discovery is used to register the node's IPv6 address and functionalities [14]. TinyOS 2.1 Blip 2.0 [28] implementation was used to support 6LoWPAN in all WSN nodes. Some new commands were added to TinyOS 2.1 UDPEcho application in order to retrieve the data from the WSN nodes, for example read temperature and read all values. The same modified version of UDPEcho was installed in all WSN nodes.

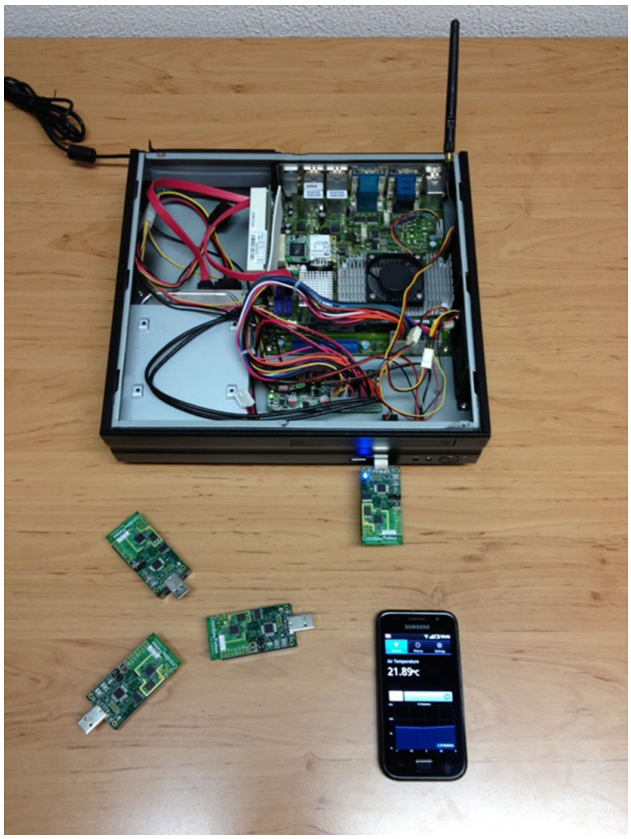


Fig. 4 Photo of the laboratory testbed

A 6LoWPAN gateway is used to provide connectivity between the WSN and the Internet. The gateway was implemented on top of Ubuntu 10.0.4 OS and it has multiple communication interfaces technologies, including IEEE 802.15.4, Ethernet and IEEE 802.11a/b/g. The gateway connects to the WSN through an IEEE 802.15.4 TelosB mote running TinyOS Blip2.0 [28] connected to an USB port. An Intel desktop board D945GCLF with a 1.6 GHz Intel Atom processor has been used to be the motherboard of the gateway. The 6LoWPAN gateway is also responsible for announcing the IPv6 prefix and the default gateway address to all sensor nodes. IPtables firewall distributed with Ubuntu 10.0.4 (ip6table) was used to permit only HTTP and HTTPS from the Internet. The IPtables is also used to rate limit the requests sent to the WSN. The rate of the requests can be adjusted and in our experiment only ten packets per second are permitted.

4.2 REST Web Service and Mobile Application

An Android application was developed to evaluate and demonstrate the proposed solution. The Android OS is an open-source mobile operating system provided and supported by Google and based on the Linux kernel. Android OS was built from scratch to run specifically on smartphones and tablets. To design and develop Android mobile applications, Google provides the Android System Development Kit (SDK) that is built on top of Java language and has the necessary libraries, APIs and tools to write native Android applications and deploy it to the mobile device. The Eclipse integrated development environment (IDE) combined with Android SDK plugins was used to design and develop the application.

The mobile application was developed accordingly to the architecture and gateway software. It sends IPv4 or IPv6 HTTP requests over the Internet to the gateway Web service and receives the responses in XML. The Web services run on top of the Tomcat application server, version 7.0, in order to provide a pure HTTP web server environment.

The user interface of the application was designed to be user friendly and to present data to the user in a simple and meaningful way. After launched, the mobile application sends a

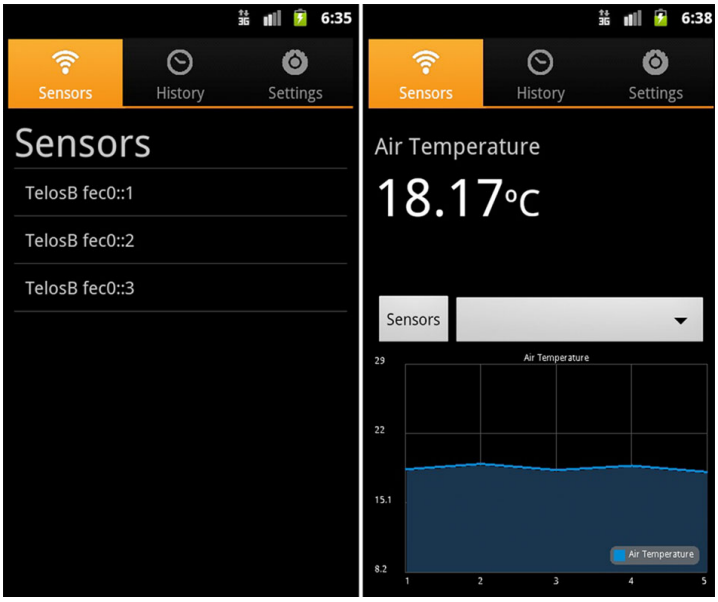


Fig. 5 Sensors and Data visualization screens

HTTP GET message to the URI [19] http://[server-ip]/rest/sensors, to request the available sensors on WSN and presents the result information in a dynamic list view. A capture of the application environment is represented on the left side of Fig. 5. When the user choose one of the sensors on the list, the application automatically sent a requests to the Web Services, after validated and if the requested value is not stored in a CSV file used to cache values, a message with the request is sent to the selected sensor node. The WSN node processes the request and the response is sent to the correspondent Web service. The received data is presented on the Android application in both scalar and graphical ways as illustrated on the right side of Fig. 5.

The application was deployed and tested using a Samsung Galaxy S smartphone running Android 2.3.3 with a 1.0 GHz CPU and a Li-Ion 1500mAh battery. The Android operating system supports both IPv4 and IPv6 protocol suite, however none of the available UMTS operator provides at the moment IPv6 connectivity. So, IPv4 connectivity was tested in both UMTS and wifi connections and IPv6 connectivity only over wifi the laboratory local area network. The tests proved that the application is stable and is able to receive WSN sensor data using both IPv4 and IPv6-enabled networks.

5 Conclusion and Future Work

This paper proposed a solution based on RESTfull Web services to allow communications between IPv4 and IPv6 capable mobile devices and IPv6 compliant WSNs. The gateway used to connect the WSN to the Internet support dual stack IPv4 to IPv6 transition mechanism in order to accept requests from both IPv4 and IPv6 clients. The transition mechanism is only supported in the gateway to save the wireless sensor device's resources. In fact, none of the available transition mechanisms are suitable to be used on wireless sensor devices. The employment of generic XML messages and REST interfaces allows mobile clients and servers to exchange data independently of the platform used and as consequence it increases simultaneously the interaction between Internet connected devices and also makes easier the applications development on the client side. The sensor devices are accessed based on its name, location and supported functionalities. The proposed solution also includes a cache to store the data retrieved from the WSN devices and a mechanism to validate the requests from the Internet devices. Both functionalities can be used to save the WSN node's resources and to protect them from denial of service attacks. Note that interconnection between different platforms and architectures through the Internet follow the internet of things vision. In order to evaluate and demonstrate the proposed model, a WSN laboratory testbed was deployed.

As future work, the proposed model may be extended outside the laboratory for real case scenarios. In this sense, issues such as power management and security must be considered. Furthermore, a push notification system may be an improvement to the model in order to alert the user if a sensor reading overcomes a given threshold.

Acknowledgments This work has been partially supported by *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Covilhã Delegation, by Government of Russian Federation, Grant 074-U01, by National Funding from the FCT - *Fundação para a Ciência e a Tecnologia* through the Pest-OE/EEI/LA0008/2013 Project, and by the AAL4ALL (Ambient Assisted Living for All), project co-financed by the European Community Fund FEDER through COMPETE *Programa Operacional Factores de Competitividade*.

References

1. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor network: A survey. *IEEE Communications Magazine*, *40*, 102–114.
2. Oliveira, L. M. L., & Rodrigues, J. J. P. C. (2011). Wireless sensor networks: A survey on environmental monitoring. *Journal of Communications (JCM)*, *6*(2), 143–151.
3. Oliveira, L. M. L., Sousa, A. F., & Rodrigues, J. J. P. C. (2011). Routing and mobility approaches in IPv6 over LoWPAN mesh networks. *International Journal of Communication Systems*, *24*(11), 1445–1466.
4. IEEE Std 802.15.4-2006. (2006). Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs). IEEE Std. 802.15.4-2006.
5. Alcaraz, C., Najera, P., Lopez, J., & Roman, R. (2010). Wireless sensor networks and the internet of things: Do we need a complete integration?. In *1st International workshop on the security of the internet of things*.
6. Hui, J., & Culler, D. (2008). IP is dead, long live IP for wireless sensor networks. In *Proc. of 6th ACM conference on embedded network sensor systems (SenSys)* (pp. 15–28). ACM.
7. Kushalnagar, N., Montenegro, G., & Schumacher, C. (2007). IPv6 over low-power wireless personal area networks (6LoWPANs): Overview, assumptions, problem statement, and goals. Internet Engineering Task Force, Request for comments 4919.
8. Vasseur, J., & Dunkels, A. (2010). *Interconnecting smart objects with IP*. Burlington: Morgan Kaufmann. ISBN:978-0123751652.
9. Waddington, D. G., & Chang, F. (2002). Realizing the transition to IPv6. *Communications Magazine*, *40*(6), 138–147.
10. Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, *54*(15), 2787–2805.
11. Li, Z., Li, M., Wang, J., & Cao, Z. (2011). Ubiquitous data collection for mobile users in wireless sensor networks. In *IEEE INFOCOM 2011*, April 10–15.
12. Belqasmi, F., Glitho, R., & Fu, C. (2011). RESTful web services for service provisioning in next-generation networks: A survey. *Communications Magazine, IEEE*, *49*(12), 66–73.
13. Guinard, D., & Vlad, T. (2009). Towards the web of things: Web mashups for embedded devices. In *Workshop on Mashups, Enterprise Mashups and lightweight composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain*.
14. Oliveira, L. M. L., Rodrigues, J. J. P. C., Sousa, A. F., & Lloret, J. (2013). Denial of service mitigation approach for IPv6-enabled smart object networks. *Concurrency and Computation: Practice and Experience*, *25*(1), 129–142.
15. Rawat, P., & Bonnin, J. (2010). Designing a header compression mechanism for efficient use of IP tunneling in wireless networks. In *The 7th annual IEEE consumer communications and networking conference (CCNC), Las Vegas, Nevada, USA*.
16. Pautasso, C., & Wilde, E. (2010). RESTful web services: Principles, patterns, emerging technologies. In *Proceedings of the 19th international conference on world wide web*. ACM.
17. Jersey Project. (2011). http://jersey.java.net. Accessed 24 Dec 2013.
18. Garrett, J.J. (2005). Ajax: A new approach to web applications.
19. Berners-Lee, T., Fielding, R., & Masinter, L. (2005). Uniform resource identifiers (uri): Generic syntax. RFC 3986, Internet Engineering Task Force.
20. Oliveira, L. M. L., Rodrigues, J. J. P. C., Elias, A. G. F., & Zarpelão, B. B. (2013). Ubiquitous monitoring solution for wireless sensor networks with push notifications and end-to-end connectivity. In: *Mobile information systems*, IOS Press, ISSN(online):1875–905X, ISSN (print):1574–017X. doi:10.3233/MIS-130170.
21. Caldeira, J. M. L. P., Rodrigues, J. J. P. C., & Lorenz, P. (2012). Towards ubiquitous mobility solutions for body sensor networks on healthCare. *IEEE Communications Magazine, IEEE*, *50*(5), 108–115. doi:10.1109/MCOM.2012.6194390.
22. Fielding, R. T. (2000). *REST: Architectural styles and the design of network-based software architectures*. Doctoral dissertation, University of California, Irvine.
23. Kumar, V., & Tiwari, S. (2012). Routing in IPv6 over low-power wireless personal area networks (6LoWPAN): A survey. *Journal of Computer Networks and Communications*. (Article ID 316839, p. 10). doi:10.1155/2012/316839.
24. Crockford, D. (2006). The application/json media type for javascript object notation (json). Internet Engineering Task Force, Request for comments 4627.
25. Li, Z., Liu, Y., Li, M., Wang J., & Cao, Z. (2012). Exploiting ubiquitous data collection for mobile users in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 99.

26. Kansal, A., Nath, S., Liu, J., & Zhao, F. (2007). Senseweb: An infrastructure for shared sensing. *IEEE MultiMedia*, *14*(4), 8–13.
27. Kemp, R., Palmer, N., Kielmann, T., & Bal, H. (2012). Energy efficient information monitoring applications on smartphones through communication offloading. *Mobile Computing, Applications, and Services*, *95*(2), 60–79.
28. Tiny OS Documentation Wiki. (2013). http://docs.tinyos.net/tinywiki/index.php/. Accessed 24 Dec 2013.

Luís M. L. Oliveira is a PhD student of Informatics Engineering at the University of Beira Interior under the supervision of Professor Joel Rodrigues and Professor Amaro de Sousa. He received his 5-year BS degree (licentiate) in Electronics from the University of Aveiro, Portugal, in 1998; and his MSc degree in Electronics and Telecommunications Engineering from the University of Aveiro, Portugal in 2004. He also teaches in the Informatics Engineering Department at the Superior School of Technology of the Polytechnic Institute of Tomar, Portugal. He is a PhD student member of the Instituto de Telecomunicações, Portugal. His current research areas are routing on wireless sensor mesh networks, Internet Protocol integration on wireless sensor networks and wireless sensor networks applications. He authors or co-authors more than fourteen international conference papers and also has seven accepted journal publications. He has been acting as a reviewer for international and conferences.

Joel J. P. C. Rodrigues a professor in the Department of Informatics of the University of Beira Interior, Covilhã, Portugal, and researcher at the Instituto de Telecomunicações, Portugal. He received the Habilitation in computer science and engineering from the University of Haute Alsace, France, a PhD degree in informatics engineering, an MSc degree from the University of Beira Interior, and a five-year BSc degree (licentiate) in informatics engineering from the University of Coimbra, Portugal. His main research interests include sensor networks, e-health, e-learning, vehicular delay-tolerant networks, and mobile and ubiquitous computing. He is the leader of NetGNA Research Group (http://netgna.it.ubi.pt), the Chair of the IEEE ComSoc Technical Committee on eHealth, the Past-chair of the IEEE ComSoc Technical Committee on Communications Software, Steering Committee member of the IEEE Life Sciences Technical Community, Member Representative of the IEEE Communications Society on the IEEE Biometrics Council, and officer of the IEEE 1907.1 standard. He is the editor- n-chief of the International Journal on E-Health and Medical Communications, the editor-in-chief of the Recent Advances on Communications and Networking Technology, and editorial board member of several journals. He has been general chair and TPC Chair of many international conferences, including IEEE ICC and GLOBECOM. He is a member of many international TPCs and participated in several international conferences organization. He has authored or coauthored over 350 papers in refereed international journals and conferences, a book, and 3 patents. He had been awarded the Outstanding Leadership Award of IEEE GLOBECOM 2010 as CSSMA Symposium Co-Chair and several best papers awards. Prof. Rodrigues is a licensed professional engineer (as senior member), member of the Internet Society, an IARIA fellow, and a senior member of ACM and IEEE.

André G. F. Elias a MSc and BSc degrees in Informatics Engineering from University of Beira Interior, Portugal, in 2012 and 2010, respectively. In 2011, he studied in the University of Campinas, Brazil, during one MSc semester. His research interests include mobile and ubiquitous computing, Web Services, and wireless sensor networks. He is member of the Next Generation Networks and Applications Group (NetGNA) at Instituto de Telecomunicações, Portugal. He authored or co-authored several papers in international journals and conferences.

Guangjie Han is currently a Professor of Department of Information & Communication System at Hohai University, China. He is also a visiting research scholar of Osaka University from Oct. 2010 to Oct. 2011. He finished the work as a post doctor of Department of Computer Science at Chonnam National University, Korea, in February 2008. He worked in ZTE Company from 2004 to 2006, where he held the position of Product Manager. He received his Ph.D. degree in Department of Computer Science from Northeastern University, Shenyang, China, in 2004. He has published over 110 papers in related international conferences and journals. He has served in the editorial board of up to 13 international journals, including Journal of Internet Technology and KSII Transactions on Internet and Information Systems. He has served as a Co-chair for more than 20 international conferences/workshops; a TPC member of more than 50 conferences. He holds 49 patents. He has served as a reviewer of more than 50 journals. He had been awarded the ComManTel 2014 and Chinacom 2014 Best Paper Awards. His current research interests are Sensor Networks, Computer Communications, Mobile Cloud Computing, Multimedia and Security. He is a member of IEEE and ACM

Chapter 6

Denial of service mitigation approach for IPv6-enabled smart object networks

This chapter consists of the following paper:

Denial of service mitigation approach for IPv6-enabled smart object networks

L. Oliveira, J. Rodrigues, A. de Sousa and J. Lloret
Concurrency and Computation: Practice and Experience, vol. 25, no. 1, pp. 129-142, 2012.

DOI: 10.1002/cpe.2850

According to Journal Citation Reports published by Thomson Reuters, this journal scored ISI journal performance metrics as follows:

ISI Impact factor (2013): 1.064

Article Influence Score (2013): 0.310

Journal Ranking (2014): 65/105 (computer science, software engineering)
Journal Ranking (2014): 50/102 (computer science, theory and methods)

CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2013; **25**:129–142
Published online 12 June 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cpe.2850

SPECIAL ISSUE PAPER

Denial of service mitigation approach for IPv6-enabled smart object networks

Luís M. L. Oliveira [1,2,3], Joel J. P. C. Rodrigues [1,2,*,†], Amaro F. de Sousa [1,4] and Jaime Lloret [5]

[1] *Instituto de Telecomunicações, Portugal*
[2] *Department of Informatics, University of Beira Interior, Covilhã, Portugal*
[3] *Polytechnic Institute of Tomar, Portugal*
[4] *Department of Electronics, Telecommunications and Informatics, University of Aveiro, Portugal*
[5] *Integrated Management Coastal Research Institute, Universidad Politécnica de Valencia, Spain*

SUMMARY

Denial of service (DoS) attacks can be defined as any third-party action aiming to reduce or eliminate a network's capability to perform its expected functions. Although there are several standard techniques in traditional computing that mitigate the impact of some of the most common DoS attacks, this still remains a very important open problem to the network security community. DoS attacks are even more troublesome in smart object networks because of two main reasons. First, these devices cannot support the computational overhead required to implement many of the typical counterattack strategies. Second, low traffic rates are enough to drain sensors' battery energy making the network inoperable in short times. To realize the Internet of Things vision, it is necessary to integrate the smart objects into the Internet. This integration is considered an exceptional opportunity for Internet growth but, also, a security threat, because more attacks, including DoS, can be conducted. For these reasons, the prevention of DoS attacks is considered a hot topic in the wireless sensor networks scientific community. In this paper, an approach based on 6LowPAN neighbor discovery protocol is proposed to mitigate DoS attacks initiated from the Internet, without adding additional overhead on the 6LoWPAN sensor devices. Copyright © 2012 John Wiley & Sons, Ltd.

Received 29 January 2012; Revised 22 March 2012; Accepted 2 April 2012

KEY WORDS: wireless sensor networks; low-power personal area networks; denial of service attacks; 6LoWPAN neighbor discovery; Internet of Things

1. INTRODUCTION

Nowadays, there is a growing tendency to embed computation and wireless communication devices on quotidian objects, transforming them into smart objects. These objects will collect and process information from different sources to both control physical processes and to interact with human users [1]. The embedded computational and communication devices are characterized by small size, power constrained, small computing, and storage resources and by reduced radio ranges and throughput [2, 3]. Networks composed of several connected smart objects are designated as low power over wireless personal area networks (LoWPAN). The provisioning of reliable energy-efficient and low-delay communications in resourced constrained network has become a challenging resource issue. A layered multipath power control scheme is proposed in [4], which has high performance on reliability, energy-efficiency, and low-delay communication in underwater sensor networks. Multiple-path forward error correction approach [5] based on Hamming codes, can also be used to improve the reliability and energy efficiency.

*Correspondence to: Joel J. P. C. Rodrigues, Department of Informatics, University of Beira Interior, Covilhã, Portugal.
†E-mail: joeljr@ieee.org

Wireless sensor networks are a subtype of smart object networks, where the devices can interact with their environment by sensing and controlling physical parameters, such as temperature, humidity, and solar radiation. A single network may comprise hundreds of smart devices working together to accomplish a common task. Self-organization, fault-tolerance, and self-optimization are the main characteristics of smart object networks [2]. Currently, there are already many technologies that can be used to connect smart objects [3], most of them based on the standard IEEE 802.15.4 layer two protocol [6] but some being proprietary, such as ZigBee [7] and WirelessHART [8]. Nevertheless, these solutions are not compatible with IP protocol and consequently require complex gateways to connect them to the Internet. The aim is that, in a near future, users can access the information collected by smart objects from the Internet, using regular devices and standard protocols. To reach this aim, a new paradigm is necessary to enable smart objects to be accessed from the Internet where all devices and networks are IP-enabled, independently of their physical and media access control (MAC) layer protocol [1]. The support of Internet protocol (IP) in all smart devices will also simplify the application development because tools in use on regular computing for commissioning, configuring, managing, and debugging can be used or adapted. Initially, the IP protocol stack was considered too heavy to run on small power and resource constrained devices. Meanwhile, the scientific community, together with the industry, started to rethink many misconceptions about the use of IP in all devices and now the IPv6 protocol is considered the most consensual solution to connect the smart objects to the Internet [9]. Nevertheless, IPv6 was not designed to be used in low power and resource constrained objects. The 6LoWPAN [10, 11] adaptation layer was defined between the data link layer and the network layer to enable the use of IPv6 protocol over IEEE 802.15.4 data link layer. Together with 6LoWPAN, other new protocols best fitted to low power and resource constrained devices were defined, such as routing and neighbor discovery protocols. In fact, the protocols designed to run over LoWPAN networks must have low overhead on data packets and on message exchange, minimal memory and computation requirements and support for sleeping nodes considering battery savings [10]. Neighbor Discovery (ND) is one of the most important protocols, because it is used by the nodes to discover each other's presence on the same link, to determine each other's link-layer addresses, to find routers and to maintain reachability information about the paths to active neighbors [12]. Concerning the ND protocol, it can be supported on the physical, data link, or network layers [13]. In this work, we only consider ND protocol on the network layer.

The original IPv6 ND protocol not only was not designed for nontransitive wireless links but also requires multicast transmission, a feature not supported by the IEEE 802.15.4 standard. As a consequence, it was necessary to optimize the IPv6 ND to fit to LoWPAN. The adapted ND protocol [12] supports sleeping hosts, eliminates the multicast-based address resolution for hosts, because it defines a registration feature that provides multihop prefix and header compression context and optional multihop duplicate address detection.

Connecting smart object networks to the Internet can be considered simultaneously an opportunity and a challenge [9, 14]. It is an opportunity because more services can be provided. It is a challenge because the smart object networks are now exposed to more security issues [15–17] because successful security attacks can now be initiated from anywhere. The security is even more alarming if the smart object networks are used to support critical infrastructures, such as smart grid applications or fire detection. Supporting security services on resource constrained devices is even more challenging because of the overhead introduced. Denial of service (DoS) and distributed denial of service (DDoS) can be done locally and remotely and are one of the most common types of security attacks, because usually they only require regular and inexpensive resources and do not require high technical skills [18]. This paper proposes a new mechanism to be supported only on edge routers, based on the ND messages exchanged by the LoWPAN devices and the edge routers, to mitigate the remotely initiated DoS and DDoS attacks.

The remainder of this paper is organized as follows. Section 2 analyses the IPv6 enabled smart object networks, while Section 3 focuses on security attacks for wireless sensors with IPv6 end-to-end connectivity support. The Sections 4 and 5 present a new countermeasure mechanism based on 6LoWPAN neighbor discovery to mitigate network and transport layer remotely initiated DoS attacks and discuss its application. Finally, Section 6 concludes the paper and pinpoints future research topics.

DOI: 10.1002/cpe

2. IPV6 ENABLED SMART OBJECT NETWORKS

IEEE 802.15.4 [6] is a data link layer standard specified to address the low-power and low-rate wireless personal area networks requirements. Two types of devices were defined: full-function devices (FFD) and reduced-function devices (RFD). FFD devices support all network functionalities and can support peer-to-peer topologies because of their multihop routing capabilities [19]. RFD devices support only a limited set of functionalities and are mainly used for sensing and/or actuation operations. Multihop communications are not supported by RFD and, thus, they can only be used in star topologies. The protocol defines a central controller device, referred to personal area network (PAN) coordinator, which builds a wireless PAN (WPAN) with other compliant devices. The PAN coordinator starts a new network by selecting a suitable channel according to energy detection scanning, which measures the interference of each channel. After the channel selection, the PAN coordinator broadcasts periodically a beacon to announce the WPAN configurations. The other nodes start listening to the beacons to search for available WPAN and to select a coordinator. Only FFD devices can operate as PAN coordinators. Two topologies are supported. In a star topology network, all communications go through the PAN coordinator (i.e., all nodes, except the PAN coordinator, can be RFD devices). In a peer-to-peer topology, devices can communicate with one another directly, but still the PAN coordinator has to exist [20].

The IEEE802.15.4 protocol defines the physical (PHY) and the MAC layers. The PHY layer defines three physical operation modes, 20 kb/s at 868 MHz, 40 kb/s at 915 MHz, and 250 kb/s at 2.4 GHz (DSSS). The MAC layer provides two operational modes: the asynchronous beaconless and the synchronous beacon-enabled mode. The beacon-enabled mode is designed to support the transmission of beacon packets between transmitter and receiver, providing synchronization among nodes. In the beacon-enabled mode, the beacon periodically broadcasted by the PAN coordinator contains information about the PAN. In this mode, the period between two consecutives beacons defines a superframe structure that is divided into 16 slots. Beacons always occupy the first slot, while the other slots are used for data communications. In these slots, slotted carrier sense multiple access with collision avoidance is used for data transmission. To support low-latency applications, the PAN coordinator can reserve one or more slots, designated by guaranteed time slots, which are assigned to devices running such applications (in this case, these devices do not need to use contention based medium access mechanisms) [21]. In the beaconless mode, there is no superframe structure and no guaranteed time slots. As a consequence, only random access methods, such as unslotted carrier sense multiple access with collision avoidance, can be used to medium access. The frame length is limited to 127 bytes because unreliable and error prone wireless links are used and the devices have limited buffering capabilities.

2.1. 6LoWPAN adaptation layer

Currently, the IEEE 802.15.4 protocol is widely accepted as the PHY and MAC layer protocol to be used on smart object networks. However, the WPAN constraints do not permit to support IPv6 directly over IEEE 802.15.4 [10]. The maximum link-layer packet size of 127 bytes is one of the most obvious limitations because implementing standard IPv6 headers over LoWPAN would result in extremely small payloads for higher-layer protocols. In the best case, the maximum size of an IP packet is 88 bytes; the IPv6 header has a minimum size of 40 bytes, which results in 48 bytes for upper-layer protocols like Transmission Control Protocol (TCP) or User Datagram Protocol (UDP); the length of the TCP header is another 20 bytes, which results in 28 bytes available for the application-layer protocol (in the TCP case). To circumvent this problem, the Internet Engineering Task Force (IETF) created the 6LoWPAN working group with the aim of defining the support of IPv6 over IEEE 802.15.4 LoWPAN networks. To comply with the maximum transmission unit requirements of IPv6 protocol and to minimize the overhead, 6LoWPAN [11] introduces an adaptation layer between data link and network layers. This layer provides a mechanism for packet fragmentation, header compression, and support for data link layer forwarding of IP packets, also known as mesh-under routing. Although 6LoWPAN was originally designed to support IPv6 over IEEE 802.15.4, it can later be adapted for other similar link technologies.

DOI: 10.1002/cpe

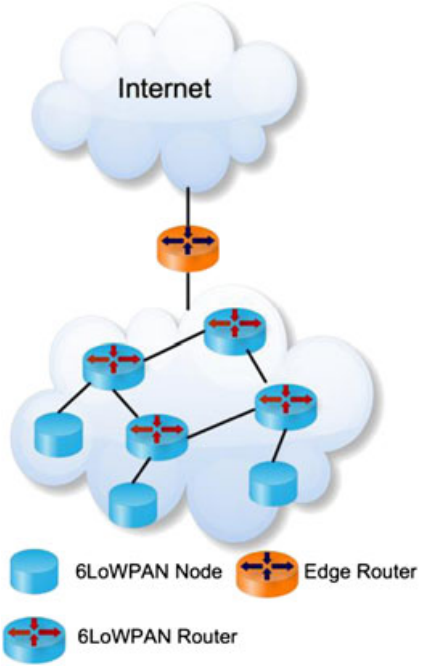


Figure 1. Illustration of 6LoWPAN network architecture.

In LoWPAN networks, packets will often have to use multiple radio hops to reach the destination. The multihop forwarding is motivated by the fact that the sending node may not have radio range to reach the destination node. To send a packet to another node, two main processes are involved: forwarding and routing. On the forwarding process, packets are moved from the input to the output interface and are executed at lower layers. Note that many times, a single physical interface is involved in the forwarding process. The routing process uses a routing protocol to evaluate the best path to reach the destination. Each node maintains a routing information base that contains all the information needed to run the routing protocol. The routing information base is used to fill the forwarding information base, which is consulted when a packet needs to be forwarded. Routing in a 6LoWPAN network can be done in three different ways: link-layer mesh-under, 6LoWPAN mesh-under, and route-over [21, 22]. Link-layer mesh-under and LoWPAN mesh-under are designated by mesh-under and are transparent to the network layer. Routing at network layer is designated by route-over.

A typical LoWPAN consists of edge routers, routers, and nodes (Figure 1). 6LoWPAN nodes usually perform only sensing and actuation operations. They send their own datagrams to other destination nodes and receive datagrams from other destination nodes but they do not forward datagrams originated on other nodes and destined to other nodes. Routers are intermediate nodes that can be used to forward datagrams to others nodes or routers in the same LoWPAN and are present only in route-over topologies. Edge routers are used to connect the LoWPAN to other networks, for example, the Internet. Typically, nodes and routers have energy and computational resources constraints and only the edge routers are main powered with more computational resources [10].

2.2. Neighbor discovery protocol for 6LoWPAN

The IPv6 ND protocol is used by the nodes on the same link to discover each other's presence, to determine each other's layer two addresses, to search routers, to maintain reachability information about the paths to active neighbors, and to address auto configuration [12, 23].

The original IPv6 ND protocol was not designed for nontransitive wireless links, making heavy use of multicast, which is inefficient and impractical in low-power networks, because broadcast is used in absence of multicast support. As a consequence, the rate of ND transmitted messages is

DOI: 10.1002/cpe

limited because of energy conservation policies. Also, IPv6 ND assumes that local link nodes are always a single hop away and nodes are always listening, but in LoWPAN networks this is not the case.

Although the standard IPv6 ND protocol should work on 6LoWPANs, the resource constraints of LoWPAN nodes, their absence of multicast support at layer two and their low duty-cycle requires a different approach for the ND protocol on 6LoWPANs, focusing on the efficient use of available energy.

Neighbor discovery optimizations for 6LoWPAN [12] are being proposed to address the specific needs of LoWPAN. Neighbor discovery optimization for low power and lossy networks (draft-ietf-6lowpan-nd-18) [12] is a work in progress specification proposed by IETF's 6LoWPAN Working Group. It describes optimizations to the IPv6 neighbor discovery, header compression context information dissemination, auto configuration addressing mechanisms, and duplicate address detection for low power networks. The neighbor discovery signaling was simplified by replacing the address resolution process with an address registration mechanism. It also eliminates the need for periodic router advertisement multicasting, by providing host-initiated request for router advertisements. Moreover, in most cases multicast messages were replaced by unicast messages. The node to router 6LoWPAN ND message exchange is not affected by the routing approach and, as a consequence, the protocol behavior is the same both in mesh-under and route-over configurations.

The edge router, designated in [12] as 6LBR, plays an important role in 6LoWPANs. Besides being responsible for connecting the LoWPAN to the Internet, it is also responsible for propagating the IPv6 prefix and header compression context information across the LoWPAN network. The 6LBR also maintains a network-wide cache of the hostsIPv6 addresses and 64-bit extended unique identifier (EUI-64), which makes it able to make layer two address resolution and detect and avoid duplicate addresses. Alternatively, DHCPv6 can be used to ensure unique addresses on the network. 6LoWPAN neighbor discovery assumes each IPv6 is derived from the unique EUI-64 address, so it does not require, by default, either duplicate-address detection or address resolution if the IPv6 link-local addresses are used [24]. There are also optional and separated mechanisms that can be used between LoWPAN routers (6LR) and 6LBR to execute multihop duplicate address detection and distribution. These optimizations lead to a significant drop in signaling messages in the local network, resulting in significant energy savings, extending the lifetime of the network.

To achieve these goals, the new ND protocol defines three new ICMPv6 message options: the required address registration option (ARO) and the optional authoritative order router option (ABRO), and 6LoWPAN Context Options (6CO).

Two new ICMPv6 message types are also defined to carry out the optional multihop duplicate address detection: duplicate address request (DAR) and duplicate address confirmation (DAC).

The nodes in a LoWPAN network use ND to perform address auto configuration, layer two address resolution, neighbor unreachability detection, and to find default routers.

When the interface on a node device is initialized, a link-local address is formed based on the EUI-64 identifier. Next, the device nodes send a router solitication message including the source link-layer address (SLLA), so that the router can reply with a unicast router advertisement message. The router advertisement message can include the SLLA, authoritative order router option, 6CO, and the IPv6 Prefix Option (Figure 2). Once an address has been configured in a node, a neighbor

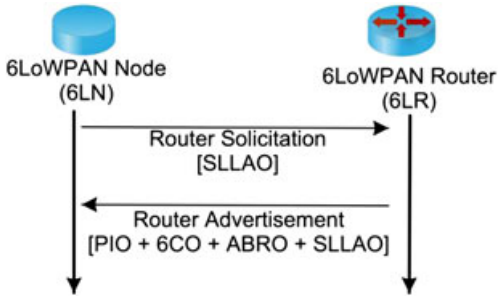


Figure 2. Host initiated router discovery.

DOI: 10.1002/cpe

solicitation (NS) message with an address registration option is sent to the edge router to register that address.

The process of address registration (Figure 3) is necessary to avoid the layer-two address layer resolution based on multicast neighbor solicitation messages. The host sends a unicast NS message to the router, with the ARO. The router replies with a unicast neighbor advertisement (NA) message with the ARO and the status of the registration. The status indicates either a successful registration or a failure (either because of a duplicated address or because the router's registration cache is full).

The address registration mechanism and the SLLA router advertisement option provide enough information in routers and nodes to resolve an IPv6 address to its associated layer two addresses. Note that all prefixes, except the link-local addresses, are always assumed to be off-link, so all communications must be through the edge router. The multicast addresses are also supposed to be off-link, because multicast-based addresses resolution between neighbors is not needed. The information transported on NA messages have a lifetime associated and the node must repeat the above described process before the lifetime expires. Note that nodes can receive router advertisements messages from multiple edge routers. In this case, they should attempt to register with more than one router to increase the network resilience.

The node device also uses neighbor solicitation messages to perform unreachability detection. This operation is mainly used to verify the default router reachability.

The optional multihop duplicate address detection process is shown in Figure 4. It can be used in route-over networks to assure address uniqueness within the 6LoWPAN for non-EUI-64 based addresses. It is similar to the standard address registration process, except that because the edge router is responsible for managing the address registration cache, the intermediate router that the host tries to register with must first check with the 6LBR if the address is not duplicated. This is carried out using the new DAR and DAC ICMPv6 messages.

An edge router does not need to send unsolicited router advertisement messages, because the node devices will send router solicitation messages whenever they need updated information. Unicast neighbor advertisement messages are always used in response to neighbor solicitation messages.

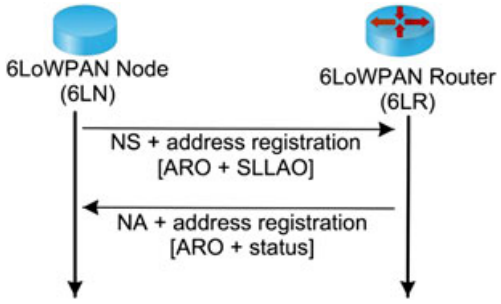


Figure 3. Node address registration.

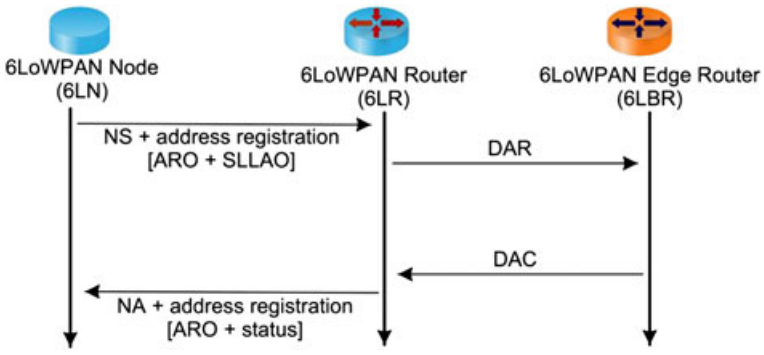


Figure 4. Host address registration with multihop Duplicate Address Discovery (DAD).

DOI: 10.1002/cpe

2.3. Connectivity models

Three main models can be considered concerning the connection of LoWPAN networks to the Internet. In the first model, all LoWPAN nodes support the IP protocol stack but they are not connected to the Internet [25]. In fact, there are several scenarios that do not require any connectivity with the Internet, as for example the smart grid applications. Smart grid networks are used to monitoring the power generation networks, the automation and control devices, smart metering, and building and home energy management. These networks can also use the IP protocol suite in all nodes but because of security and privacy reasons, in most of the cases they are completely disconnected from the public Internet. In this case, the assignment of global IPv6 addresses to all devices is not desirable.

In the second model, a proxy device is used to connect the smart object network to the Internet. Internet users will have access to the information provided by smart objects, such as environmental data, using the proxy device. The proxy can act as a server that collects data from the smart objects. This connectivity model can be used to connect networks without IP support, to preserve scarce resources on such networks and to increase scalability, although it does not provide end-to-end connectivity. Supporting more than one point of connection between the smart object network and the Internet is not possible if the proxy uses stateful translation mechanisms. This connectivity model is similar to the previous model. Therefore, the support of IP protocol stack continues to represent a benefit, but assigning IP global addresses to all devices is optional. The second model can be considered an intermediate model between the first model and the smart object fully integrated in the public Internet.

In the third model (Figure 1), the smart object networks are considered as an extension to the Internet. This connectivity model can be used in the near future to support services provided by smart cities, where the citizens can use the Internet to make quotidian decisions based on environmental data such as air quality, temperature, and real-time transportation information. All of these networks will make use of the IP protocol stack and more than one router can be used, for redundancy and scalability purposes, to connect these networks to the Internet. In such model, the IP end-to-end connectivity is required and, at least, one IP global address must be assigned for each device.

3. SECURITY ATTACKS IN SMART OBJECT NETWORKS

Protecting the resources and the information transmitted over the network from attacks is the main concern of the security services [15–17]. Besides the differences between smart object networks and the other network types, both share some security requirements but, because of resource constraints and the number of nodes, providing security services in smart object networks is even more challenging when compared with standard networks. Confidentiality, integrity, availability, freshness, robustness, and survivability are the most relevant security requirements in smart objects networks [26–28]. Confidentiality ensures that no other than the legitimate entities have access to the data transmitted and stored in the smart object network. Authenticity is a central concept to confidentiality, because it ensures that the identity of the sender is correct (authentication is also necessary for automated node interactions). Integrity ensures that no message can be altered by any entity, without being detected, as it transverses from the sender to destination. Availability ensures that services provided by the smart object network are always available to be used by the legitimate users. Data freshness prevents other parties from replaying old messages. There are two types of data freshness requirements, strong and weak data freshness. In the first type, it guarantees data framing ordering and delay. In the second type, a partial message ordering is provided, but does not guarantee delay. Robustness and survivability is the guarantee that the network remains operational even if a set of nodes are compromised because of a security attack.

Smart object networks are vulnerable to several types of security attacks, which can be classified, according to security requirements in three main groups [15]: attacks on secrecy and authentication, attacks on network availability, and stealthy attacks against service integrity. Eavesdropping, packet reply attacks, tampering and spoofing of packets are examples of attacks against the secrecy and authenticity. Several mechanisms can be used to prevent attacks on secrecy and authenticity, most of

DOI: 10.1002/cpe

which are based on cryptography. Device data confidentiality and integrity are harder to obtain when compared with communication confidentiality because they require both logical and physical measures to protect against attackers. Introducing false data into the smart object network is the main goal of stealthy attacks against service integrity. Attacks on network availability are often designated as DoS attack. This paper focus on DoS and their countermeasures, in particular to those that can be used to prevent remote initiated attacks.

A DoS attack is characterized by an explicit attempt to prevent the network to perform its expected functions [29]. During a DoS attack, the attacker attempts to reduce the network's capacity. Several strategies can be used to perform a DoS attack. Flooding the network with junk traffic or disrupting network connections are two of the most common techniques. DoS attacks can be classified as logic attacks and resource exhaustion flooding attacks. Logic attacks exploit security vulnerabilities to cause a server or service to crash or significantly reduce its performance. Resource exhaustion flooding attacks cause the network nodes or network resources to be consumed to the point where the service is no longer responding or the response is significantly reduced [15]. When a DoS is originated from several sources, it is designated as a DDoS attack. In both cases, the attack sources can be locally or remotely located. The techniques that can be used to perform a DoS attack can be classified according to the protocol layer that is to be attacked [15, 29]. Jamming and tampering are the most common strategies against the physical layer. The jamming is intended to interfere with the normal radio communication link where the attacker uses the same spectrum that legitimate network nodes are using. Defenses against jamming involve code spreading and frequency hop techniques.

The link layer is responsible for medium access control, error detection, frame construction and detection, and reliable point-to-point and point-to-multipoint connections between adjacent nodes. Forcing frame collisions can be used to achieve resource exhaustion and unfairness and it is the main technique to perform a DoS attack on link layer protocols. Using small frames, error-correction codes and rate limitation are three of the most used mechanisms to mitigate link layer DoS attacks.

In smart object networks, the routing can be performed either on link layer (mesh-under approach) or at network layer (route-over approach). Therefore, DoS attacks directed to routing information protocols can point to both layers. Creating loops and attracting (or repelling) network traffic from selected nodes are the main strategies of DoS attacks directed at routing protocols. Adding message authentication to routing information messages is one of the main countermeasure techniques, because the receivers can detect if the messages have been tampered or spoofed [30, 31].

Managing end-to-end connections is the transport layer main function. Flooding and desynchronization are two of the possible attacks in this layer [15]. In flooding attacks, several new connection requests are sent until the exhaustion of the receiver resources. To avoid this attack, it is necessary to identify the legitimate requests to avoid wasting resources with bogus connections. The desynchronization attack refers to the disruption of an existing connection. This attack uses spoofed messages causing the retransmission of missing frames because of errors that have never really existed. Puzzle resolution and authentication techniques are the most common countermeasures to prevent transport layer DoS attacks [28, 32]. Note that the UDP protocol is much more used in smart object networks than TCP and, therefore, flooding and desynchronization attacks are not so disruptive. However, any unnecessary transmitted message has an important impact on the energy consumption and UDP is harder to control when end-to-end connections between the smart object and the Internet are supported.

DoS attacks can also be directed to the application layer protocols. Application layer DoS attacks are even more difficult to detect because the transport layer connection is valid and so are the requests. During the attack, one or more clients send a large number of requests reducing drastically the server processing capability. Defending against application layer DoS attacks usually involves some sort of rate-shaping algorithm that monitors client's behavior and ensures that they request no more than a configurable number of requests per time period. If the client generates requests more than the configurable number, the client's IP address is blacklisted for a specified time period and subsequent requests are denied until the address has been released from the blacklist.

In smart object networks, the adequate entity to implement the previous described countermeasures is the edge router for two reasons. First, the edge routers have more energy and computation

DOI: 10.1002/cpe

resources than smart object nodes. Second, it makes more sense to filter the traffic closer to the source.

In the third connectivity model presented in Section 2.3, the smart object networks are connected to the Internet just like any other network. Any Internet user, potentially, have access to the information provided by smart objects accessing the device. This connectivity model can be used to support a myriad of new services and applications. However, the smart object network becomes also exposed to remotely initiated security attacks, in particular to DoS and DDoS.

4. MITIGATION OF DOS ATTACKS ON WIRELESS SENSOR NETOWRK WITH IPV6 END-TO-END CONNECTIVITY

This section proposes a countermeasure mechanism based on 6LoWPAN neighbor discovery to mitigate remotely initiated DoS and DDoS attacks. The proposed security mechanism runs only on the edge routers, not overloading the smart object nodes. It reuses the registration address process messages and protects the wireless sensor networks against transport and application layer DoS and DDoS attacks, filtering unsupported traffic at the edge and rate-shape the requests from the Internet to ensure that any Internet client generates no more requests than the imposed limits.

4.1. Proposed mechanism

As explained in Section 2.2, the address registration process is necessary to avoid the layer-two address layer resolution and to guarantee the node's IP address uniqueness. Depending on the routing approach, two different procedures can be used to perform the address registration [12]. In the mesh-under routing approach, the nodes exchange the NS and the NA messages with the edge router. In the route-over routing approach, the process is similar to the one for mesh-under between the nodes and the 6LRs and, additionally, the 6LR uses the new DAR and DAC messages to verify the address uniqueness on the edge router.

Note that the current ARO option contains two fields reserved for future use, the first with 8 bits and the second with 16 bits length. Moreover, the DAR messages also contain an 8-bit length reserved field. We propose the use of the 8-bit length reserved fields of both cases to implement the security mechanism. The new information to be included on these field is: (i) the transport-layer protocol switch are to be accepted, (ii) the reachability acceptance from the Internet, and (iii) the maximum Internet clients request rate-shape limit. Figure 5 presents the proposed format for the new ARO and DAR messages. Table I presents the proposed valid values and the description of each new value.

Three new data structures are created at the edge routers: the filtering database, the Internet client's address table, and the Internet client blacklist table.

Information extracted from the new ARO and DAR messages are used to fill the filtering database, according to the correspondence defined in Table II. The filtering database is used to filter unwanted traffic and it is composed of the node's IP address (IP address), the registered lifetime (Lifetime), the reachability acceptance from the Internet (Accept data from the Internet (AFI)),

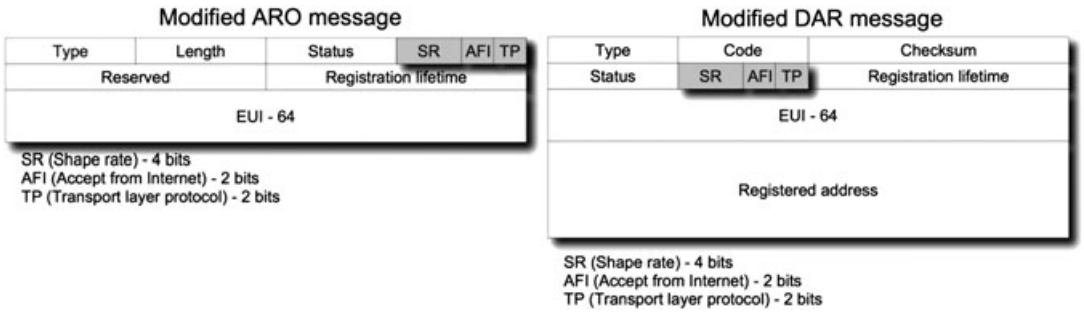


Figure 5. New ARO and DAR message formats.

DOI: 10.1002/cpe

Table I. ARO and DAR new data fields.

Field	Length	Values	Description
Shape rate	4 bits	0000	Not used
		0001–1111	Rate limit value
AFI	2 bits	00	Not used
		01	Do not accept packets from the Internet
		10	Accept packets from the Internet
		11	To be defined
TP	2 bits	00	Not used
		01	UDP
		10	TCP
		11	Accept any

AFI, Accept from Internet; TP, Transport layer protocol.

Table II. Filtering database fields correspondence.

Filtering database fields	ARO message fields	DAR message fields
IP address (128 bits)	EUI-64	Registered address
Lifetime (16 bits)	Registration lifetime	Registration lifetime
Accept data from Internet (2 bits)	Accept data from Internet	Accept data from Internet
Accepted transport layer protocol (2 bits)	Accepted transport layer protocol	Accepted transport layer protocol
Rate request limit (4 bits)	Rate request limit	Rate request limit

IP address (128 bits)	Lifetime (16 bits)	Accept data from the Internet (2 bits)	Accepted transport layer protocol (2 bits)	Rate request limit (4 bits)

Figure 6. Filtering database table format.

Client IP address (128 bits)	Lifetime (16 bits)	IP destination address (128 bits)	Rate request (4 bits)	Rate request limit (4 bits)

Figure 7. Internet client address table.

Client IP address (128 bits)	Lifetime (16 bits)	IP destination address (128 bits)	Counter

Figure 8. Internet client blacklist table.

the accepted transport layer protocol (Accepted transport layer protocol (TP)) and the Internet client rate request limit (Rate request limit (SR)) (Figure 6).

The Internet client address table is used for ensuring that no Internet client generates more packets than the imposed limits. Limits per client and per node will be applied. As may be seen in Figure 7, this table is composed of the Internet client IPv6 address (Client IP address), lifetime (Lifetime), smart object IP address (Destination address), the rate packet computed per minute (Rate request), and the rate request limit (Rate request limit) copied from the filtering database table.

The Internet client blacklist table is used to store the Internet client's IP address that exceeds the imposed rate limits (Figure 8) and comprise the following fields: Internet client's IP address (IP address), the configurable amount of time in seconds that the IP address must remain in the blacklist (Lifetime), IP address of the destination node (IP destination address), and the number of times that this address was added to the blacklist (Counter). The Lifetime value must be increased if the same client IP address repeats several times for the same or for different destination address.

DOI: 10.1002/cpe

Therefore, the blacklist table entries should not be removed after the lifetime goes to zero. However, the oldest entries must be periodically flushed.

4.2. Mesh-under networks

In the mesh-under routing approach [15], nodes register the address directly on the edge router. When a node has configured a nonlink local IPv6 address, it registers that address in one or more edge router, using the NS message with ARO option. Besides the behavior defined in the 6LoWPAN neighbor discovery working progress document, the node also adds to ARO information related to the new fields (i.e., SR, AFI, and TP). If these values are equal to zero, the edge router handles neighbor solicitation message as specified in the 6LoWPAN neighbor discovery working progress document. If the new fields are different from zero, and in addition to the normal behavior, the new field values are copied into the filtering database table according to Table II. The edge router should ignore the new ARO fields if the new format is not supported.

4.3. Route-over networks

In the route-over routing approach [15], the ARO is used to register an address in a 6LR (6LoWPAN router). In this case, the 6LR reuses the information contained in the ARO, sent by the node, in the DAR message (Figure 4). Therefore, in addition to the normal operation defined in the 6LoWPAN neighbor discovery working progress document, the 6LR must copy the new fields (i.e., SR, AFI, and TP) from the ARO message into the new DAR message (Figure 5) before sending it to the edge router. The edge router updates the filtering database table according to the correspondence defined in the Table II. The 6LR should ignore the new DAR fields if the new format is not supported.

4.4. Filtering packets received from the Internet

When the edge router receives a packet from the Internet destined to an address of the smart object network, it must first verify if the destined address exists, if the destination node accepts the transport layer protocol of the packet, and if the packet IP source address is not present in the Internet client blacklist table with lifetime value greater than zero. Then, the packet is forwarded, using the regular routing mechanisms, if the previous mentioned conditions are true or discarded, otherwise. Internet client's address and Internet client blacklist tables are updated for each packet received from the Internet.

5. DISCUSSION OF THE PROPOSED SOLUTION

Denial of service and DDoS can be carried out locally and remotely, and they are among the most common types of security attacks, because they require only regular and inexpensive resources, and do not require high technical knowledge. The frequency and sophistication of DoS and DDoS are rapidly increasing based on several techniques including direct attacks, remote controlled attacks, reflective attacks, worms, and viruses.

Although there are several techniques to prevent or to mitigate DoS attacks, a generic defense mechanism against these security attacks is considered a research open issue. Furthermore, most of the proposed defense mechanisms require high computational resources making them inappropriate to be used on smart object networks. DoS is even more destructive to smart object networks when compared with other networks. First, it is easier to exhaust resources on constrained networks. Second, sensors energy can be rapidly consumed making them unavailable until the attack is ended and the battery is recharged.

The proposed security mechanism prevents smart object networks from remotely initiated DoS (and DDoS) network and transport layer attacks. The mechanism filters unwanted traffic originated on the Internet and destined to the smart object network nodes and it is based on the address registration process defined in the ND protocol proposed for 6LoWPAN. With this mechanism, the traffic is forwarded from the Internet to the smart object networks only if it is in accordance with the following rules:

DOI: 10.1002/cpe

- The destination node address must be registered; this condition guarantees that traffic is not forward to nonexisting nodes.
- The nodes must previously declare willingness to accept data from the Internet; in this way, nodes that make no sense to be addressed from outside will not be reached as, for example, the 6LR routers.
- Information about the node's supported transport layer protocol must be previously registered on the edge router; in this way, only traffic of such protocols will be forwarded.
- Nodes should previously inform the edge router about the accepted traffic rate limit; in fact, in most sensor cases, measurements data is generated at a slow acquisition rate(for example, air temperature monitoring), which puts a limit on acceptable request rates preventing, in this way, flooding attacks.

To implement the proposed mechanism, it is only necessary to define three fields in ARO and DAR messages. These fields do not increase the length of the messages because they use already existing 8-bit length reserved fields. Moreover, the mechanism does not increase the overhead on the resource constrained nodes (i.e., smart object nodes and 6LoWPAN routers) because the filtering mechanisms and all processing (and storing) overhead run only on the edge routers, which have less resource constraints. The proposed mechanism uses stateless traffic processing, so it can run simultaneously in different edge routers, providing more robustness to the network. In the original ARO and DAR messages, the zeros are used to fill the reserved data fields. As a consequence, the compression rates are not compromised on the new messages because different values are used on the same fields [33].

6. CONCLUSIONS AND FUTURE WORK

Smart object networks, which include wireless sensor networks, can provide support for numerous applications. In fact, the sensors give the smart objects the capacity to sense the physical world and to control some physical processes because of actuation capabilities. There are already a number of emerging applications of smart objects in power grid monitoring and control, e-health, intelligent transport systems, environmental monitoring, and energy management. So far, the smart object networks are isolated from the Internet because of two reasons. First, a large number of technologies is used and some of them are incompatible with IP protocol. Second, the security problems because of outside attacks are not an issue.

Providing security services in smart object networks connected to the Internet is considered an open issue. Providing security in resource constrained network is even more challenging when compared with standard networks. Therefore, special protocols and mechanisms have been developed for use in smart object networks. The frequency and sophistication of DoS attacks are rapidly increasing.

This paper has presented a security mechanism to prevent remotely initiated transport level DoS attacks. The proposed mechanism filters at the edge router the traffic received from the Internet and destined to smart object nodes. The edge router only forwards the Internet traffic into the smart objects network if the traffic meets predefined conditions. In the proposed solution, smart nodes use an adapted version of 6LoWPAN neighbor address registration mechanism to inform the edge router about the conditions used to filter the Internet received traffic. In this mechanism, all the required information is carried on address registration messages and the edge routers are the only entities that are required to support storage and processing overhead. The proposed mechanism requires no additional messages than those used to perform the address registration and also does not increase the length of the messages.

The security model used in the proposed mechanism can also be used to enforce security services on the edge to provide confidentiality and authenticity based on cryptography. Internet client authenticity must be ensured to provide a more robust remote DoS attack control. Authentication and client puzzles based mechanisms [17, 28, 32, 34, 35] can be used in the edge router to provide a more coarse traffic admission control. Adding authentication, client puzzle mechanisms to the current

DOI: 10.1002/cpe

solution, providing more application-based control and conducting a performance evaluation in real scenarios will be addressed as future work.

ACKNOWLEDGEMENTS

This work has been partially supported by the *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal, and by National Funding from the FCT – *Fundação para a Ciência e Tecnologia* through the Pest-OE/EEI/LA0008/2011.

REFERENCES

1. Gershenfeld N, Krikorian R, Cohen D. The Internet of Things. *Scientific American* 2004; **291**(4):76–81.
2. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: a survey. *Computer Networks* 2002; **38**(4):393–422.
3. Karl H, Willig A. *Protocols and architectures for wireless sensor networks*. John-Wiley: New York, 2005. ISBN 978- 0470095102.
4. Xu J, Li K, Min G, Lin K, Qu W. Energy-Efficient Tree-based Multi-path Power Control for Underwater Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems,* **99**. DOI: 10.1109/TPDS.2012.49.
5. Xu J, Li K, Min G. Reliable and Energy-Efficient Multi-path Communications in Underwater Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems,* **99**. DOI: 10.1109/TPDS.2011.266.
6. IEEE Std 802.15.4-2006. Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications-for low-rate wireless personal area networks (LR-WPANs). 2006. IEEE Std. 802.15.4–2006.
7. ZigBee Alliance. ZigBee Specification, October 2007.
8. WirelessHARThomepage. January 2012. Available from: http://www.hartcomm.org/
9. Hui J, Culler D. Extending IP to Low-Power, Wireless Personal Area Networks. *IEEE Internet Computing* 2008; **12**(4):37–45.
10. Kushalnagar N, Montenegro G, Schumacher C. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. August 2007. Internet Engineering Task Force Request for comments 4919.
11. Montenegro G, Kushalnagar N, Hui J, Culler D. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. September 2007. Internet Engineering Task Force Request for comments 4944.
12. Shelby Z, Thubert P, Hui J, Chakrabarti S, Bormann C, Nordmark E. 6LoWPAN Neighbor Discovery. October 2011. Internet Engineering Task Force, IETF draft draft-ietf-6lowpan-nd-18 working progress.
13. Chang M, Chao C, Chen L, Lai F. An Efficient Service Discovery system for Dual-Stack Cloud File Q14 Q15 Service. *IEEE System Journal* 2011; **99**. DOI: 10.1109/JSYST.2011.2177131.
14. Zhou BL, Chao H-C, Vasilakos A. Joint Forensics-Scheduling Strategy for Delay-Sensitive Multimedia Applications over Heterogeneous Networks. *IEEE Journal on Selected Areas in Communications* 2011; **29**(7):1358–1367.
15. Roman R, Lopez J. Integrating Wireless Sensor Networks and the Internet: a Security Analysis. *Internet Research* 2009; **19**(2):246–259.
16. Yong W, Attebury G, Ramamurthy B. A survey of security issues in wireless sensor networks. *Communications Surveys & Tutorials, IEEE* 2006; **8**(2):2–23. DOI: 10.1109/COMST.2006.315852.
17. Du X, Chen H. Security in Wireless Sensor Networks. *IEEE Wireless Communications* 2008; **15**(4):60–66.
18. Pelechrinis K, Iliofotou M, Krishnamurthy V. Denial of Service Attacks in Wireless Networks: The Case of Jammers. *Communications Surveys & Tutorials, IEEE* 2011; **13**(2):245–257. DOI: 10.1109/SURV.2011.041110.00022.
19. Zhou CL, Wang X, Tu W, Mutean G, Geller B. Distributed Scheduling Scheme for Video Streaming over Multi-Channel Multi-Radio Multi-Hop Wireless Networks. *IEEE Journal on Selected Areas in Communications* 2010; **28**(3):409–419.
20. Lin K, Chin-Feng L, Xingang L, Xin G. Energy Efficiency Routing with Node Compromised Resistance in Wireless Sensor Networks. *ACM/Springer Mobile Networks and Applications* 2012; **17**(1):75–89. DOI: 10.1007/s11036-010-0287-x.
21. Hongjuan L, Lin K, Keqiu L. Energy-efficient and high-accuracy secure data aggregation in wireless sensor networks. *Computer Communications* 2011; **34**(4):591–597.
22. Oliveira L, Sousa A, Rodrigues J. Routing and mobility approaches in IPv6 over LoWPAN mesh networks. *International Journal of Communication Systems* 2011; **24**:1445–1466. DOI: 10.1002/dac.1228.
23. Narten T, Nordmark E, Simpson W, Soliman H. Neighbor Discovery for IP version 6 (IPv6). September 2007. Internet Engineering Task Force Request for comments 4861.
24. Singh H, Beebee W, Nordmark E. IPv6 Subnet Model: The Relationship between Links and Subnet Prefixes. July 2010. Internet Engineering Task Force Request for comments 5942.
25. Vasseur J, Dunkels A. *Interconnecting Smart Objects with IP*. Morgan Kaufmann: San Francisco, CA, USA, 2010. ISBN 978-0123751652.
26. Ramen R, Lopez J, Gritzalis S. Situation awareness mechanisms for wireless sensor networks. *IEEE Communication Magazine* 2008; **46**(4):102–107.

27. Sakerindr P, Ansari N. Security Services in Group Communications over Wireless infrastructure, Mobile Ad Hoc and Sensor Networks. *IEEE Wireless Communications* 2007; **14**(5):8–20.
28. Lopez J, Roman E, Alcaraz C. *Analysis of Security Threats, Requirements, Technologies and Standards in Wireless Sensor Network, Foundations of Security Analysis and Design*, LNCS 5705. Springer: Berlin/Heidelberg, 2009. 289–338.
29. Peng T, Leckie C, Ramamohanarao K. Survey of network-based defense mechanisms countering the DoS and DDoS problems. *ACM Computer Surveys* 2007; **39**(3):224–260. DOI: 10.1145/1216370.1216373.
30. Tsao T, Alexander R, Dohler M, Daza V, Lozano A. A Security Framework for Routing over Low Power and Lossy Networks. September 2009. Internet Engineering Task Force, draft draft-tsao-roll-security-framework-01.
31. Karlof C, Wagner D. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. *First IEEE International Workshop on Sensor Network Protocols and Applications*, Alaska, USA, May 2003; 113–127, DOI: 10.1109/SNPA.2003.1203362.
32. Newsome J. The Sybil Attack in Sensor Networks: Analysis and Defenses. In *3rd International Symposium on Information Processing in Sensor Networks (IPSN 2004)*. ACM: New York, NY, USA, Apr. 2004, DOI: 10.1145/984622.984660. (Availablefrom:http://doi.acm.org/10.1145/984622.984660).
33. Hui J, Thubert P. Compression Format for IPv6 Datagrams in 6LoWPAN Networks. October 2009. Internet Engineering Task Force,draft draft-ietf-6lowpan-hc-06 working progress.
34. Shi E, Perrig A. Designing Secure Sensor Networks. *Wireless Communications Magazine* 2004; **11**(6):38–43.
35. Akkaya K, Younis M. A survey of routing protocols in wireless sensor networks. *Elsevier Ad Hoc Network Journal* 2005; **33**:325–349.

Chapter 7

Security solutions for 6LoWPAN enabled NetworksA Network Access Control Framework for 6LoWPAN Networks

This chapter consists of the following paper:

A Network Access Control Framework for 6LoWPAN Networks

L. Oliveira, J. Rodrigues, A. de Sousa and J. Lloret
Sensors, vol. 13, no. 1, pp. 1210-1230, 2013.

doi: 10.3390/s130101210

According to Journal Citation Reports published by Thomson Reuters, this journal scored ISI journal performance metrics as follows:

ISI Impact factor (2013): 2.048

Article Influence Score (2013): 0.576

Journal Ranking (2013): 10/56 (Instruments and instrumentation)

Sensors **2013**, *13*, 1210-1230; doi:10.3390/s130101210

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

A Network Access Control Framework for 6LoWPAN Networks

Luís M. L. Oliveira [1,2], Joel J. P. C. Rodrigues [1,*], Amaro F. de Sousa [3] and Jaime Lloret [4]

[1] Instituto de Telecomunicações, Universidade da Beira Interior, Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal; E-Mail: loliveira@ipt.pt

[2] Instituto Politécnico de Tomar, Quinta do Contador, Estrada da Serra, 2300-313 Tomar, Portugal

[3] Instituto de Telecomunicações, Universidade de Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal; E-Mail: asou@ua.pt

[4] Instituto de Investigación para la Gestión Integrada de Zonas Costeras, Universidad Politécnica de Valencia, C/Paranimf, n° 1, 46730 Grao de Gandia, Spain; E-Mail: jlloret@dcom.upv.es

* Author to whom correspondence should be addressed; E-Mail: joeljr@ieee.org; Tel.: +351-275-242-081; Fax: +351-275-319-899.

Received: 30 November 2012; in revised form: 8 January 2013 / Accepted: 16 January 2013 / Published: 18 January 2013

Abstract: Low power over wireless personal area networks (LoWPAN), in particular wireless sensor networks, represent an emerging technology with high potential to be employed in critical situations like security surveillance, battlefields, smart-grids, and in e-health applications. The support of security services in LoWPAN is considered a challenge. First, this type of networks is usually deployed in unattended environments, making them vulnerable to security attacks. Second, the constraints inherent to LoWPAN, such as scarce resources and limited battery capacity, impose a careful planning on how and where the security services should be deployed. Besides protecting the network from some well-known threats, it is important that security mechanisms be able to withstand attacks that have not been identified before. One way of reaching this goal is to control, at the network access level, which nodes can be attached to the network and to enforce their security compliance. This paper presents a network access security framework that can be used to control the nodes that have access to the network, based on administrative approval, and to enforce security compliance to the authorized nodes.

Keywords: wireless sensor networks; WSN; 6LoWPAN; network access control

---

List of Acronyms

6LN	6LoWPAN Node
6LR	6LoWPAN Router
6LBR	6LoWPAN Border Router
AES	Advanced encryption standard
CGA	Cryptographically generated addresses
DAD	Duplicate Address Detection
DAG	Directed Acyclic Graph
DAR	Duplicate Address Request
DODAG	Destination Oriented DAG
ECC	Elliptic curve cryptography
LSEND	Lightweight Secure Neighbor Discovery for Low-power and Lossy
MAC	Medium Access Control sub-layer protocol
MTU	Maximum Transmission Unit
NA	Neighbor advertisement
NDP	Neighbor discovery protocol
NS	Neighbor solicitation
PHY	Physical layer protocol
RA	Router advertisement
RPL	IPv6 Routing Protocol for Low-power and lossy networks
RS	Router solicitation
SEND	Secure neighbor discovery protocol

1. Introduction

Low power wireless personal area networks (LoWPAN) [1] comprise devices compliant with the IEEE 802.15.4 [2] standard and are widely used in embedded applications such as environmental monitoring, smart-grids, surveillance, industrial and home automation. These applications often require a large number of small devices to cover large areas and must operate unattended for years equipped with small batteries. Many of the IEEE 802.15.4 compliant devices are characterized by small size, power constraints, small computing and storage resources and by reduced radio ranges and throughput. Self-organization, fault-tolerance, and self-optimization are the main characteristics of LoWPAN networks [2].

Initially, the IP protocol stack was considered too complex to be supported by IEEE 802.15.4 devices. Meanwhile, the scientific community, together with the industry, started to rethink many of the misconceptions about the use of IP protocol stacks in resource-constrained devices [3]. Now, the IPv6 protocol is the most consensual solution to connect LoWPAN networks to the Internet facilitating the design and deployment of applications [4]. However, IPv6 was not designed to be used in such devices and, therefore, an adaptation layer was introduced between the link layer and the network layer. This adaptation layer, proposed by IETF 6LoWPAN Working Group [1], enables the transmission of IPv6 datagrams over IEEE 802.15.4 links by providing header compression (to reduce

overhead), fragmentation (to support the IPv6 minimum MTU requirement) and support for layer-two forwarding (to deliver IPv6 datagram over multiple radio hops). Although the standard IPv6 neighbor discovery protocol [5] might work on 6LoWPAN networks, it exhibits high overhead and includes no security support. The lightweight secure neighbor discovery optimizations for 6LoWPAN protocol (the LSEND protocol) [6,7] was proposed to circumvent these problems.

LoWPAN networks exhibit a large number of vulnerabilities, which make them even more prone to security attacks [8–12] than traditional IP networks. In fact, a single LoWPAN network can scale up to thousands of nodes without any fixed infrastructure and these nodes are often installed in harsh and unattended environments. Moreover, the addition of new nodes makes the network topology dynamic and complex to manage. Several security mechanisms have been proposed, some of them defined to address some particular well-known attacks [13]. Besides protecting the network from well-known attacks, it is also important that security mechanisms be able to withstand attacks that have not been identified before [14]. If a malicious node is prevented from becoming attached to the network, it cannot communicate with any network element and, therefore, it cannot launch any type of security attack. Therefore, one way of reaching this goal is to control, at the network access level, which nodes can be attached to the network and to enforce their security compliance. Besides the security advantages, such methodology also makes the network more manageable, while increasing its reliability and extending its lifetime [15].

This paper proposes a network access control security framework for 6LoWPAN networks, that controls the access of nodes to the network, based on administrative authorization, and enforces security compliance to the authorized nodes. The proposed framework makes use of LSEND protocol (for secure neighbor discovery and key pairwise generation), RPL (for datagram routing), and Seluge (for security compliant code dissemination). Unlike other access control mechanisms, this proposal includes an automatic remediation mechanism to enable nodes to become security compliant, if necessary, in order to have their access granted by the network.

The remainder of this paper is organized as follows: Section 2 addresses the security support requirements for LoWPAN networks and reviews the current solutions. Section 3 focuses on the technologies and protocols used to support the proposed network access control security framework. The Sections 4 and 5 present the proposed network access control design framework and discuss its application and the requirements for its implementation. Finally, Section 6 concludes the paper and identifies future research topics.

2. Security on LoWPAN Networks

Note that both LoWPAN networks and general resource unconstrained networks share almost the same security requirements [12,16]. However, due to the node resource constraints, the number of nodes and the absence of an organized communication infrastructure, supporting security services in LoWPAN networks is more challenging when compared with resource unconstrained networks. Confidentiality, authenticity, integrity, availability, data freshness, robustness, and survivability are the most relevant security requirements in LoWPAN networks [17,18]. Confidentiality ensures that only legitimate entities have access to the data transmitted and stored in the network nodes. Authenticity ensures that data is actually provided by its source nodes. Integrity guarantees that no data is changed

by any other entity, without being detected. Availability ensures that services provision is always available to their legitimate users. Data freshness prevents other entities from replaying old messages. Network robustness and survivability guarantees that the network still works properly even in the presence of intrusions, attacks, accidents and failures.

Following [19], LoWPAN security attacks can be broadly classified as external *versus* internal attacks, passive *versus* active attacks and mote-class *versus* laptop-class attacks. In external attacks, the attacker device can only use its own resources to perform the attack and has no access to the resources of the other LoWPAN nodes. External attacks can be prevented with cryptography. For example, a cryptographic mechanism used to support authentication and confidentiality prevents an external attacker from eavesdropping third-party messages or injecting false messages. In internal attacks, the attacker device is able to compromise legitimate nodes by using their resources to perform the attacks. Internal attacks involve either the injection of malicious code into target nodes (exploiting flaws in the application modules) and/or the access of key material code and data of the target nodes. This type of attack enables the attacker to appear as a legitimate node in the network gaining the trust of the other legitimate nodes. Internal attacks are much harder to detect when compared with external attacks [20].

Passive attacks are based on information gathering without modification. Eavesdropping and monitoring are examples of passive attacks. The active attacks involve modifications on the legitimate data stream or false data injection. In the mote-class attacks, nodes with similar resources as the legitimate nodes are used to perform the attack. In laptop-class attacks, an attacker uses devices with more resources, such as laptops, to perform the attack. Laptop-class attacks are especially effective, for example, to jam the wireless channel.

Security attacks against LoWPAN networks can also be classified, according to security requirements, in the following three main groups [16]: attacks against secrecy and authenticity, attacks against network availability, and stealthy attacks (*i.e.*, attacks against service integrity). Eavesdropping, packet replay, tampering and spoofing are examples of attacks against the secrecy and authenticity. Denial of service (DoS) is the main example of an attack against the network availability and can be targeted at the different layers of the networking stack [9,10]. In the stealthy attacks, the main goal is to make a legitimate node to accept false data values generated by the attacker and, in this way, the compromised legitimate node can also be used to amplify the dissemination of false data through all other legitimate nodes. Several mechanisms, most of them based on cryptography, can be used to address these different requirements in LoWPAN networks. Currently, the research on providing security solutions for LoWPAN networks has been focused mainly in three categories: (i) key management, (ii) authentication and secure routing and (iii) secure services. Some solutions were proposed to establish and manage cryptographic keys between nodes to enable authentication and encryption mechanisms [21] while others have been proposed to protect routing protocols [22]. Progress has been achieved on specialized secure services, such as secure localization, secure data aggregation and secure time synchronization [23,24].

There are also many proposals of security mechanisms addressing only particular attacks or used in particular layers as security tools. However, the major disadvantage with single layer security approaches is that security mechanisms are introduced on each layer, which in most of the cases tends to overall solutions with waste of resources power and exaggerated delays on message forwarding. Recently, researchers are pursuing security-integrated systems instead of concentrating on particular

attacks or layered based mechanisms. Some frameworks have been proposed to address simultaneously more than one security attack or to mitigate attacks using more than one security mechanism [25]. The most significant proposal is Security Protocols for Sensor Networks (SPINS) [26] which is composed by the Secure Network Encryption Protocol (SNEP) and micro-TESLA. SNEP provides data confidentiality and two-way data authentication with reduced overhead while micro-TESLA is a lightweight version of Time Efficient Streamed Loss-tolerant Authentication (TESLA) providing authenticated streaming broadcast.

SPINS does not cover, though, some relevant security issues, such as compromised nodes detection, DoS attacks or network and traffic analysis issues. Moreover, SPINS assumes a static network topology ignoring the ad hoc and mobile nature of LoWPAN networks. In [27], the authors propose a framework to provide secure cluster formation, security key management scheme and secure routing. It includes three components: a security mechanism to provide secrecy for communications in LoWPAN, an efficient session key distribution mechanism and a centralized key revocation scheme. The proposed framework does not depend on a specific key mechanism scheme and can be used to support many security applications, such as secure group communications.

Currently, the control of the access to the network is considered a critical security service in LoWPAN networks because it can be used to prevent malicious nodes from joining the network and launching internal attacks [15]. With such service, only eligible nodes can access the network, while queries from external attacker nodes are not answered or forwarded by regular nodes. In defining the recently proposed access control schemes, three main aspects can be distinguished: new node addition, authenticated querying and user authentication [14]. Query authentication schemes guarantee data origin authentication and data integrity while user authentication schemes are the basic solution used for the access control issue. New node addition schemes use mechanisms based on Elliptic Curve Cryptography (ECC) [28] to prevent malicious nodes from joining the network. Most of the secure network access systems provide node authentication and packet authentication, integrity verification and confidentiality. In [29], a self-certified elliptic curve Diffie-Hellman (ECDH) cryptosystem is used to establish a pairwise key between a new sensor node and a required Certificate Authority (CA) element, which can be implemented on regular nodes or on more powerful nodes such the border routers. The CA launches a two-way authentication procedure with the new node and establishes a pairwise key using the self-certified ECDH based protocol. The ECDH is used to guarantee nodes identification and to deliver a shared key to the new node. The shared key is the same for all nodes and is used to guarantee packet privacy and integrity. Note that the security of this scheme depends strongly on the secrecy of the shared key, *i.e.*, it fails if a single node is compromised.

To the best of our knowledge, though, none of the existing security frameworks includes the following important aspects which are dealt with by this proposal: to enable the node security compliance evaluation and enforcement (mitigating the security threats of internal attacks) and to use the same key pairwise for node authentication and routing protocol security.

3. Related Technologies

3.1. 6LoWPAN

Supporting IPv6 on sensor nodes simplifies simultaneously the task of connecting LoWPAN devices to the Internet and the application developing process. Currently, the IEEE 802.15.4 protocol is widely accepted as the PHY and MAC layer protocol for LoWPAN networks. Nevertheless, the network layer protocol must comply with the constraints imposed by the IEEE 802.15.4 protocol and the properties of the standard IPv6 protocol do not fully match with such constraints. Low bandwidth, low-power resources and the maximum packet size of 127 bytes are the most relevant characteristics of the IEEE 802.15.4 standard, which must be dealt with in proper protocol adaptation. Moreover, the support of standard IPv6 headers over LoWPAN would result in extremely small payloads for higher layer protocols. Besides LoWPAN requirements, it is also necessary to guarantee that 6LoWPAN is compliant with the IPv6 minimum MTU of 1280 bytes and, consequently, fragmentation and reassembly is required.

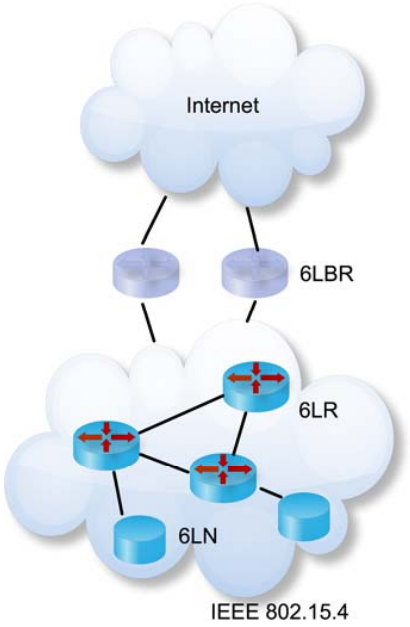
The IETF created the 6LoWPAN working group to define how to support IPv6 over IEEE 802.15.4 protocol. The 6LoWPAN working group was focused on the following issues [1]: (i) to define a neighbor discovery protocol fitted for low-power networks, (ii) to describe mechanisms allowing compression of 6LoWPAN headers in order to reduce the header overhead and (iii) to define the 6LoWPAN routing requirements and approaches. Two RFCs were released, the RFC 4919 [1] and the RFC 4944 [30]. The first document describes the assumptions, problem statement, and goals of 6LoWPAN. The second document proposes a 6LoWPAN adaptation layer (between IPv6 and IEEE 802.15.4) describing the frame format for transmission of IPv6 packets, the method for defining IPv6 link-local addresses and stateless auto configured addresses, the header compression and the frame delivery process in IEEE 802.15.4 mesh networks.

Rather than defining a single header (like IPv4), the 6LoWPAN uses stacked headers as the original IPv6 protocol does. The 6LoWPAN standard defines four header types: the dispatch header, the IPv6 compression header, the fragmentation header and the mesh header (by default, only the dispatch and compression headers are used). At the beginning of each header, a header type field identifies the header format.

An illustrated in Figure 1, a typical LoWPAN [31] consists of nodes (named 6LN or 6LoWPAN Nodes), routers (named 6LR or 6LoWPAN Routers) and border routers (named 6LBR or 6LoWPAN Border Routers).

Nodes (or 6LNs) usually do sensing and actuation operations but they do not forward datagrams form other nodes to their destination nodes. Routers (or 6LRs) are intermediate nodes that forward datagram from others nodes (or routers) to their destination nodes in the same LoWPAN and are present only in route-over topologies. Border routers (or 6LBRs) are the interconnection devices between the LoWPAN network and others networks as, for example, the Internet. Typically, nodes and routers have energy and computational resource constraints while border routers are mainly powered with much more computational resources.

Figure 1. 6LoWPAN network architecture.



3.2. Lightweight Secure Neighbor Discovery for 6LoWPAN (LSEND)

In traditional IPv6 networks, both nodes and routers use the neighbor discovery protocol [5] and the stateless address autoconfiguration [32], which, together, are referred to as the neighbor discovery protocols (NDP). These protocols enable the following functions: (i) learning prefixes and configuration parameters related to address configuration, (ii) locating neighborhood routers, (iii) maintaining reachability information on active neighbors and (iv) detecting duplicate addresses. Note that NDP was proposed for unconstrained node devices. Moreover, NDP for IPv6 networks uses multicast to exchange most of the protocol messages and was designed considering that routers and nodes are always active. Given the resource constraints of sensor nodes, the resource inefficiency associated to multicast based mechanisms, the low duty-cycle and multi-hop support, NDP on 6LoWPAN networks requires a different approach. Neighbor discovery optimization for low power and lossy networks [6] is a work in progress at IETF 6LoWPAN working group. It proposes optimizations to NDP, header compression context information dissemination, auto configuration addressing mechanisms and duplicate address detection for low power networks. The NDP signaling was changed by replacing the standard address resolution mechanism (based on multicast messages between hosts) with an address registration mechanism. Moreover, some multicast messages associated to node address configuration were replaced by unicast messages, providing host-initiated request for router advertisements (RA) and eliminating in this way the need for periodic router advertisement multicasting. In this way, NDP for 6LoWPAN is more suitable for multi-hop sensor networks and is independent of the selected routing approach.

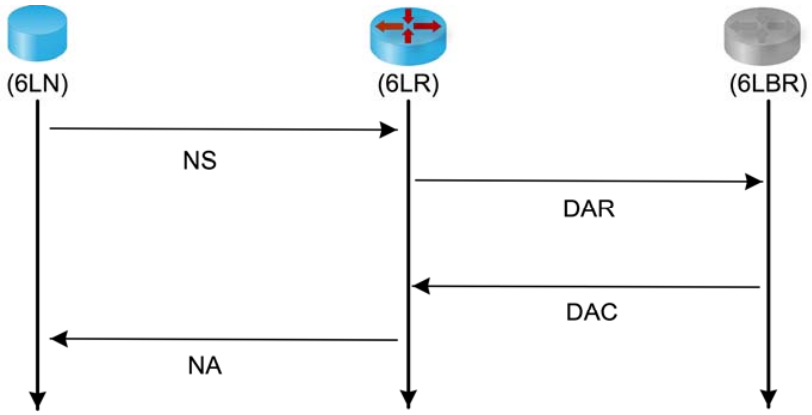
The 6LBR is responsible for interconnecting the LoWPAN to the Internet and for disseminating IPv6 prefixes and header compression context information across the LoWPAN. The 6LBR also maintains a network cache of all IPv6 addresses and EUI-64 identifiers. In this way, 6LBR is able to make layer-two address resolution and to perform duplicate address detection (DAD).

Besides the Router Solicitation (RS), Router Advertisement (RA), Neighbor Solicitation (NS) and Neighbor Advertisement (NA) message types which were already defined for IPv6 networks, NDP for 6LoWPAN [6] defines two new ICMPv6 message types to implement DAD on multi-hop networks: Duplicate Address Request (DAR) and Duplicate Address Confirmation (DAC) and some new options on the previous message types.

When a 6LN interface is initialized, a link-local address is formed based on the EUI-64 [33]. Next, the 6LN multicasts a RS message indicating its source link-layer address. All 6LRs with direct connectivity reply with a unicast RA message indicating the available IPv6 prefix(es). Once an address has been configured, the following messages exchange is shown in Figure 2. First, a unicast NS message is sent to the selected 6LR to register its configured address (if 6LN receives RA messages from different 6LRs, it should attempt to register its address in more than one 6LN to increase the network resilience). Then, the 6LR sends a unicast DAR message to the 6LBR to check if the IPv6 address is already in use and the 6LBR replies to the 6LR with a DAC message indicating the status of the registration. The status indicates either a successful registration or a failure due to a duplicated address (or any other reason like, for example, the routers registration cache exhaustion). Finally, the 6LR sends a unicast NA message to the 6LN indicating the same status as received from the 6LBR in the DAC message.

The information contained on NA messages have an associated lifetime and the address registration process is repeated before the lifetime expires. In this way, NS messages are also used to perform unreachability detection and are mainly used by nodes to verify the default router reachability.

Figure 2. 6LoWPAN neighbor discovery address registration.



NDP is not secure when physical security on the link is not guaranteed. If a malicious node knows, by spoofing, the link-layer and the IPv6 addresses previously registered by a legitimate node, it might register the same IPv6 address either with its own link-layer address or with a fictitious address. This vulnerability can be exploited for different types of security attacks: redirection, denial-of-service and flooding denial-of-service. In redirection attacks, the malicious node receives the packets and reroutes them (either changed or unchanged) to their legitimate node. In denial-of-service attacks, the malicious node prevents communications between the legitimate node and either all other nodes or some particular destination node. In flooding denial-of-service attacks, the malicious node redirects the packets from other nodes to a victim node to create on it a flood of bogus traffic.

Meanwhile, the secure neighbor discovery protocol (SEND) [34] was proposed for traditional IPv6 networks to protect NDP against these attacks. The SEND protocol uses: (i) an authorization delegation discovery process to prove the routers' identity, (ii) an address ownership proof mechanism based on cryptographically generated addresses (CGA) and (iii) digital signatures for all NDP messages. SEND specification uses a cryptographically generated address method to bind a RSA public key to an IPv6 address and to digitally sign all NDP messages.

However, RSA is not suitable for low power and resource constrained nodes, because it is computationally intensive and leads to long message sizes [34]. To overcome this problem, the lightweight secure neighbor discovery for low-power and lossy networks work in progress specification (LSEND) [7] relies on elliptic curve cryptography (ECC) to generate the CGA and on elliptic curve digitally signature algorithm (ECDSA) to sign the NDP messages. In fact, elliptic curve cryptography can provide the same level and type of security guarantees as RSA using much shorter keys [34,35]. The computational overhead of both ECC and RSA raises with $O(N^3)$, where N is the bit length, while ECC and ECDSA lead to much smaller message sizes and lower computational load when compared with RSA [36]. All nodes generate a public and private key pair for each network interface in order to generate their own CGA addresses and to create the digital signatures [34], necessary to sign NDP messages. The digital signature is a hash code based on nodes private key, source and destination IPv6 addresses, header type protocol (8 bits), header checksum value (16 bits), the NDP message header (and its options) and a 128 bits constant, randomly generated and designated by message type tag. The ECC based algorithm used to generate CGA takes three parameters: the EUI-64 identifier, the public key of the interface and a three bit security parameter used to hamper brute force attacks.

Note that both SEND and LSEND protocols require no public key infrastructure. Therefore, any node, including potential attacker nodes, may generate and register valid CGAs, but an attacker node cannot use a CGA previously registered by legitimate nodes, preventing in this way the previously described attacks.

When a 6LN receives a RA message from a 6LR, it configures its own CGA address and launches the address registration process (as illustrated in Figure 2) by sending a NS message with both the configured address and the CGA options (*i.e.*, the IPv6 source address of the NS message is set to its CGA address, the message carries the 6LN public key and it is signed with the 6LN private key). The 6LR receives the NS message and, based on its CGA options, runs two verification steps: (1) it verifies the source address using the claimed IPv6 source address and (2) it runs a cryptographic check of the signature included in the NS message. If both steps are successful, the 6LR proceeds with the address registration process as previously described. In this case, the 6LBR caches not only the 6LN IPv6 address but also its public key.

3.3. IPv6 Routing Protocol for Low Power and Lossy Networks

The IETF routing over low-power and lossy networks (RoLL) working group was chartered to design a routing protocol to be used in 6LoWPAN networks that addresses the requirements described in RFCs 5548 [37], 5673 [38], 5826 [39], and 5867 [40]. In 2010, RoLL introduced the IPv6 Routing Protocol for Low-power and lossy networks (RPL) [31].

Currently multipoint-to-point traffic pattern is dominant, because in most LoWPAN applications a few nodes are used to retrieve data from sensor nodes and the sensors rarely communicate between each other. To support this traffic pattern, RPL builds a destination oriented directed acyclic graph (DODAG) to route the data traffic. RPL defines a new ICMPv6 message with three possible types: DODAG information object (DIO) used to transport information that allows a node to discover an RPL instance, learn its configuration parameters and select DODAG parents; the DODAG information solicitation (DIS) to request a DODAG information object from a RPL node and the destination advertisement object (DAO) used to propagate information upwards along the DODAG.

During the DODAG construction and maintenance, nodes send DIO messages to their neighbors, carrying the objective function used to compute the rank value of each node. The rank defines the node relative position within a DODAG with respect to the root. The objective function specifies the metrics and constraints used to compute the routing path, the node rank position and the parent node set. RPL includes a flexible framework that incorporates dynamic routing metrics, such as expected number of transmissions (ETX). RPL also describes the constraints on how nodes select potential parents from their neighbors. For example, the node security support can be used as a constraint and, in this case, the nodes that do not support some security mechanism cannot be members of the DODAG. Nodes listen for DIO messages and use their information to join a new DODAG or to maintain an existing DODAG. Based on the DIO information, nodes choose parents that minimize path cost to the DODAG root. In the steady state, each node has a set of parents where the one with least rank value is selected as the preferred parent and the others are used as backup nodes.

DIO messages only enable upward routes computation and nodes have no knowledge about their children. In order to support routing to other destinations within the DODAG, RPL uses DAO messages to inform parents of their presence and reachability to descendants. The root node gathers the DAO messages from all other nodes and uses them to build downward routes to all destinations. RPL also defines local and global repair methods for re-computing routes when some inconsistency is detected.

Security is an important design consideration for LoWPAN networks because several attacks against the routing protocols were already identified. In order to guarantee the integrity of routing messages, RPL defines an optional cryptographic operation mode, in which advanced encryption mechanisms are used for message authentication. AES for message authentication and RSA signatures for checking the integrity of routing messages are already considered. Using ECC signatures to substitute RSA is possible and desirable because it provides the same security guarantees with much smaller keys, although the ECC inclusion is a work in progress [41].

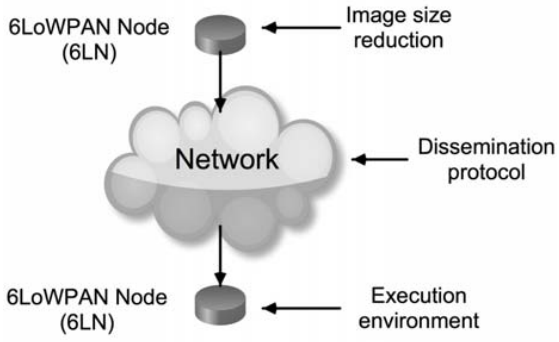
3.4. Node Remote Reprogramming Mechanisms

In the general case, wireless node reprogramming, also named over-the-air reprogramming, is a useful tool for remote uploading new codes or for changing the functionalities of existing codes [42]. Remote reprogramming is especially useful on large networks installed on harsh or difficult access environments. In our case, the proposed network access control framework relies on remote reprogramming as a means to enforce node security, *i.e.*, to guarantee that legitimate nodes are using security free codes before being connected to the network.

Remote software installation is more challenging on LoWPAN networks mainly because of the resource constraints of the nodes, the fact that a single network might have thousands of nodes and the fact of communications being over multihop links. Moreover, a program image is relatively large to be transmitted over low-power wireless links where link losses and collisions often occur. Consequently, efficient mechanisms must be used to upload the software, minimizing the nodes energy consumption. Significant research has been done in order to address the resource constraints nature of LoWPAN networks. The first proposed reprogramming mechanisms have only assumed single-hop networks. In this case, only nodes with direct links with the border router (usually selected to store the new software because it has more resources than the other nodes) can be reprogramed. Recently, more mechanisms were proposed to address multi-hop reprogramming [43]. In fact, multi-hop reprogramming can be more efficient, both in terms of time and energy, when compared with single-hop reprogramming. First, shorter links can be used, which reduces the retransmissions due to collisions. Second, the multi-hop protocols divide the entire code image into pages and when a node completes downloading a single page, it can send it to other nodes in the network while downloading the next page (this process is referred to as spatial multiplexing).

Remote reprogramming research is organized under the following three categories [44] (Figure 3): the sensor node execution environment, the protocols for update dissemination, and the image size reduction. Power efficiency, performance, security and reliability are common issues addressed by the solutions proposed for each category.

Figure 3. Node remote reprogramming mechanisms.



3.4.1. Sensor Node Execution Environment

The execution environment is the responsible to run programs on top of the available hardware. The first systems had dedicated execution environments to make the best use of the available hardware heterogeneity. The presence of a memory management unit (MMU) can have a significant impact on the software update process. The MMU is the hardware component that manages virtual memory systems and, in most situations, it is included on the CPU chip. When the MMU is available, much of the code position is independent, because virtual memory addressing is used, adding safety boundaries to programs. Moreover, individual modules are executed as separate processes. Since processes run in a virtual address space, they can be easily upgraded at runtime without using references to absolute memory addresses. However, many of microcontrollers used on sensor nodes are not equipped with MMU. Without the MMU, a node operates in a single address space and, in such case, the node is

more vulnerable to incorrect or malicious memory references. Execution environments without MMU can be classified in monolithic environments, modular environments and virtual machines.

In monolithic environments, the entire executable program is statically optimized during the compilation. In this case, a single system image containing both all applications and the system kernel is produced making efficient use of CPU and memory. The TinyOS operating system is an example of a monolithic execution environment. In such execution environments, a new complete image must be uploaded into the node, resulting in large update patches.

In modular execution environments, individual modules can be independently loaded on demand. In this type of environment, the system is divided into two parts: the kernel, usually static, and the loadable component images, usually dynamic. The kernel provides services to the modules, such as memory management, I/O, and communications. The modules access the services provided by the kernel at predefined addresses or through a system jump table. The updates are smaller than in monolithic counterpart because the modules are separated from the kernel part. Moreover, the kernel part requires less frequent updates than in monolithic environment because it only provides the interfaces to the module applications. Contiki and Bertha operating systems use modular execution environments.

Virtual machine (VM) environments are used to virtualize underlying hardware, providing high-level operations to applications through an instruction interpreter. VM typically executes a program in a sandbox where direct access to hardware is not allowed. In sensor networks, a VM environment also allows the implementation, by software, of some features that the hardware might not provide, such as the memory management unit. Unfortunately, VM introduces overhead in both execution time and memory resources. First, the runtime interpretation causes programs to run slower by at least an order of magnitude when compared with native execution. Second, the VM itself requires a certain fixed amount of memory to operate, and interpreted programs are generally more memory demanding. Several VM environments for TinyOS and Contiki operating systems were proposed, such as Maté and Agilla [45].

3.4.2. Protocols for Update Dissemination

The proposed protocols for software update dissemination are mainly based on data dissemination protocols, such as RMTS [46] and direct diffusion [47]. Software dissemination protocols operate in three steps: (i) advertisement of available software, (ii) source selection and (iii) reliable download to the destination.

Deluge is generally accepted as the state of the art for code dissemination in wireless sensor networks, and has been included in the latest TinyOS distributions [48]. Deluge is a reliable data dissemination protocol for disseminating data objects from one or more source nodes to many other nodes over a multihop wireless network. Data objects are represented as a set of fixed size pages to enable incremental updates and to allow spatial multiplexing.

Code dissemination protocols can be used to compromise LoWPAN networks. For example, an attacker may attempt to modify or replace the authentic code image introducing malicious code into the network nodes. Code dissemination protocols can also be used to perform denial-of-service attacks, where the malicious node injects bogus code and force network nodes to verify and forward them leading to exhausting their battery power. Several recent protocols, based on Deluge, were proposed to

address secure code dissemination. For example, Sluice integrates cryptographic digital signatures and hash functions to provide authentication for code dissemination. Like Deluge, Sluice also splits code images into fixed size pages. The hash code of each page is included in the previous page. The hash code of the first page is signed and included in the packet signature. This approach solves the authenticity attacks, but it does not address denial-of-service attacks. Since a node can only perform authentication when a complete page is received, many packets must be processed before the authenticity can be verified. Seluge [49,50] inherits the efficiency and robustness properties from Deluge providing authentication mechanisms for code dissemination and protection for DoS attacks against signature packets, code dissemination packets and maintenance packets. Seluge uses public key mechanisms based on elliptic curve cryptography.

3.4.3. Size Reduction Mechanisms

While the dissemination protocol aims to minimize the overhead related to delivering updates, reducing the size of transmitted software is used to decrease the size of the updates. Three main techniques are used to achieve this objective: compression, differential patching and high-level instructions. Several reducing size mechanisms were proposed, such as: Reijers [51], Rsync [52] and Remote Incremental Linking [53]. Reijers reduces the image size generation differential script. The script is downloaded as a series of packets where each packet contains the new instruction code and its address. The script is applied against the currently running program stored in the external memory, usually an EEPROM, gradually building a new image. When this process is completed, the boot loader loads the new image into the running memory. The scripting language is cpu-specific, which is the Reijers algorithm major flaw. Rsync also provides incremental software updates but its algorithm is CPU independent. The Rsync script can either specify unmodified block to be copied for a new address or can contain modified code. The updater node asks the node being updated for the checksum of its current blocks and, according to the response, only sends changed code. Remote Incremental Linking was designed for mica2 mote hardware platform and supports dynamic and static updates. An image program is composed by several functions and at the end of the functions some space is reserved to allow future expansion. New versions of the functions can be written at the same start address memory, as long as they fit, without the need to update references to the functions elsewhere in the program. Otherwise, either adjacent functions or the new functions are moved, which minimizes page writes.

4. Network Access Control Security Framework

This paper proposes a network access control security framework, for 6LoWPAN networks, that controls the access of nodes to the network, based on administrative authorization, and enforces security compliance to the authorized nodes. The proposed framework can operate in two different modes: listening and active. In the listening mode, no security assessment and enforcement is performed. This mode is useful for network visibility, because it can be used to gather information about the connected nodes. The active mode is more secure than the listening mode because only security compliant nodes can be connected to the network.

4.1. Nodes Requirements

In order to provide a secure and manageable solution, a few assumptions related to nodes and border routers must be defined, according to the selected operating mode. To operate in listening mode, all nodes must support 6LoWPAN and LSEND protocol. To operate in active mode, in addition to the previous requirements, the nodes must also support a secure reprogramming mechanism and the RPL routing protocol with message authentication. The secure reprogramming mechanism can be implemented using Seluge combined with Maté execution environment and Rsync size reduction mechanism. The border router must support both operating modes simultaneously and, therefore, their requirements are the same as the nodes when supporting the active mode. We assume that border routers are hard to be compromised.

4.2. Node Identification, Compliance and Data Security

As sensor networks are mostly deployed in human-unattended environments, usually for critical sensing measurements tasks, the authentication of the data source as well as the data itself is of critical concern. In fact, authentication guarantee can provide both sensor and router identification ability, in order to protect the integrity and freshness of critical data, and forbid and/or identify several security attacks. Traditionally, there are two schemes to provide authentication: digital signatures based on public-keys and message authentication code based on symmetric-keys. In the current proposal, the same public and private key pairwise, generated by the LSEND protocol to build the cryptographically generated addresses (CGA) addresses, is used to authenticate the network nodes. As aforementioned, the CGA are IPv6 addresses where the least 64 address bits are generated by computing a cryptographic one-way hash function from a public key and other auxiliary parameters. The address owner uses the corresponding private key to declare the address ownership and to sign the messages sent without a certification authority or other security infrastructure. The binding between the address and the public key can be confirmed by re-computing the hash value and by comparing the hash with the interface identifier. As a consequence, messages sent from a CGA IPv6 address can be protected by attaching the public key and auxiliary parameters and by signing the message with the corresponding private key. Every node can generate valid private and public key pairwise and the correspondent CGA address and, once registered in the border router, no other node can use the same address. In our proposal, after the LSEND registration procedure, the administrator approves manually each new node based on its CGA address. Once authenticated, each node is able to verify if the sender node is the right owner of the CGA generated source address. One of the key advantages of this approach is that, in addition to ensuring the identity of the sender, the same public and private pairwise can be also used to guarantee the authenticity, confidentiality and data freshness to all protocols or mechanisms in use.

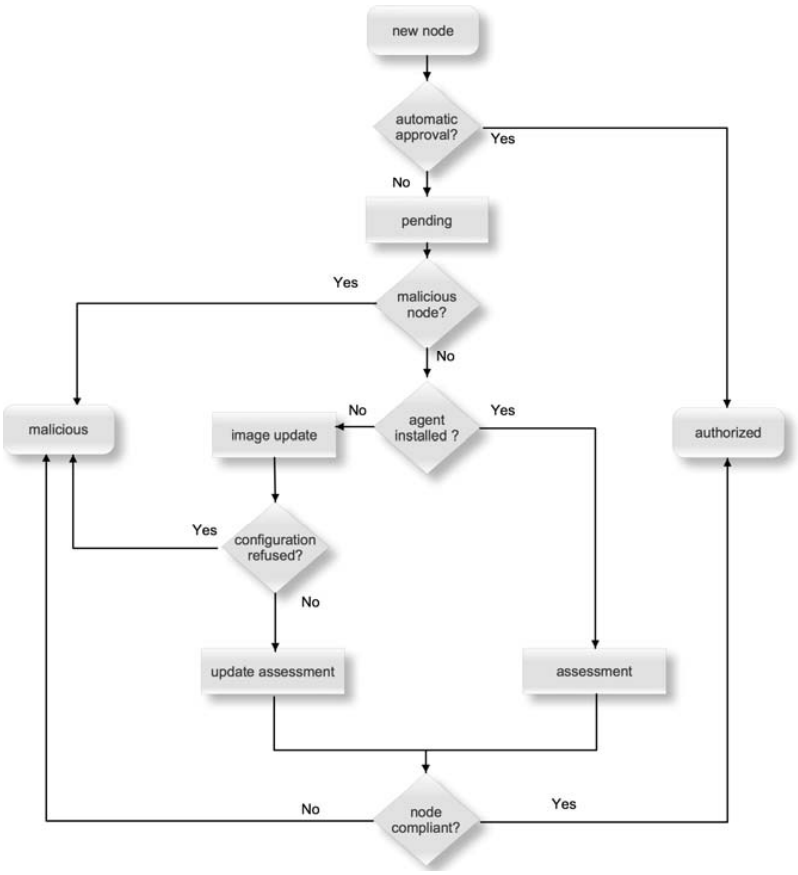
Traditionally, two different approaches are used to perform node security compliance. In the first approach, a piece of software named agent is used to perform the nodes assessment. The agent returns to the security enforcement point (in our case, the border router) the result of the assessment. In the second approach, the assessment is performed based on the traffic generated by the node under assessment. The security enforcement point sends requests to the node under assessment and the evaluation is made based on the responses. Usually, the first approach returns more accurate results

than the second approach. Moreover, the second approach is more verbose than the first one, because multiple messages are exchanged. The current proposal adopts the agent based assessment approach. The agent can be previously installed on the nodes and the assessment is based on cryptographic hash of the installed software. If the agent is not installed, we assume that the node is not compliant and a new image, which includes the assessment agent, will be uploaded by the reprogramming mechanism on the fly. The hardware configuration can also be evaluated during the assessment, although this feature is not considered in our solution. Note that any solution to perform nodes security assessment is very dependent on the operating system and the hardware used in the network devices. So, different agent versions might be required if different operating systems or hardware platforms are to be used. The border router can use operating system fingerprinting in order to determine which agent version has to be used.

4.3. Access Control Algorithm Description

The access control algorithm of our proposal assumes that each node is always in one of seven states: new node, pending, assessment, image update, update assessment, authorized and malicious. Figure 4 illustrates the decision process from the initial 'new node' state until one of the final states is reached, which is either 'authorized' or 'malicious'.

Figure 4. Access control decision process.



The 6LoWPAN nodes can use link layer detection mechanisms and/or router advertisement messages to detect new networks. If a new network is detected, the 6LoPWAN nodes must use

LSEND protocol to perform the address registration process, as described in section 3.2. The address registration process is used to detect new nodes presence. In the listening mode, the automatic approval is in use and, therefore, the new node moves to the authorized state and its address is added to the border router as a registered entry [6]. This is the less secure operation mode and is used by the administrator to determine which nodes are connected.

In the active mode, the automatic approval is not set and, therefore, the new node moves to the pending state and its address is stored in the border router neighbor cache as a tentative entry [6]. Before the authorized state is reached, only messages related to the assessment, image deployment and authorization are allowed. In the pending state, the administrator can classify the new node as malicious if: (i) additional nodes are not expected, (ii) the administrator does not rely on the received information or (iii) the new node was considered malicious in the past. If the administrator does not consider the new node as malicious, the next step is to verify if the assessment agent is installed. If yes, the node moves to the assessment state and its security compliance is evaluated. Then, the node transits to the authorized state if the evaluation is successful or to the malicious state, otherwise. If the assessment agent is not installed, the node moves to the image update state, where the border router tries to install on the node a new image using Deluge (the new image includes the operating system, the assessment agent and all required modules). If the configuration of the new image is refused by the node, if moves to the malicious state. Otherwise, the node moves to the update assessment state and its security compliance is evaluated. Then, the node transits to the authorized state if the evaluation is successful or to the malicious state, otherwise.

When a node is moved to the malicious state, its address is marked as garbage-collectible in the border router neighbor cache, the address is propagated by the RPL messages to all authorized nodes as a restriction, and the address is pruned from all nodes RPL candidate neighbor list and cannot belong to the DODAG instance in use. So, in the malicious state, no messages from these nodes are processed or forwarded by any other LoWPAN node.

5. Discussion

The security research applied to LoWPAN networks is considered by the industry and by the research community a very hot topic. Several security mechanisms were proposed, some of them to address some particular scenarios, application and protocols. Besides protecting the network from some well-known threats, it is important that security mechanisms can withstand to attacks that have not been identified before. Therefore, the challenge is on how can the attackers explore vulnerabilities that have not been yet identified. The control of which nodes can take part of the network and their security compliance can help to reduce several vulnerabilities, making the network more manageable, while increasing its reliability and extending its lifetime. The proposed security framework proposal can be used to achieve these objectives, in particular to make the network more resilient to internal attacks. First, network assessment control restricts the network access only to authorized nodes. Second, node security compliance is enforced before node can access to the network. Finally, beyond to security compliance assessment, the software image can also be checked in order to guarantee that node is able to realize the expected functions. Therefore, only authorized nodes that fulfill security and

functions requirements are accepted. This is particularly relevant if multi-hop networks are used. Node authorization depends on:

- Administrator authorization: a manual authorization was considered because it is very hard to define rules that can be applied to all network security requirements. For example, in a monitoring network installed in a nuclear power plant, if a new node tries to access the network, it will be most probably a malicious node since the network infrastructure remains unchanged for long time periods. Therefore, the administrator can approve the new nodes based on: hardware type, layer-two address and location. This approval method also protects the framework against DoS attacks, because only approved nodes will be evaluated. All nodes are identified by a cryptographic generated address, according to LSEND protocol.
- Security check compliance: several conditions can be considered as inputs to the agent used to assess the security compliance such as, for example, the installed software image and the security protocols in use. Note that multiple agents might be required if different operating systems or hardware platforms are used in the same network. The decision on which agent should be used on each device node is a challenge.
- Hardware and software image compliance: providing plug-and-play mechanism is not enough to guarantee that node is able to realize the desired functions. For example, a sensor node is unable to monitor the temperature if the module used to retrieve the temperature is missing. The same occurs with the hardware. Software image compliance also helps to protect against malicious code injection.

The proposed framework also improves the network manageability, since remote software installation is supported. The implementation success of the proposed security framework depends on the ability to integrate the above described mechanisms and protocols in order to take advantage of synergies between them. In fact, the following modifications are required in order to maximize the integration benefits.

LSEND requires the addition of a secure mechanism to inform nodes that a neighbor must be removed from its neighbor cache in order to avoid communications between authorized nodes and malicious nodes. It is also necessary to define a data structure to share the ECC key pairwise generated by the LSEND protocol with the RPL routing protocol and remote image installation mechanism. Note that the ECC key pairwise can also be used to protect the application layer data exchange, in order to guarantee data authentication and authenticity. In fact, the reutilization of the same ECC private and public key pairwise between several protocols and mechanisms simplifies the operations related to the key management [25]. The key pairwise reutilization also extends the network lifetime because fewer messages are used when compared with other solutions that use one key for each protocol or mechanism. In fact, the transmission energy consumption rate, in wireless sensor networks can be over three orders of magnitude greater than the energy consumption rates for computing [54].

In the RPL protocol, the ECC support must be defined in order to protect the routing messages exchange. Also, an efficient mechanism must be defined in order to propagate efficiently addresses as a constraint in order to avoid the malicious nodes to participate in the routing tree.

Concerning node remote reprogramming mechanisms, several mechanisms must be combined to enable node remote reprogramming. These mechanisms are very dependent on the operating system and the hardware in use. As a consequence, it is foreseen that early implementations of the proposed

framework in real deployment scenarios will support node remote reprogramming mechanism only for a few hardware platforms and operating systems. Currently, a laboratory testbed is being implemented where the above-mentioned protocol modifications are being conducted and integrated aiming to validate the proposed security framework.

6. Conclusions and Future Work

Improving the security is critical for the success of LoWPAN networks, because these types of networks are particularly vulnerable, they are used in critical services and the data collected is often sensitive. Several security solutions were proposed, most of them designed to address known attacks and implemented on particular layers. Several security attacks can be avoided if a network access control mechanism is used to restrict the network access only to authorized nodes which are compliant with the defined security requirements. This paper, we have proposed a network access control framework that can be used to accomplish these objectives. The proposed solution enables the node identification based on cryptographically generated addresses, node security compliance evaluation and node remediation with secure remote software installation. This solution is mainly based on the following open protocols: LSEND, used to secure neighbor discovery and node secure identification, RPL with ECC support, to protect the routing messages and Seluge, to enable secure remote software installation. In the current proposal, synergies between the protocols were taken in consideration. Work is still required to improve and integrate these protocols in order to create a solution. Moreover, the current proposal only enables one border router. As a consequence, when the border router becomes unreachable, all nodes must be approved and evaluated. Further research is required to circumvent this limitation since a secure mechanism is required to synchronize the ECC keys and nodes authorization state between multiple border routers. Besides the current lab implementation to validate our current proposal, this research is another direction of our future work.

Acknowledgments

This work has been partially supported by the *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal, by National Funding from the FCT — *Fundação para a Ciência e Tecnologia* through the Pest-OE/EEI/LA0008/2011, by the AAL4ALL project (Ambient Assisted Living for All), co-funded by COMPETE under FEDER via QREN Programme, and by the LOPIX QREN Project.

References

1. Kushalnagar, N.; Montenegro, G.; Schumacher, C. *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals, RFC 4919*; 2007. Available online: http://www.ietf.org/rfc/rfc4919.txt (accessed on 27 August 2012).
2. Oliveira, L.M.L.; Sousa, A.F.; Rodrigues, J.R. Routing and mobility approaches in IPv6 over LoWPAN mesh networks. *Int. J. Commun. Syst.* **2011**, *24*, 1445–1466.
3. Gershenfeld, N.; Krikorian, R.; Cohen, D. The internet of things. *Sci. Am.* **2004**, *4*, 76–81.
4. Hui, J.; Culler, D. Extending IP to low-power, wireless personal area networks. *IEEE Internet Comput.* **2008**, *4*, 37–45.

5. Narten, T.; Nordmark, E.; Simpson, W.; Soliman, H. *Neighbor Discovery for IP version 6 (IPv6), RFC 4861*; 2007. Available online: http://www.ietf.org/rfc/rfc4861.txt (accessed on 27 August 2012).
6. Shelby, Z.; Chakrabarti, S.; Nordmark E. *Neighbor Discovery Optimization for Low Power and Lossy Networks*; Draft−ietf−6lowpan−nd−21; 2012; unpublished work.
7. Sarikaya, B.; Xia F.; Zaverucha, G. *Lightweight Secure Neighbor Discovery for Low-Power and Lossy Networks*; Draft-sarikaya-6lowpan-cgand-03; 2012; unpublished work.
8. Yong, W.; Attebury, G.; Ramamurthy B. A survey of security issues in wireless sensor networks. *IEEE Commun. Surv. Tut.* **2006**, *8*, 2–23.
9. Oliveira, L.; Rodrigues, J.; Sousa, A.; Lloret, J. Denial of service mitigation approach for IPv6-enabled smart object networks. *Concurr. Comp.-Pract. E.* **2013**, *25*, 129–142.
10. Du, X.; Chen, H. Security in wireless sensor networks. *IEEE Wirel. Commun.* **2008**, *15*, 60–66.
11. Pelechrinis, K.; Iliofotou, M.; Krishnamurthy, V. Denial of service attacks in wireless networks: The Case of Jammers. *IEEE Commun. Surv. Tut.* **2011**, *13*, 245–257.
12. Lopez, J.; Roman, E.; Alcaraz, C. Analysis of security threats, requirements, technologies and standards in wireless sensor network. *Lect. Notes Comput. Sci.* **2009**, *5705*, 289–338.
13. Kavitha, T.; Sridharan, D. Security vulnerabilities in wireless sensor networks: A survey. *J. Inf. Assur. Secur.* **2010**, *5*, 31–44.
14. Faye, Y.; Niang, I.; Noel, T. A survey of access control schemes in wireless sensor networks. *Proc. World Acad. Sci. Eng. Tech.* **2011**, *59*, 814–823.
15. Sun, K.; Liu, A.; Xu, R.; Ning, P.; Maughan, D. Securing Network Access in Wireless Sensor Networks. In *WiSec '09 Proceedings of the Second ACM Conference on Wireless Network Security*; ACM: New York, NY, USA, 2009.
16. Shi, E.; Perrig, A. Designing secure sensor networks. *IEEE Wirel. Commun* **2004**, *11*, 38–43.
17. Ramen, R.; Lopez, J.; Gritzalis, S. Situation awareness mechanisms for wireless sensor networks. *IEEE Comm. Mag.* **2008**, *46*, 102–107.
18. Sakerindr, P.; Ansari, N. Security Services in Group Communications over wireless infrastructure, mobile Ad Hoc and sensor networks. *IEEE Wirel. Commun.* **2007**, *14*, 8–20.
19. Singh, S.K.; Singh, M.P.; Singhtise, D.K. A survey on network security and attack defense mechanism for wireless sensor networks. *Int. J. Comput. Trends Tech.* **2011**, *5–6*, 1–9.
20. Khan, M.K.; Alghathbar, K. Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'. *Sensors* **2010**, *10*, 2450–2459.
21. Xiao, Y.; Rayi, V.K.; Sun, B.; Du, X.;, Hu, F.; Galloway, M. A survey of key management schemes in wireless sensor networks. *Comput. Commun.* **2007**, *30*, 2314–2341.
22. Wood, A.; Fang L.; Stankovic, J., He, T. SIGF: a family of configurable, secure routing protocols for wireless sensor networks. In *SASN '06 Proceedings of the Fourth ACM Workshop on Security of Ad Hoc and Sensor Networks*; ACM: New York, NY, USA, 2006; pp. 35–48.
23. Alzaid, H.; Foo E.; Gonzalez N.J. Secure Data Aggregation in Wireless Sensor Network: A Survey. In *AISC '08 Proceedings of the Sixth Australasian Conference on Information Security*; Brankovic, L., Miller, M., Eds.; Australian Computer Society, Inc.: Darlinghurst, Australia, 2008; Volume 81, pp. 93–105.
24. Sun, K.; Ning, P.; Wang, C. Fault-tolerant cluster-wise clock synchronization for wireless sensor networks. *IEEE Trans. Depend. Secure.* **2005**, *2*, 177–189.

25. Yong, W.; Ramamurthy, B.; Xue, Y.; Zou, X. A security Framework for Wireless Sensor Networks Utilizing a Unique Session Key. In *Proceedings of Broadband Communications, Networks and Systems*, London, UK, 8–11 September 2008; pp. 487–494.
26. Perrig, A.; Szewczyk, R.; Tygar, J.D.; Wen, V.; Culler, D. SPINS: Security protocols for sensor networks. *Wirel. Netw.* **2001**, *8*, 521–534.
27. Zia, T.A.; Zomaya, A.Y. A lightweight security framework for wireless sensor networks. *J. Wirel. Mobile Netw., Ubiquitous Comput. Dependable Appl. (JoWUA)* **2011**, *2*, 53–73.
28. Gura, N.; Patel, A.; Wander, A.; Eberle, H.; Chang-Shantz, S. Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. In *Proceedings of CHES '2004 Workshop on Cryptographic Hardware and Embedded Systems-Lecture Notes in Computer Science*; Springer-Verlag: Cambridge, MA, USA, 2004.
29. Ortal, A.; Qi, H. Load balanced key establishment methodologies in wireless sensor networks. *Int. J. Secur. Netw.* **2006**, *1*, 158–166.
30. Montenegro, G.; Kushalnagar, N.; Hui, J.; Culler, D. *Transmission of IPv6 Packets over IEEE 802.15.4 Networks, RFC 4944*; 2007. Available online: http://www.ietf.org/rfc/rfc4944.txt (accessed on 25 August 2012).
31. Winter, T.; Thubert, P.; Brandt, A.; Hui, J.; Kelsey, R.; Levis, P.; Pister, K.; Struik, R.; Vasseur, JP.; Alexander, R. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, RFC 6550*; 2012. Available online: http://www.ietf.org/rfc/rfc6550.txt (accessed on 25 August 2012).
32. Thomson, S.; Narten, T.; Jinmei, T. *IPv6 Stateless Address Autoconfiguration, RFC 4862*; 2007. Available online: http://www.ietf.org/rfc/rfc4862.txt (accessed on 25 August 2012).
33. Hinden, R.; Deering, S. *IP Version 6 Addressing Architecture, RFC 4291*; 2006. Available online: http://www.ietf.org/rfc/rfc4291.txt (accessed on 25 August 2012).
34. Arkko, J.; Kempf, J.; Sommerfeld, B.; Zill, B.; Nikander, P. *SEcure Neighbor Discovery (SEND), RFC 3971*; 2005. Available online: http://www.ietf.org/rfc/rfc3971.txt (accessed on 25 August 2012).
35. Driessen, B.; Poschmann, A.; Paar, C. Comparison of Innovative Signature Algorithms for WSNs. In *WiSec '08 Proceedings of the 1st ACM Conference on Wireless Network Security*; ACM: New York, NY, USA; pp. 30–35.
36. Rongbo, Z.; Ya, M. Research on Key Management Scheme for WSN Based on ECC. In *Information Engineering and Applications*; Zhu, R., Ma, Y., Eds.; Springer: London, UK, 2012; Volume 153, pp. 219–216.
37. Dohler, M.; Watteyne, T.; Winter, T.; Barthel, D. *Routing Requirements for Urban Low-Power and Lossy Networks, RFC 5548*; 2009. Available online: http://www.ietf.org/rfc/rfc5548.txt (accessed on 25 August 2012).
38. Pister, K.; Thubert, P.; Dwars, S.; Phinney, T. *Industrial Routing Requirements in Low-Power and Lossy Networks, RFC 5673*; 2009. Available online: http://www.ietf.org/rfc/rfc5673.txt (accessed on 25 August 2012).
39. Brandt, A.; Buron, J.; Porcu, G. *Home Automation Routing Requirements in Low-Power and Lossy Networks, RFC 5826*; 2010. Available online: http://www.ietf.org/rfc/rfc5826.txt (accessed on 25 August 2012).
40. Martocci, J.; Mi, P.D.; Riou, N.; Vermeylen, W. *Building Automation Routing Requirements in Low Power and Lossy Networks, RFC 5867*; 2010. Available online: http://www.ietf.org/rfc/rfc5867.txt (accessed on 25 August 2012).

41. Ko, K.; Dawson-Haggerty, S.; Hui, J.; Culler, D.; Levis, P.; Terzis, A. Connecting low-power and lossy networks to the Internet. *IEEE Commun. Mag.* **2011**, *49*, 96–101.
42. Mottola, L.; Pietro, G. Programming wireless sensor networks: Fundamental concepts and state of the art. *ACM Comput. Surv*. **2011**, *43*, 1–51.
43. Hui, J.; Culler, D. The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale. In *SenSys '04 Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*; ACM: New York, NY, USA, 2004; pp. 81–94.
44. Brown, S.; Sreenan, J. *Updating Software in Wireless Sensor Networks: A Survey*; Tech. Rep. UCC-CS-2006-13-07; Department of Computer Science, University College Cork, Cork, Ireland, 2006.
45. Levis, P.; Culler, D. Maté: A Tiny Virtual Machine for Sensor Networks. In *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems*, San Jose, CA, USA, 5–9 October 2002.
46. Stann, F.; Heidemann, J. RMST: Reliable Data Transport in Sensor Networks. In *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, Anchorage, AK, USA, 11 May 2003.
47. Intanagonwiwat, C.; Govindan, R.; Estrin, D.; Heidemann, J.; Silva, F. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.* **2011**, *11*, 2–16.
48. Hui, J.; Culler, D. The Dynamic Behavior of a Data Dissemination Protocol for network Programming at Scale. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, MD, USA, 3–5 November 2004.
49. Hyun, S.; Ning, P.; Liu, A.; Du, W. Seluge: Secure and DoS-Resistant Code Dissemination in Wireless Sensor Networks. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks*, St. Louis, MO, USA, 22–24 April 2008.
50. TinyOS Community. Deluge T2-TinyOS Documentation Wiki. 16 March 2010. Available online: http://docs.tinyos.net/tinywiki/index.php/Deluge_T2 (accessed on 1 September 2012).
51. Reijers, N.; Langendoen, K. Efficient Code Distribution in Wireless Sensor Networks. In *WSNA '03 Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*; ACM: New York, NY, USA, 2003.
52. Jeong, J. Incremental Network Programming for Wireless Sensors. In *Proceedings of 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Networks and Communications (SECON 2004)*, Santa Clara, CA, USA, 4–7 October 2004.
53. Koshy, J.; Pandey, R. Remote Incremental Linking for Energy-Efficient Reprogramming of Sensor Networks. In *Proceedings of the Second European Workshop on Wireless Sensor Networks*, Istanbul, Turkey, 31 January–2 February 2005.
54. Carman, D.; Kruus, S.; Matt, B. *Constraints and Approaches for Distributed Sensor Network Security*; NAI Labs Technical Report #00-010; September 2000.

Chapter 8

Network Admission Control Solution for 6LoWPAN Networks Based on Symmetric Key Mechanisms

This chapter consists of the following paper:

Network Admission Control Solution for 6LoWPAN Networks Based on Symmetric Key Mechanisms

L. Oliveira, J. Rodrigues, A. de Sousa and V. Denisov
IEEE Transactions on Industrial Informatics, vol. 12, no. 6, pp. 2186-2195, 2016.

doi: 10.1109/TII.2016.2601562

According to Journal Citation Reports published by Thomson Reuters, this journal scored ISI journal performance metrics as follows:

ISI Impact factor (2016): 6.837

Article Influence Score (2016): 1.964

Journal Ranking (2016): 2/60 (Automation and Control Systems)
Journal Ranking (2016): 2/105 (Computer Science, Interdisciplinary Applications)
Journal Ranking (2016): 1/44 (Engineering, Industrial)

2186 IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 12, NO. 6, DECEMBER 2016

Network Admission Control Solution for 6LoWPAN Networks Based on Symmetric Key Mechanisms

Luís Miguel L. Oliveira, Joel J. P. C. Rodrigues, *Senior Member, IEEE*, Amaro F. de Sousa, and Victor M. Denisov

***Abstract*—Wireless sensor networks (WSNs) are a promising technology for several industrial and quotidian applications. IPv6 is the most consensual solution to connect such networks to the Internet, and 6LoWPAN is the adaptation layer to run IPv6 over WSNs. Self-organization and self-configuration are key characteristics of WSN because they minimize the network configuration efforts and simultaneously increase the network robustness but they can also be exploited to perform security attacks. This paper proposes a network admission control solution for 6LoWPAN WSN that prevents unauthorized nodes from using the network to communicate either with the legitimate nodes and with the Internet, reducing in this way the security attacks that can be performed. The proposed solution includes node presence detection and authentication, administrative node authorization, and data filtering to discard frames from/to unauthorized nodes. It uses the standard 6LoWPAN neighbor discovery and RPL protocols, minimizing the number of additional required control messages. It includes cryptographic mechanisms, based on the AES symmetric key algorithm, to guarantee node authenticity and integrity, source authenticity, and data freshness of data frames. This paper also presents the design and deployment of a laboratory testbed validating the proposed network admission control solution.**

***Index Terms*—6LoWPAN, Internet of things, network admission control, wireless sensor networks (WSN).**

Manuscript received September 23, 2015; revised April 21, 2016 and July 2, 2016; accepted July 30, 2016. Date of publication August 18, 2016; date of current version December 6, 2016. This work was supported in part by the Instituto de Telecomunicações, Next Generation Networks and Applications Group, Portugal, in part by the Government of Russian Federation under Grant 074-U01, and in part by National Funding from the Fundação para a Ciência e a Tecnologia through the UID/EEA/500008/2013 Project. Paper no. TII-16-0150. *(Corresponding Author: J. Rodrigues.)*

L. M. L. Oliveira is with the Instituto de Telecomunicações, University of Beira Interior, Covilhã 6201-001, Portugal, and also with the Superior School of Technology, Polytechnic Institute of Tomar, Tomar 2300-463, Portugal (e-mail: loliveira@it.ubi.pt).

J. J. P. C. Rodrigues is with the National Institute of Telecommunications, Inatel, Santa Rita do Sapucaí 37540-000, Brazil, and also with the Instituto de Telecomunicações, University of Beira Interior, Covilhã 6201-001, Portugal and with University ITMO, St. Petersburg 197101, Russia. (e-mail: joeljr@ieee.org).

A. F. de Sousa is with the Department of Electronics, Telecommunications and Informatics, University of Aveiro, Aveiro 3810-193, Portugal (e-mail: asou@ua.pt).

V. M. Denisov is with the University ITMO, St. Petersburg 197101, Russia (e-mail: 070255@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2016.2601562

I. Introduction

WIRELESS sensor networks (WSN) are widely used in embedded applications such as environmental monitoring, surveillance, smart grids, industrial, and home automation. These applications often require a large number of small devices to cover large areas and must operate unattended for long periods of time. Many wireless sensor devices are compliant with the IEEE 802.15.4 standard [1] and are characterized by small size, power constrained, small computing and storage resources, and reduced radio ranges and throughput. Currently, the IPv6 protocol is the most consensual solution to connect WSN to the Internet [2], and 6LoWPAN [3] is the proposed adaptation layer to support IPv6 over IEEE 802.15.4 networks. Moreover, new protocols were also proposed which are best fitted to low power and resource constrained devices, such as the 6LoWPAN neighbor discovery (ND) [10] and the IPv6 routing protocol for 6LoWPAN (RPL) [4].

Self-organization and self-configuration are some of the main characteristics of WSN. These characteristics are of key importance because they minimize the network configuration efforts and simultaneously increase the network robustness in case of failures or topology changes. However, they can also be used to perform security attacks, because third party nodes can easily join the network [5]. With the advent of the Internet of things and the increased interest in deploying WSN based on open protocols, like IPv6, the same security requirements of the unconstrained networks must be extended to WSN. However, WSN exhibits a larger number of vulnerabilities and unique new security challenges, which make them even more vulnerable to security attacks than unconstrained networks [6]. Consequently, the application of traditional security techniques to WSN is not straightforward. Most of the security challenges are related with the resource constrained nature of WSN devices where efficient public-key cryptography and fast symmetric ciphers must be used with care due to the power consumption they require. Additionally, any message overhead caused by security mechanisms must also be minimized whenever possible. In recent works, cryptography has been proposed to guarantee confidentiality, authenticity, integrity, and availability in WSN [7]. Different security mechanisms have been proposed, some of them defined to address particular well-known attacks [8]. Besides these attacks, it is also important that security mechanisms can withstand to new unknown attacks [9].

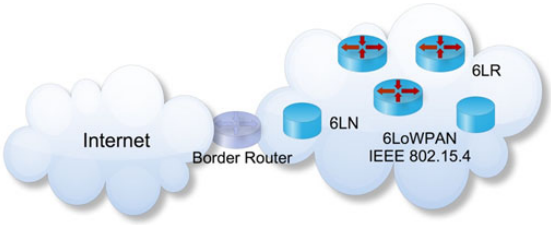


Fig. 1. Illustration of a 6LoWPAN network architecture.

Network admission control can be used to manage the size of the WSN in terms of number of nodes, making the network more manageable, increasing its reliability, and extending its lifetime [5]. Moreover, network admission control can also be used for security since, if a malicious node is prevented from using the network, it cannot communicate with other network elements, and, therefore, the number of possible security attacks is drastically reduced. Our network admission control solution follows this approach by preventing unauthorized nodes to communicate with authorized nodes or to use the WSN to communicate with the Internet. The proposed solution uses the standard 6LoWPAN ND and RPL protocols, minimizing the number of additional required control messages. Moreover, the advanced encryption standard (AES) symmetric key algorithm is used to guarantee authenticity of legitimate nodes and integrity, source authenticity, and data freshness of data frames. The proposed solution can be used as a component of a global network access control solution, as proposed in [5]. A laboratory testbed validates the proposed solution.

This paper is organized as follows. Section II presents the most relevant security issues and technologies related to WSN. Section III focuses on the system architecture of the proposed network admission control solution. Section IV describes the laboratory testbed and presents the validation conclusions. Section V presents the main conclusions of this study and pinpoints future research directions.

II. Security on WSN

As shown in Fig. 1, a 6oWPAN WSN [3] consists of a set of nodes (named 6LN or 6LoWPAN Nodes), a set of routers (named 6LR or 6LoWPAN Routers), and one or more border routers (named 6LBR or 6LoWPAN border routers).

Nodes (or 6LNs) are reduced function devices [1] which are used mainly for sensing and actuation operations and do not forward frames from other nodes. Routers (or 6LRs) are full function devices that, besides being used for sensing and actuation, are also intermediate nodes that forward frames from others nodes to their destination in the same WSN. Border routers (or 6LBRs) are the interconnection elements between the WSN and other networks, such as the Internet. Nodes and routers are resource-constrained devices equipped with small size batteries, while border routers are mainly powered with much more computational resources.

Both WSN and resource unconstrained networks have almost the same security requirements [8]. However, the node resource constraints, the possible huge number of nodes and the lack of an organized communication infrastructure make security more challenging in WSN than in resource unconstrained networks [5]. A standard approach to ensure message authenticity, integrity, and confidentiality is to use an end-to-end security protocol such as IPsec or SSH [11]. Such approach ensures that only legitimate nodes can communicate between them but does not prevent illegitimate nodes to communicate between them or to use the WSN to communicate with the Internet. In multihop WSN, this approach penalizes the overall energy consumption of intermediate routing nodes. A better solution requires the WSN nodes to prune traffic generated by illegitimate nodes, in order to save energy and CPU resources [12].

Confidentiality, authenticity, integrity, availability, data freshness, robustness, and survivability are the most relevant security requirements both in resource unconstrained networks and in WSN [8]. Confidentiality ensures that only authorized entities have access to the data transmitted and stored in the WSN nodes. Authenticity ensures to the receiver that data are provided by the claimed source node identity. Integrity guarantees that data are not changed by any other entity, without being detected. Availability ensures that services provided by the network are always available to their legitimate nodes. Data freshness prevents the nodes to consider previous replayed messages as new messages. Network robustness and survivability guarantees that the network remains operational even in the presence of attacks.

There have been different criteria to classify security attacks [13]. In one hand, they can be classified as external or internal, according to the ownership of the resources used to perform the attack. In external attacks, the attacker uses its own resources to perform the attack. In internal attacks, the attacker first compromises one legitimate node (usually, by the injection of malicious code or by accessing important data stored on them), and, then, uses its resources to perform the attacks. Internal attacks are much harder to detect because the compromised node appears as legitimate to the other nodes, and, therefore, has their trust. On the other hand, security attacks can also be classified as passive or active based on the modification of the regular data streaming [8]. Passive attacks are based on message gathering without modification, while active attacks involve modification on the legitimate data stream and/or false data injection.

Finally, security attacks can also be clustered in three main groups according to their security requirements [5]: attacks against secrecy and authenticity, attacks against network availability, and stealthy attacks. Eavesdropping, packet replay, tampering, and spoofing are examples of the first group. Denial of service is the main example of an attack against network availability. In the stealthy attacks, the attacker successfully injects fake data into a legitimate node, therefore becoming a compromised node, which consequently, spreads fake data over other legitimate nodes.

The research on security solutions has been focused on four areas: security key management, authentication, secure routing, and secure services. Some solutions were proposed to establish and manage cryptographic keys between nodes to enable authentication and encryption mechanisms, while others were proposed to protect routing protocols [14].

Both software-based and hardware-accelerated cryptography implementations were proposed to WSN. A well-known example is TinySec, a link layer security architecture used to guarantee message confidentiality, integrity, and authenticity [15]. TinySec is a pure software-based implementation integrated in TinyOS. It supports two security options: authentication with encryption and authentication without encryption. In the first case, TinySec encrypts the frame payload and authenticates the frame with a message authentication code (MAC) based on the encrypted data and the frame header. In the second case, TinySec authenticates the entire frame with a MAC but the frame payload is not encrypted. Initially, both 3-DES [16] and AES [17] algorithms were evaluated for TinySec but they were shown to be too heavy. Subsequently, RC5 and Skipjack algorithms were evaluated and the tests have shown that they are more suitable for software-based implementations. Skipjack was chosen because RC5 is patented and requires more RAM. To guarantee message integrity, TinySec uses the Cipher Block Chaining MAC mode of operation (CBC-MAC) for computing and verifying MACs.

IEEE 802.15.4 [1] defines four operational cryptographic modes: 1) no cryptography, 2i) authentication-only with CBC-MAC, 3) encryption-only with counter mode (CTR), and 4) authenticated encryption with counter and CBC-MAC mode (TinySec supports all these modes). Some of the wireless chipsets compliant with IEEE 802.15.4 provide accelerated hardware to run cryptography algorithms according to these modes. The Chipcon CC2420 chipset [18] is the most popular radio chipset to WSN devices and provides two types of security operations: standalone encryption operation and inline security operation. The standalone encryption operation uses AES encryption, with 128 bit message blocks and 128 bit length keys. The inline security operation provides encryption and authentication. This chipset supports all IEEE 802.15.4 operational cryptographic modes. AMSecure is an implementation module included in TinyOS that uses hardware-accelerated cryptography taking advantage of cryptographic hardware embedded in Chipcon CC2420 chipset.

There are many proposals on security mechanisms that address only single attacks or to be used in particular layers as security tools. Nevertheless, solutions addressing only a single attack or a single layer result in waste of resources and energy consumption. Recently, researchers are focused on developing security-integrated frameworks that can address multiple security attacks or to use multiple security mechanism, such as Security Protocols for Sensor Networks [19]. Currently, network admission control is seen as an important security mechanism in WSN because it can be used to prevent malicious nodes from joining the network and launching inside attacks [5]. With network admission control, the aim is that only authorized nodes can access the network, and, consequently, the malicious nodes cannot exchange data with legitimate nodes or use the WSN to communicate with the Internet.

A key aspect of any network admission control mechanism is authentication. There are several properties that should be pursuit by authentication protocols [20]: 1) resistance against node compromise, 2) low-computation overhead, 3) low communication overhead, 4) robustness to message loss, 5) immediate authentication, 6) messages sent at irregular times, and 7) high message entropy. In a symmetric key-based mechanism, a single secret key is used by all nodes. In this case, a third party device able to compromise a legitimate node can discover the secret key used by all legitimate nodes. On the other hand, the asymmetric mechanisms involve a private key associated with each node and a compromised node does not allow the attacker to discover the private keys of the other nodes. In fact, several problems related with authentication on broadcast environments can be solved if asymmetric mechanisms are used [19].

Asymmetric cryptography mechanisms require high computation and storage overhead when compared with symmetric key mechanisms. So, their use on WSN must be carefully tested. Moreover, symmetric key-based mechanisms can be made more robust if associated with tampering resistance mechanisms. The use of such mechanisms goes back centuries. Naval codebooks were weighted so they could be over boarded if capture was imminent. Nowadays, this function is usually implemented by a small security processor, such as a smartcard, to store cryptographic keys and where some control functions are executed to avoid its illegal usage.

III. System Architecture

In this section, we describe the system architecture of a WSN network admission control solution able to prevent unauthorized nodes to use the network infrastructure in both single and multihop networks. The proposed solution is based on two key components. The first component is a data frame filtering function, running on all WSN devices, that only accepts data frames from/to legitimate nodes. The second component is a distributed mechanism that sets the required information on all legitimate nodes for letting them to identify the legitimate data frames. This information is updated by the network admission control whenever a new legitimate node is connected to the WSN.

The proposed solution uses the AES symmetric key algorithm to guarantee 1) the authenticity of the new nodes and 2) the source authenticity, data integrity, and data freshness of data frames. It requires one private key per node, named node preshared key to guarantee the node authenticity when first joining the network and a unique global key used in the data frames filtering. For the authenticity of new nodes, we adopt a private key per node which must be known (i.e., preshared) only by the node and by the border router. Its use is restricted to the initial admission of the node to the WSN. For data frame filtering, we adopt a single global key, which is initially set in the border router and is dynamically provided by the border router to each new joined node. As discussed in Section II, the use of a single network-wide secret key (i.e., the global key) is best suited to WSN since data frame filtering has a great impact on the devices resource and energy consumption. Nevertheless, we also consider that legitimate devices have a tampering resistance mechanism. In our testbed, all sensor devices are protected by a box that, when improperly opened, triggers an external electronic circuit (placed inside the box) applying a high voltage to the CPU and memory chips, thus avoiding the external access to the stored security keys.

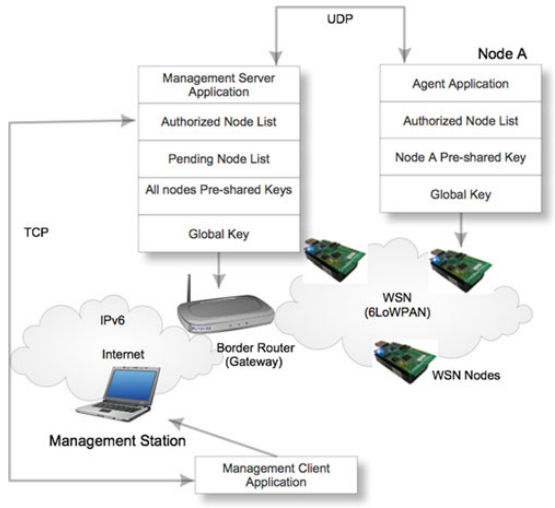


Fig. 2. Admission control entities and components.

The proposed network admission control solution is illustrated in Fig. 2. Its architecture is composed by the following entities: 1) the sensor nodes, compliant with the IEEE 802.15.4 standard and with 6LoWPAN ND and RPL support (the RPL is supported only in full function devices); 2) a border router, connecting the WSN to the Internet, compliant with the IEEE 802.15.4 standard and with 6LoWPAN ND and RPL support, and 3) the Management Station, placed outside the WSN and used primarily for remote border router administration.

Fig. 2 identifies the components added to each entity to support the proposed solution. The management server application, running in the border router, and the agent application, running in the sensor nodes, are used to exchange all required control messages over a UDP control channel. the authorized node list is the list of nodes that were administratively authorized. The pending node list, in the border router, is the list of legitimate nodes waiting for administrative authorization. The preshared keys and the global key were already described. Finally, the management client application, running on the management station, enables the remote administration of the border router.

From the moment, a sensor device is acquired until it is running on its target place, the proposed solution comprises five sequential steps: 1) node provisioning, 2) node presence detection, 3) node authentication and authorization, 4) propagation of the authorized node list, and 5) data filtering. In the following, we describe separately each step and how the components added to the different entities (as presented in Fig. 2) are managed on each of them.

A. Node Provisioning

In an ideal plug-and-play WSN, the sensor devices automatically detect the network presence, autoconfigure their network stack and their firmware to access the network services, without human intervention, no matter their firmware and manufacturer. Nevertheless, the plug-and-play is hard to be supported because some specific firmware is always required in order to accomplish the device objectives in terms of sensing and actuation.

Node provisioning designates the node configuration that is conducted before it is put on its target place. The appropriate firmware is manually installed connecting the device to a provisioning station using a wired connection such as a JTAG or a USB connector. Note that TinyOS is a monolithic OS, and, therefore, a new firmware must be always installed when some new software component is required and our network admission control solution requires the installation of new components. We use the node provisioning step to configure the preshared key of the device. For each new device, a new 128 bits preshared key is generated which is stored in the management server database, with the associated physical address, and embedded in the device firmware.

B. Node Presence Detection

The node presence detection is the step from the moment; the device is initialized on its target place until its presence is detected by the border router. This step is based on the 6LoWPAN ND and RPL protocols. The border router plays a key role in 6LoWPAN WSN. Besides being responsible for connecting the WSN to outside, it is also responsible for propagating the IPv6 prefixes and IPv6 header compression context information. Moreover, the border router maintains a network-wide cache of active IPv6 addresses and EUI-64 identifiers, which enable it to perform link layer address resolution and duplicate addresses detection.

When the interface on a new node is initialized, an IPv6 link-local address is configured based on the interface EUI-64 identifier. Next, the node sends a ND router solicitation message which is replied by the border router with a ND router advertisement (RA) message. The RA message provides all necessary information for the node to configure itself with a global IPv6 address. Once the IPv6 address has been self-configured, a ND neighbor solicitation message, with an address registration option, is sent by the node to the border router in order to register its addresses. In full function devices, the RPL protocol makes also the node to announce its IPv6 address to the whole WSN. In this way, the border router detects accurately the presence of the new node and the routing tables of all WSN nodes are set appropriately to support the communications with the new node without the need for additional messages.

C. Node Authentication and Authorization

This step lasts from the moment the node presence is detected by the border router until the node admission is administratively authorized. First, the node authenticity is validated using a mechanism similar to the one-way challenge authentication protocol. Fig. 3 illustrates this procedure with the new Node A. The first 3 interactions are the three 6LoWPAN ND messages belonging to the previous step. The node authentication is supported by the two last messages, named authentication messages.

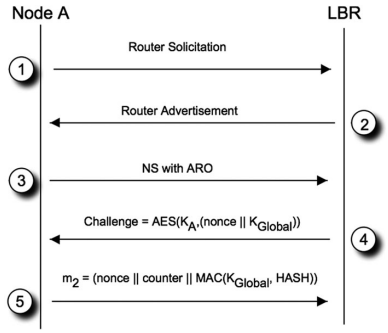


Fig. 3. Node presence and authentication steps.

Fig. 4. Fields used to calculate the HASH.

In order to minimize the number of required messages, the two authentication messages also include the provision of the global key to the new node. This process is as follows. First, the border router generates a random 128 bit number, designated by nonce, concatenates it with the global key (K_{Global}) and uses AES to encrypt the concatenated sequence with the preshared key of Node A (K_{A}). The resulting cryptogram, designated by Challenge, is given by

$$\text{Challenge} = \text{AES}(K_{\text{A}}, (K_{\text{Global}}||\text{nonce})) \tag{1}$$

and is sent by the border router to the new Node A (message 4 of Fig. 3). Note that the *Challange* can only be deciphered by Node A (i.e., eavesdropping of this message by third party nodes does not let them know the global key).

Node A deciphers the *Challenge* using its preshared Key. Then, it composes a message m_2 with the *nonce* received from the border router and a local generated incremental *counter*. Message m_2 is authenticated with a MAC generated with the AES algorithm, using the global key (K_{Global}), as

$$\text{MAC} = \text{AES}(K_{\text{Global}}, (\text{counter}||\text{HASH})) \tag{2}$$

where the HASH is a fingerprint of the frame calculated using the entire IEEE 802.15.4 frame from the FCF to the FCHK fields (see Fig. 4).

The *counter*, used in m_2, can be used for data freshness, i.e., for preventing security attacks based on replaying previous frames. Message m_2 is sent (message 5 of Fig. 3) as a response to the received *Challenge*. Upon reception of m_2, the Border Router verifies the authenticity of Node A by checking if the received *nonce* is equal to its own one. Also, m_2 authenticity, verified by the received MAC using the global key, lets the border router know that the global key has been known by Node A (once again, eavesdropping of this message does not allow to know the global key). If node authentication is successful, the border router inserts IPv6 address of Node A on its pending node list.

Afterward, the administrator uses the management client application, hosted on the management station, to authorize Node A based on its address. If authorization is granted, Node A address is moved from the pending node list to the Authorized node list. Note that this authorization step can be made automatic since node authentication has been already checked. Nevertheless, administrative authorization enables the administrator to verify if the time instant of node detection is consistent with the time instant of node activation and gives him the possibility to prevent nodes to access the network when they are initialized unexpectedly.

D. Propagation of the Authorized Node List

Whenever the authorized node list changes in the border router, it is propagated to all currently active legitimate nodes. A UDP control channel, established between the management server application (on the border router) and each Agent Application (on each WSN node), is used by the border router to update the authorized node list on all nodes. A message with the authorized node list is sent both periodically and when the list is changed. Upon reception of this message, each node replies with an acknowledgement message. An error message is sent by the management server application to the agent application, if it does not receive the acknowledgement message after three retries. All these messages are authenticated with a MAC generated according to (2) in the same way as to m_2 messages, providing in this way source authentication and data integrity.

E. Data Filtering

As previously explained, data filtering is one of the key components of the proposed solution. It runs on all legitimate nodes and makes them to accept incoming data frames that are only from/to legitimate nodes. For each incoming data frame, the node first verifies the frame source identity and/or the data frame integrity. To do so, the source nodes must authenticate their data frames in the following way. On each data frame, the source node adds a local generated incremental *counter* to the frame payload and authenticates the frame with a MAC field based on the global key. In here, we consider two options depending if data integrity is or is not required. In the first option, the data frame integrity is required and the MAC is generated according to (2) in the same way as to m_2 messages. In the second option, data frame integrity is not required and the frame is authenticated with the AES algorithm as

$$\text{MAC} = \text{AES}(K_{\text{Global}}(\text{counter}\,||\text{PHY\_Addr}) \tag{3}$$

where PHY_Addr is the node physical address, instead of being the HASH of the frame as in (2). In practice, this authentication option requires a much lower computation effort and the two options represent different tradeoffs between security and resource and energy consumption.

All other nodes use the *counter* field (for data freshness) and either (2) or (3), depending of the adopted security option, to compute the valid MAC for each incoming data frame. The

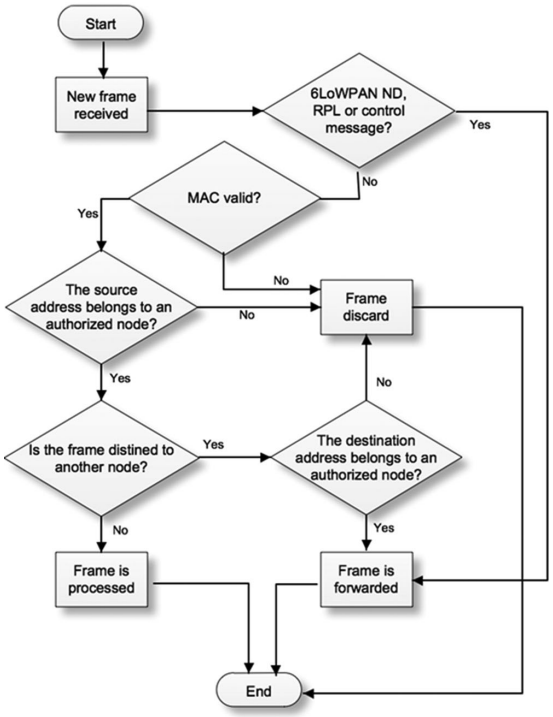


Fig. 5. Data filtering function.

MAC computed by the node is compared with the MAC value carried in the frame. If both values are equal, the sender is authentic (and, in the second option, the frame was not changed in transit); otherwise, the frame is dropped.

The authentication provided by the MAC of the incoming data frames is only the first frame validation. The data filtering function is a decision process whose algorithm flowchart is presented in Fig. 5. If the MAC validation is successful, the frame is still discarded if one of its addresses (origin or destination) is not in the authorized node list. In this way, the receiving node only accepts incoming frames if both the source and the destination addresses belong to the authorized node list. Note that the 6LoWPAN ND, RPL, and control messages (exchanged in the node authentication and in the authorized node list propagation steps) are never discarded since a new node can only become an authorized node after all steps of the network admission control being concluded and, therefore, all messages associated to these steps must be always forwarded.

The use of the authorized node list might seem redundant since when the MAC validation of an incoming frame is successful, it means that the frame source node knows the global key and, therefore, must be a legitimate node. The authorized node list is required, though, to provide full administrative control. If for some reason an authorized node has to be prevented from using the network, without the authorized node list, it is impossible to do it since the node has already been provided with the global key. In our solution, the border router propagates a

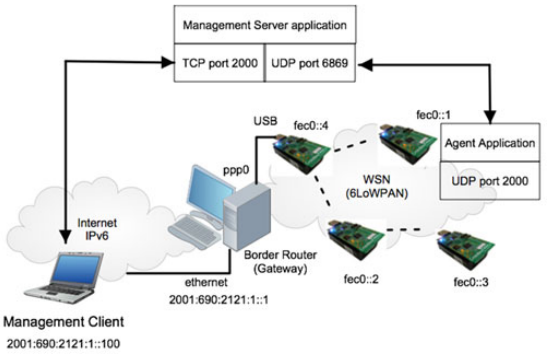


Fig. 6. Laboratory testbed.

new authorized node list without the address of the authorized node and all other nodes start dropping data frames to/from that node.

Finally, note that, since both 6LoWPAN ND and RPL protocols are active, the node routing tables can include entries to unauthorized nodes and the node neighbor lists can include unauthorized node addresses. So, data filtering must be such that its dropping decisions must always override any standard decision based on the routing table or on the standard neighbor list.

IV. Laboratory Testbed

A laboratory testbed was implemented in order to validate the proposed network admission control solution. The testbed scenario, shown in Fig. 6, comprises three sensor nodes, a border router (gateway), and a management client station. Four TelosB motes were used, three of them as sensor nodes and the fourth one as the border router WSN interface.

This section presents the testbed deployment details and the validation results. In the following sections, we describe separately 1) the management system, running between the management client and the border router, 2) the TinyOS solution components added on the TelosB motes, and 3) the validation tests conducted on the testbed.

A. Management System

The management system is composed by the management client application, running on the management client, and the management server application, running on the PC side of the border router. A multiplatform IPv6 management client application was developed that communicates with the management server application primarily to manage the authorized node list but also to run debugging tasks. The management client application was developed using Java IDE NetBeans and uses TCP over IPv6 for communications. In the testbed, the management client application platform (a regular PC with Windows 7 OS) is connected to the border router using an Ethernet connection.

The Management server application, running in the PC side of the border router, was developed in Java and operates on the TCP port number 2000. This application is responsible for 1)

receiving commands from the management client application, parsing the commands to appropriate TinyOS commands which are sent by the ppp0 port (through the USB interface, see Fig. 6) to the TelosB mote running as the WSN interface, 3) receiving TinyOS responses from the ppp0 interface, and 4) sending the received responses back to the management client application. The management server application also implements the node provisioning, which includes: 1) Preshared key generation, 2) copy of the preshared key and the node physical address to the management server application database, 3) copy of the preshared key to the TinyOS source code, and 4) source node firmware compilation and installation.

B. Border Router and WSN Nodes

The implementation of the proposed solution has involved 1) the modification of the TinyOS IP protocol stack and 2) the development of the management server application, running on the TelosB mote of the border router, and the agent application, running on each sensor.

The standard IPv6 protocol stack of TinyOS is composed by four layers: protocol, routing, ND, and dispatch. Each of these layers has its own component. In the sender side of a node, the protocol layer, implemented by the IPProtocolsP component, takes an upper layer UDP datagram as input and generates the IP packet adding the appropriate IP header. Then, the routing layer, implemented by the IPForwardingEngineP component, takes the IP packet as input and makes the routing decision based on its routing table. If the decision is to forward the IP packet to another node, the IP packet is submitted to the ND layer, implemented by the IPNeighborDiscoveryC component, in order to identify the link layer address of the next node. Finally, the IP packet is submitted to the dispatch layer, implemented by the IPDispatchC component, responsible for the 6LoWPAN adaptation layer and for delivering the resulting 6LoWPAN packets to the IEEE 802.15.4 layer. In the receiver side, the dispatch layer receives from the lower IEEE 802.15.4 layer the 6LoWPAN packets, recovers the IP packets, and deliver them to the protocol layer, which decides if the IP packets are to be delivered to the upper UDP layer or to be forwarded to another node.

The proposed network admission control solution has required the modification of the IPForwardingEngineP and IPDispatchC components. The routing layer, implemented by the modified IPForwardingEngineP component, still considers the standard routing table (dynamically set by the RPL protocol) but includes also the authorized node list. For each IP packet submitted to the routing layer, the modified component first checks if the IPv6 addresses are in the authorized node list. If this check is successful, it makes the routing decision based on the routing table; otherwise, it drops the IP packet. The dispatch layer, implemented by the modified IPDispatchC component runs the data frame authentication. For the incoming data frames, it first checks the frame authenticity. If successful, the IP packets are delivered to the protocol layer; otherwise, they are discarded. For the outgoing frames, it computes and adds the appropriate authentication fields before delivering the frame to the IEEE 802.15.4 layer: 1) for the IP packets such that the node is the packet source, a local generated incremental *counter* is used and 2) for the IP packets such that the node is acting as an intermediate router, the *counter* value of the incoming packet is maintained when it is forwarded to the next node.

Concerning the development of the management server and agent applications, they were implemented as modifications of the UDPEcho application available in TinyOS. In its basic version, UDPEcho can be used in one node to send a given message for a given IPv6 address: 1) the UDPEcho local application sends the message over UDP to the destination address and 2) the remote node replies back with a UDP datagram with the same message.

The UDPEcho was modified is the following way. For some particular messages, upon reception of the message in the remote node, it interprets the message as a command, runs locally the command and, instead of returning the received message, it returns the result of the command. This method was used for the exchange of the control messages (the two authentication messages and the authorized node list dissemination messages). When the new node authentication is successful, the global key is set by the agent application into a variable of the IPDispatchC component (to enable the dispatch layer to run the data frame MAC validations). When the dissemination messages of the authorized node list are received by the agent application, it updates the local authorized node list copy of the IPForwardingEngineP component.

For debugging purposes, we have also included two additional commands in the modified version of the UDPEcho application: 1) one for executing a ping6 command, based on nc6 tool, from a remote node to another node and 2) one for retrieving the current routing table and authorized node list of a remote node. The management server application (at the border router side) uses UDP port number 6869 and the agent application (at the WSN nodes side) uses UDP port number 2000.

The implementation effort of the proposed network admission control solution was very low. The number of additional source code lines is 200 in IPForwardingEngineP and 70 in IPDispatchC. In addition, the source code of the modified UDPEcho version (including the debugging exchanged messages) has 100 additional lines. In all cases, the number of additional lines includes the comment lines.

To run the AES algorithm, we have resorted to a software-based AES implementation developed by the École Polytechnique Federal of Lausanne (the source code of this implementation has a total of 2000 code lines including comment lines). Next, we describe how the AES-based authentication operations have been implemented. In the node authentication step, the border router sends a message with a *Challenge* following (1), as presented in Section III. In the implementation, the *Challenge* is computed with the function `encrypt` whose interface is

```
void encrypt(uint8_t *in_block, uint8_t
*expkey, uint8_t *out_ block)
```

where `in_block` is the concatenation of a local random generated *nonce* and the global key, `expkey` is the preshared key of the node to be authenticated and `out_block` is the resulting *Challenge* (prototypes are presented in nesC, the programming

language used by TinyOS). Upon reception of the *Challenge*, the node retrieves the *nonce* and the global key with the function `decrypt` whose interface is

```
void decrypt(uint8_t *in_block, uint8_t
*expkey, uint8_t *out_block)
```

where `in_block` is the received *Challenge*, `expkey` is the node preshared key, and `out_block` is the concatenation of the *nonce* and the global key.

The MAC field that authenticates both m_2 authentication message and all messages of the authorized node list propagation must follow (2), as presented in Section III. In the implementation, this is done with two functions. First, the HASH of the frame is computed with the function `hash` whose interface is

```
uint32_t hash(uint32_t *message, uint32_t
*key, uint16_t l)
```

where `message` is the IEEE 802.15.4 frame content from the FCF to the FCHK fields (see Fig. 4) and `key` is the global key. The function returns the HASH. Then, the MAC of the frame is computed using again the fucntion

```
encrypt    void encrypt(uint8_t *in_block,
uint8_t *expkey, uint8_t *out_block)
```

where now `in_block` is the concatenation of a local generated incremental *counter* with the previous HASH, `expkey` is the global key, and `out_block` is now the resulting MAC. In the source node of the frame, the *counter* and the MAC are inserted at the beginning of the data payload. In all intermediate nodes (including the destination), the MAC is computed in the same way and the frame is accepted if the computed MAC is equal to the received one.

The data frame authentication, when data integrity is required, is processed by the source nodes and all other nodes as described above. When data integrity is not required, the process is similar and the only difference is that in the `hash` function, the input variable `message`, instead of containing the IEEE 802.15.4 frame content, it contains only the physical address of the source node.

C. Validation Tests

The tests were conducted in the setup presented in Fig. 4 where, instead of setting the border router announcing IPv6 prefixes, we have manually configured the IPv6 addresses on the nodes in order to make easier their identification on the resulting configurations. WSN nodes were configured with the site-local addresses fec0::1, fec0::2, and fec0::3 and the border router was configured with the site-local address fec0::4. Note that besides the manually configured address, each node configures its interface with an IPv6 link-local address based on its interface EUI-64 identifier (the addresses with prefix fe80::0/64). The following experimental procedure was conducted in order to validate all functions associated with our network admission control solution.

In the first experiment, we validate the correctness of the presence detection of new nodes by the border router. The three sensor nodes were activated but no authorization was granted by the administrator. Fig. 7 shows the resulting border router IPv6 routing table, together with the authorized node list (at each table entry, column OnNet is 0 if the node is in the Authorized Node List, or 0 otherwise). The routing table includes an entry for the IPv6 address of each node (i.e., the network is prepared to support the communications with all nodes) but all IPv6 addresses are unauthorized. This state remains unchanged until the administrator grants authorization to some of the addresses.

```
Key SiteLocal/Mask  Next_Hop _Address       OnNet   Iface   Link_Local_Address
1   ff02::1:2/128   ::  0   ppp null
3   fec0::1/128     fe80::212:6d45:5065:f748    0   pan fe80::212:6d45:5065:f748
4   fec0::2/128     fe80::212:6d45:5066:1da4    0   pan fe80::212:6d45:5066:1da4
5   fec0::3/128     fe80::212:6d45:5066:1da4    0   pan fe80::212:6d45:50e2:4421
2   ::/0    ::  0   ppp null
                        My Link Local Address is [fe80::212:6d45:50e3:31dc]
```

Fig. 7. New node detection.

```
(fec0::1>command): route able fec0::1 fe80::212:6d45:5065:f748
(fec0::1>response)...
Key SiteLocal/Mask  Next_Hope_Address       OnNet   Iface   Link_Local_Address
1   ff02::1:2/128   ::  0   ppp null
3   fec0::1/128     fe80::212:6d45:5065:f748    1   pan fe80::212:6d45:5065:f748
4   fec0::2/128     fe80::212:6d45:5066:1da4    0   pan fe80::212:6d45:5066:1da4
5   fec0::3/128     fe80::212:6d45:5066:1da4    0   pan fe80::212:6d45:50e2:4421
2   ::/0    ::  0   ppp null
                        My Link Local Address is [fe80::212:6d45:50e3:31dc]
```

Fig. 8. Node authorization.

```
ping6 fec0::2
(fec0::1>command): ping6 fec0::2
(fec0::1>response)...
Key SiteLocal/Mask  Next_Hope_Address       OnNet   Iface   Link_Local_Address
1   ff02::1:2/128   ::  0   ppp null
3   fec0::1/128     fe80::212:6d45:5065:f748    1   pan fe80::212:6d45:5065:f748
4   fec0::2/128     fe80::212:6d45:5066:1da4    0   pan fe80::212:6d45:5066:1da4
5   fec0::3/128     fe80::212:6d45:5066:1da4    0   pan fe80::212:6d45:50e2:4421
2   ::/0    ::  0   ppp null
                        My Link Local Address is [fe80::212:6d45:50e3:31dc]
```

Fig. 9. Connectivity between authorized and pending nodes.

In the second experiment, we validate the correctness of the administrative authorization of a node whose IPv6 address is in the pending node list. Taking as the initial stage, the network state at the end of the first experiment, the administrator changes the state of the address fec0::1 to authorized. The command *route able <list of node addresses>* implements this authorization decision for multiple addresses in a single configuration step (since each node has two assigned IPv6 addresses, both addresses must be authorized). Fig. 8 shows, first, the execution of the command to authorize both addresses (the site-local address fec0::1 and the link-local address fe80::212:6d45:5065:f748) of the sensor node and, then, the resulting border routing configuration. Now, the routing table has the value 1 in the OnNet column associated to the address fec0::1.

In the third experiment, we check the connectivity between an authorized node and a node whose IPv6 address is in the pending node list. Taking as the initial stage, the network state at the end of the second experiment, Fig. 9 shows the result of a ping6 command executed in the sensor node fec0::1, which was previously authorized, to the address fec0::2 which is currently in the pending state. The absence of responses indicates that there is no connectivity between them. Fig. 9 also shows the routing table of the border router which remains unchanged since the administrator has not granted authorization to other nodes.

```
ping6 fec0::2
(fec0::1»command): ping6 fec0::2
(fec0::1»response)...
fec0::2 icmp_seq=0 ttl=15 time=437 ms
fec0::2 icmp_seq=1 ttl=15 time=61 ms
fec0::2 icmp_seq=2 ttl=15 time=71 ms
5 packets transmitted, 3 received
Key SiteLocal/Mask  Next_Hope_Address       OnNet   Iface   Link_Local_Address
1   ff02::1:2/128   ::  0   ppp null
3   fec0::1/128     fe80::212:6d45:5065:f748    1   pan fe80::212:6d45:5065:f748
4   fec0::2/128     fe80::212:6d45:5066:1da4    1   pan fe80::212:6d45:5066:1da4
5   fec0::3/128     fe80::212:6d45:5066:1da4    1   pan fe80::212:6d45:50e2:4421
2   ::/0    ::  0   ppp null
                    My Link Local Address is [fe80::212:6d45:50e3:31dc]
```

Fig. 10. Connectivity between authorized nodes.

In the last experiment, we check the connectivity between authorized nodes. Taking as the initial stage the network state at the end of the third experiment, first, the administrator changes the state of the two other sensor nodes to authorized. Then, the ping6 command was successfully executed between all sensor nodes. Fig. 10 shows, first, the result of a ping6 command executed in the sensor node fec0::1 to the address fec0::2 and, then, the resulting border router configuration. Now, all routing entries have the value 1 in OnNet column showing that all data frames are being accepted by the frame filtering function running on all nodes. The communication between nodes is successful because their IPv6 addresses are now in the authorized node list and all nodes have received from the border router the updated authorized node list containing all addresses.

V. Conclusion and Future Work

Security is considered as a key factor for the success of WSN. However, the key characteristics of WSN (such as self-organization and self-configuration) together with the resource constrained characteristics of sensor devices, make security provision more difficult on WSN than in unconstrained networks. In this paper, a network admission control solution was proposed and validated which controls the nodes that use the network, based on administrative authorization. Different modifications in the routing and packet forwarding functions of TinyOS were implemented and a modified version of TinyOS 2.1 UDPecho application was developed implementing the node authentication and the authorized nodes list dissemination. Symmetric-key cryptographic mechanisms based on AES were used to guarantee node authenticity and confidentiality, integrity, and data freshness to data frames. The testbed has shown that the implementation effort of the proposed network admission control solution is very low, mainly because it relies on the available 6LoWPAN ND and RPL protocols for the implementation of the network functions not related with security. The validation tests have shown that the proposed solution works correctly while achieving its security goals.

Additional research is still required to improve the proposed solution. Currently, the global key is provided in the node authentication step, with the advantage or minimizing the number of required messages but not allowing the dynamic change of the global key. If a global key change is required, a mechanism to dynamically provide the new global key would prevent the need to reset all nodes. Moreover, while using the 6LoWPAN ND and RPL protocols, the proposed solution is not immune to the security threads of these protocols. An important improvement is to extend the data frame filtering to RPL messages, obtaining, in this way, an integrated solution which provides also security of the WSN routing. Another possible improvement direction is to investigate the possibility of using asymmetric key cryptographic-based mechanisms. Such mechanisms, with cryptographic generated addresses based on elliptic curve mechanisms, might be not as resource consuming as traditional techniques and do not require a global key avoiding the need to use tampering resistance mechanisms.

References

[1] Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Standard 802.15.4-2006, 2006.

[2] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of things: A survey," *IEEE Commun. Surv. Tuts.*, vol. 16, no. 1, pp. 414–454, Jan.–Mar. 2014, doi: 10.1109/SURV.2013.042313.00197.

[3] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over low-power wireless personal area networks (6LoWPANs): Overview, assumptions, problem statement, and goals," Internet Engineering Task Force, Request for comments 4919, Aug. 2007.

[4] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann, "Neighbor discovery optimization for IPv6 over low-power wireless personal area networks (6LoWPANs)," RFC 6775, Nov. 2012.

[5] T. Winter *et al.*, IPv6 routing protocol for low-power and lossy networks, RFC 6550, Aug. 2012.

[6] L. Oliveira, J. Rodrigues, A. de Sousa, and J. Lloret, "A network access control framework for 6LoWPAN networks," *Sensors*, vol. 13, no. 1, pp. 1210–1230, 2013.

[7] X. Du and H. Chen, "Security in wireless sensor networks," *IEEE Wireless Commun.* vol. 15, no. 4, pp. 60–66, Aug. 2008.

[8] Y. Yu, K. Li, W. Zhou, and P. Li, "Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures," *J. Netw. Comput. Appl.*, vol. 35, no. 3, pp. 867–880, 2012.

[9] T. Kavitha and D. Sridharan, "Security vulnerabilities in wireless sensor networks: A survey," *J. Inf. Assur. Secur.*, vol. 5, pp. 31–44, 2010.

[10] K. Pelechrinis, M. Iliofotou, and V. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *IEEE Commun. Surv. Tuts.*, vol. 13, no. 2, pp. 245–257, Apr./Jun. 2011.

[11] K. Gupta and S. Silakari, "ECC over RSA for asymmetric encryption: A review," *Int. J. Comput. Sci. Issues*, vol. 8, no. 3, pp. 370–375, 2011.

[12] H. Tan, J. Zic, S. K. Jha, and D. Ostry, "Secure multihop network programming with multiple one-way key chains," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 16–31, Jan. 2011.

[13] P. Sakerindr and N. Ansari, "Security services in group communications over wireless infrastructure, mobile ad hoc and sensor networks," *IEEE Wireless Commun.*, vol. 14, no. 5, pp. 8–20, Oct. 2007.

[14] W. Gu, N. Dutta, S. Chellappan, and X. Bai. "Providing end-to-end secure communications in wireless sensor networks," *IEEE Trans. Netw. Service Manage.*, vol. 8, no. 3, pp. 205–218, Sep. 2011.

[15] N. Bandirmali and I. Erturk, "WSNSec: A scalable data link layer security protocol for WSNs," *Ad Hoc Netw.*, vol. 10, no. 1, pp. 37–45, 2012.

[16] G. Singh and A. Supriya, "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security," *Int. J. Comput. Appl.*, vol. 67, no. 19, pp. 33–38, 2012.

[17] F. Zhang, R. Dojen, and T. Coffey, "Comparative performance and energy consumption analysis of different AES implementations on a wireless sensor network node," *Int. J. Sens. Netw.*, vol. 10, no. 4, pp. 192–201, 2011.

[18] (2015, Sep.). Texas Instrument. CC2420 Datasheet. [Online]. Available: http://docs.tinyos.net/index.php/CC2420.

[19] C. Xiangqian, M. Kia, Y. Kang, and N. Pissinou, "Sensor network security: A survey," *IEEE Commun. Surv. Tuts* , vol. 11, no. 2, pp. 52–73, Apr.–Jun. 2009, doi: 10.1109/SURV.2009.090205.

[20] H. Tan, J. Zic, S. K. Jha, and D. Ostry, "Secure multihop network programming with multiple one-way key chains," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 16–31, Jan. 2011.

Luís Miguel L. Oliveira is currently working toward the Ph.D. degree in informatics engineering at the University of Beira Interior, Covilhã, Portugal, under supervision by Professor Joel Rodrigues and Professor Amaro de Sousa, the M.Sc. degree in electronics and telecommunications engineering from the University of Aveiro, Aveiro, Portugal, in 2004.

He also teaches in the Informatics Engineering Department, Superior School of Technology, Polytechnic Institute of Tomar, Tomar, Portugal. He is currently a Ph.D. Student Member of the Instituto de Telecommunicações, University of Beira Interior. He is a Consultant on several large-scale networking projects. He authors or coauthors more than ten international conference papers and also has eight accepted journal publications. He has been a reviewer for international journals and conferences. His current research interests include software-defined networks, Internet protocol integration on wireless sensor networks, and wireless sensor networks applications.

Joel J. P. C. Rodrigues (S'01–M'06–SM'06) received the *Academic Title of Aggregated Professor* in informatics engineering from UBI, the Habilitation in computer science and engineering from the University of Haute Alsace, France, a PhD degree in informatics engineering and an MSc degree from the UBI, and a five-year BSc degree (licentiate) in informatics engineering from the University of Coimbra, Portugal. He is currently a Professor and senior researcher with the National Institute of Telecommunications, Inatel, Santa Rita do Sapucaí, Brazil, and a Senior Researcher with the Instituto de Telecomunicações. He has been professor at the University of Beira Interior (UBI), Portugal. He is the leader of the NetGNA Research Group, the President of the scientific council at ParkUrbis—Covilhã Science and Technology Park, Past-Chair of the IEEE ComSoc Technical Committee on eHealth and on Communications Software, Steering Committee Member of the IEEE Life Sciences Technical Community, and a Member Representative of the IEEE Communications Society on the IEEE Biometrics Council. He is the Editor-in-Chief of the *International Journal on E-Health and Medical Communications*, Editor-in-Chief of the R*ecent Advances on Communications and Networking Technology*, Editor-in-Chief of the *Journal of Multimedia Information Systems*, and an Editorial Board Member of several journals. He has been a General Chair and TPC Chair of many international conferences, including the IEEE ICC, GLOBECOM, and HEALTHCOM. He has authored or coauthored more than 500 papers in refereed international journals and conferences, three books, and two patents.

He is a Licensed Professional Engineer (as Senior Member), a Member of the Internet Society, an IARIA Fellow, and a Senior Member ACM.

Amaro F. de Sousa received the five-year B.S. degree in electronics and telecommunications engineering from the University of Aveiro, Aveiro, Portugal, in 1989, the M.Sc. degree in telecommunications engineering from the University College of North Wales, Bangor, U.K., in 1991, and the Ph.D. degree in electrical engineering from the University of Aveiro in 2001.

He is currently an Assistant Professor in the Department of Electronics, Telecommunications and Informatics, University of Aveiro, Aveiro, Portugal, and a Senior Researcher with the Instituto de Telecomunicações—Pole of Aveiro, Aveiro. He has coauthored several papers in refereed international journals and conferences. He has been involved in different European Union funded and Portuguese funded projects more than last 20 years. He has been a reviewer for many international journals and conferences. His research interests include advanced services and protocols for telecommunications, traffic engineering and network design, and optimization algorithms for efficient network resource management.

Victor M. Denisov received the Graduate degree in optoelectronic devices from the Leningrad Institute of Precision Mechanics and Optics, in 1977.

He is currently a Professor in the Department of Optical-Electronic Devices and Systems, ITMO University, Saint Petersburg, Russia. He is the author of more than 80 scientific works and inventions. He is an expert of the Ministry of Education and Science of the Russian Federation in the field of information and computer technologies. He is a Member of the Editorial Board of the *International Journal of E-Health and Medical Communications*. He is also the Chief Designer of the family of the newest field of geophysical sensor devices. Under his leadership made a lot of major projects for the monitoring of complex engineering structures and dangerous natural objects. He developed a new method for geotechnical monitoring based on the use of arrays of micromechanical sensors based on flexible chassis. His main research interests include information and computer technology, instrumentation, measuring devices and systems, geotechnical monitoring, mobile and cloud computing, sensors, intelligent sensor networks, and mobile medicine.

Chapter 9

Conclusions

This chapter presents the main conclusions that result from the research work described in this thesis and also the research topics related to the work developed that can be addressed in further research works.

9.1 Final Conclusions

This thesis addressed the challenges associated with the interconnectivity between the Internet and the IoT devices and with the security aspects of the IoT. The research work was organized in three main parts. The first part is dedicated to the state of the art, conducted at the first stages of the work, on available solutions to support routing and mobility over 6LoWPAN mesh networks and on environmental monitoring using WSN based technologies. The second part is dedicated to the proposal of solutions to provide valuable information to the Internet connected devices, independently of the supported IP protocol version, without being necessary accessed directly to the LLN nodes. The third part is dedicated to the proposal of security mechanisms to mitigate the effects of internal and external initiated attacks, minimizing the overhead on the LLN nodes.

The first part of this research work was presented in Chapter 2 and Chapter 3 of this thesis. Chapter 2 presents a detailed survey on routing and mobility approaches over 6LoWPAN mesh networks. In this survey, a special attention was dedicated to the 6LoWPAN and to the IEEE 802.15.4 protocols. The main conclusions were as follows. The use of the IP protocol stack on all IoT devices is viable even when energy and resource constrained devices are in use. IPv6 protocol, due to its address space size and to the native auto configuration mechanisms, is a consensual solution (i) to connect LLN devices running layer two incompatible protocols, (ii) to facilitate the connection to the Internet and (iii) to the design and deployment of applications. The IEEE 802.15.4 was identified as one of the most relevant layer two protocols for wireless LLNs and, therefore, to the IoT, not only due to its hardware requirements and the energy consumption but also due to diversity of compatible hardware commercially available. Chapter 3 presents a comprehensive review of available solutions and projects on environmental monitoring based on WSN technology. This chapter describes the main advantages of using cheap and less accurate devices compared to traditional monitoring stations on environmental monitoring. In this review, some of the most relevant projects in this area were identified, and the results analysed, and the conclusion used to identify the challenges that needed to be addressed. In fact, environmental monitoring is a challenging IoT application because, in most cases, devices are operating autonomously in harsh and unattended conditions without any predefined network infrastructure. Several open issues and challenges were identified and a special attention was given to the ones related to the Internet integration, while preserving the LLN devices' resources and mitigating the effects of internal and external security attacks.

The second part of this work, presented in Chapter 4 and Chapter 5, addressed the challenges related on how to integrate the IoT in the Internet, providing valuable data to the Internet connected devices and avoiding direct connectivity from them to IoT devices. Chapter 4 proposes a ubiquitous solution that allows Internet connected mobile phones to receive the most up to date sensor readings, as well as historical measures and to be notified whenever a predefined condition occurs. This solution, based on open standards, uses a REST Web service, a relational database and an Android mobile application. The RESTfull webservice, running on the border router, is used to provide efficient data access and a standard application interface to make mobile applications development easier and independent of the hardware and software used in the IoT devices. A push notification system was implemented to send push messages to smartphones if a sensor reading overcomes a predefined threshold. The proposed solution was evaluated using a laboratory testbed and the results showed significant savings on LLN devices and on mobile phone energy consumption. The seamless integration of the IoT in the Internet must consider both IP protocol version running in the Internet. In order to address this requirement, the above described solution was extended to support IPv4 to IPv6 dual stack transition mechanism. Note that none of the IPv4 to IPv6 transition mechanisms is suitable to be used on the LLN devices. In the new solution, detailed in the Chapter 5, the gateway used to connect the LLN to the Internet supports a dual stack IPv4 to IPv6 transition mechanism in order to accept requests both from IPv4 and IPv6 clients.

The third part of this work, presented in the Chapter 6, Chapter 7 and Chapter 8, addressed some of the most relevant IoT's security challenges. Chapter 6 proposes a security mechanism to prevent remotely initiated transport level DoS attacks. The proposed mechanism avoids the use of inefficient and hard to manage traditional firewalls to filter at the border router the traffic received from the Internet and destined to the LLN. The mechanism is supported by the border router and only forwards the Internet traffic into the LLN if it meets predefined conditions announced by each LLN node. In the proposed solution, the LLN nodes use an adapted version of 6LoWPAN neighbour address registration mechanism to notify the border router on the conditions used to filter the Internet received traffic. This mechanism requires no other messages than those used to perform the address registration and also does not increase the length of the messages because it uses not assigned message fields. Several security attacks, both internal and external, initiated by neighbour LLN devices can be avoided if a network access control mechanism is used to restrict the network access only to authorized nodes which are compliant with the defined security requirements. Chapter 7 proposes a network access control framework that can be used to realise these objectives. The proposed framework enables the node identification based on cryptographically generated addresses, node security compliance evaluation and node remediation with secure remote software installation. This solution is mainly based on the following open protocols: (i) LSEND, used to secure neighbour discovery and node secure identification, (ii) RPL with ECC support, to protect routing messages and (iii) Seluge, to enable secure remote software installation. In this proposal, synergies between the involved protocols were taken in consideration. Chapter 8 details a network admission control solution to authorize which nodes can have access to the network based on administrative authorization. Different modifications in the routing and packet forwarding functions of TinyOS were implemented and a modified version of TinyOS 2.1 UDPecho application was developed implementing the node authentication and the authorized nodes list dissemination. Symmetric-key cryptographic mechanisms based on AES were used to guarantee node authenticity integrity and, confidentiality, and data freshness to data frames. The testbed has shown that the imple-

mentation effort of the proposed network admission control solution is very low, mainly because it relies on the available 6LoWPAN ND and RPL protocols for the implementation of the network functions not related with security. The validation tests have shown that the proposed solution works correctly while achieving its security goals.

9.2 Future Work

To conclude this thesis, the next paragraphs detail some future research directions worthwhile to be addressed as future work and which resulted from this research work.

In all contributions of this thesis, the proposed solutions are based on running the appropriate mechanisms on one border router connecting each LLN to the Internet. Due to issues such as scalability and robustness to failures, IoT deployments must evolve to network scenarios where the connection between each LLN and the Internet must be based on multiple border routers. The proposed solutions are not easily extendable to these scenarios and, therefore, such extensions must be carefully studied.

In the network access control solution, as proposed in Chapter 8, there are two aspects that were implemented aiming to minimize the device' energy consumption. One is the use of a single global key, provided in the node authentication step, which minimizes the number of required messages but cannot change dynamically. The other is the use of cryptographic mechanisms, based on the AES symmetric key algorithm, which requires low processing complexity. In the first aspect if a global key change is required, a mechanism to dynamically provide the new key preventing the need to reset all nodes should be investigated. In the second aspect, the possibility of using asymmetric key cryptographic-based mechanisms must be investigated since such mechanisms, with cryptographic generated addresses based on elliptic curve mechanisms, might be not as resource consuming as traditional techniques. Such solution would make the key management operations easier.

Finally, note that, while using the 6LoWPAN ND and RPL protocols, the proposed network access control solution is still vulnerable to the security threads of these protocols. An important improvement worthwhile being investigated is to extend the data frame filtering to RPL messages, obtaining, in this way, an integrated solution which provides also security of the LLN routing. Such solution, though, must be investigated with care in order to reach the optimum trade-off between security gains against additional energy consumption penalties.

Appendix A

This appendix is based on the following paper:

IoT based solution for home power energy monitoring and actuating

L. M. L. Oliveira, J. Reis, J. J. P. C. Rodrigues and A. F. de Sousa 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), Cambridge, 2015, pp. 988-992.

DOI: 10.1109/INDIN.2015.7281869

IOT based Solution for Home Power Energy Monitoring and Actuating

Luís M. L. Oliveira [1,2,3], João Reis [1,4], Joel J. P. C. Rodrigues[1,3,5], Amaro F. de Sousa [1,6]
[1] Instituto de Telecomunicações, Portugal
[2] Instituto Politécnico de Tomar, Tomar, Portugal
[3] University of Beira Interior, Covilhã, Portugal
[4] Withus Inovação e Tecnologia, Aveiro, Portugal
[5] King Saud University, Riyadh, Saudi Arabia
[6] Universidade de Aveiro, Portugal

loliveira@ipt.pt; joao.reis@withus.pt; joeljr@ieee.org; asou@ua.pt

***Abstract*— The current tendency to embed computational resources on quotidian objects transforms them into smart objects. This is the vision of Internet of things, where many different devices collect and process information from different sources to both control physical processes and interact with human users. Initially wireless sensor networks and Internet of things concepts were used with the same meaning. The same happened with wireless sensor networks and the protocol IEEE 802.15.4. It is now accepted that the Internet of tings can comprise more than one type of networks and therefore several layer two protocols. In such scenarios the IP protocol it is used, like as before, to make this heterogeneity interoperable. This paper presents a demonstrator for power energy monitoring and actuating system and it was developed for home environments. This system is based on 6LoWPAN to connect to the Internet a smart object network with two layer 2 technologies: IEEE 802.156.4 and power line communication. IPv4 to IPv6 transition mechanisms were included to provide connectivity to both IPv4 and IPv6 Internet end systems.**

***Keywords*—** 6LoWPAN; IoT; Internet of Things; IEEE 802.15.4; power line communication

I. INTRODUCTION

Kevin Ashton proposed the term of Internet of Things (IoT) before 2000 in the context of supply chain management. Now the IoT it is more inclusive and includes a wide range of applications. Also the concept of IoT network nodes, known as Things, was evolved as a consequence of technology evolution, in particular due the MEMS devices advances.

The IoT it is the driver to change the traditional Internet into a network of interconnected objects with resources to harvest information from the surrounding environment (i.e. sensing) and to interact with the physical world (i.e. actuation and control). In this scenario, the existing Internet protocols are required to support information transfer, analytics, applications and communications. At the beginning, the IoT and the wireless sensor networks concepts were used interchangeably. The same happened with wireless sensor networks and IEEE 802.15.4 layer two protocol. Now IoT it is more embracing thus, it is not restricted to only one device type neither to only one particular layer two technology.

In the IoT networks, some of the devices are embed on quotidian objects and therefore they must have small size, restricted computational resources and energy constraints. In such situations, the IEEE 802.15.4 [1] support may be a requirement because it is a wireless communication protocol and it was designed to operate on computational and energy resource constrained devices. The power line communications (PLC) [2] solution it is the natural layer two protocol to connect power energy monitoring and actuation devices. In such situations, the energy wires are used also for communication purposes and there is no severe restriction about energy consumption because the energy on the wires can be used to feed the monitoring devices. However, the computational resource constrains remains valid because the size and the price of the devices. In home power energy monitoring and actuation solutions IEEE 802.15.4 and PLC are the most promising layer two technologies to provide wireless and wired connectivity respectively [3].

Although initially IP was not considered for IoT environments, the scientific community and the industry have rethink many misconceptions about the use of IP protocol in all nodes [4]. First, the IP protocol can be used as a standard solution for the interconnectivity between incompatible lower layer protocols. Second, it provides an application developing environment, which is open and royalty free. Finally, if all nodes are compatible with IP protocol the use of complex and hard to manage proxies and gateways necessary to connect non-IP nodes to the Internet can be avoided. While the IPv6 protocol has enough address space to accommodate also the IoT devices, it was not originally designed to be used on power and resource constrained nodes. The 6LoWPAN adaptation layer [5] was proposed to be used on such devices between data link and the network layers to make the IEEE 802.15.4 layer two standard protocol compatible with IPv6. It provides new routing approaches, header compression and fragmentation support above 1280 bytes. It also includes auto configuration and neighbor discovery mechanisms, which are more adapted to energy and resource constrained devices.

978-1-4799-6649-3/15/$31.00 ©2015 IEEE

This paper presents a solution based on 6LoWPAN to provide connectivity between incompatible layer 2 IoT devices and also to provide the interaction between this devices and the Internet. Both the IoT nodes and the gateway were developed in house and are based on COTS chipsets. At the application layer, two main applications were developed. The first one is running in the gateway and it is used to manage the IoT connected devices providing it with a resource discovery mechanism. The second one implements the home power energy monitoring and actuating system. The second application can be used to interact with IoT devices, to store and analyse harvested data and to provide a web interface to the users. The last application it is installed on the IoT devices, on the gateway and on the application server. A demonstrator has been constructed to evaluate the proposed solution and to prove their capabilities.

The remainder of this paper is organized as follows. Section II presents the related technologies, while Section III focuses on the system architecture. Section IV presents the system evaluation and demonstration. Finally, Section V concludes the paper and pinpoints future research work.

II. RELATED TECNOLOGIES

A. IEEE 802.15.4

The IEEE 802.15.4 protocol defines the physical and the media access control (MAC) layers to address the low-power and low-rate wireless personal area networks requirements (WPAN). The PHY layer defines three physical operation modes, 20 kbps at 868 MHz, 40 kbps at 915 MHz, and 250 kbps at 2.4 GHz (DSSS). The MAC layer provides two operational modes, the synchronous beacon-enabled mode and the asynchronous beaconless. The beacon-enabled mode is designed to support the transmission of beacon packets between transmitter and receiver, providing synchronization among nodes. In the beacon-enabled mode the period between two consecutives beacons defines a superframe structure that is divided into 16 slots. Beacons always occupy the first slot, while the other slots are used for data communications. In order to support low-latency applications, the PAN coordinator can reserve one or more slots, designated by guaranteed time slots, avoiding the use of MAC mechanisms. In the beaconless mode, there is no superframe structure and no guaranteed time slots. As a consequence, only random access methods, such as unslotted CSMA/CA can be used to medium access. The frame length is limited to 127 bytes because unreliable and error prone wireless links are used and the devices have limited buffering capabilities.

B. Power line communication

The idea of using power lines, not only as electricity conductors, but also for communications purposes was proposed at the beginning of the last century [4]. This technology is known as power line communication (PLC). The main advantage is the wide spread availability of power distribution infrastructure and therefore the theoretical deployment costs are limited to connecting the communication devices to the existing electrical grid. Power line communication technologies can be grouped into narrow-band PLC (NB-PLC) and broadband PLC [7].

The narrow-band PLC systems operate in the frequency range between 3 kHz and 500 kHz, regulated by CENELEC, ARIB and FCC organizations. PRIME, G3-PLC and IEEE 1901.2 are examples of PLC narrow band standards. The PRIME was developed within PRIME alliance and uses up to 96 OFDM subcarriers over the frequencies from 42kHz to 89 kHz and it is able to achieve a bit rate of 128.6 Kbit/s. In order to deal with unpredictable impulsive noise, the PRIME standard includes an automatic repeat request (ARQ) mechanism, based on selective repeat retransmission. PRIME has de ability to form sub-networks, each one has one base node and several service nodes. The base node manages the sub-network's resources, such as the PLC channel access arbitration. The medium access control can be based on contention free and contention-based access mechanisms. The contention free mechanism relies on time division multiplex channel access period, where the base node assigns the channel to only one node at a time. The CSMA/CA it is used on contention-based. The G3-PLC can operate from 10 kHz to 490 kHz and it reaches peak bit rates up to 300 kbits/s. The MAC layer it is based on IEEE 802.15.4-2006 and therefore 6LoWPAN can be used to fulfil the IPv6 requirements. When PRIME and G3-PLC are compared, the first allows cheaper implementation because it is less complex, while the second it is more robust under interference conditions. The PRIME and G3-PLC form the IEEE 1901.2 protocol baseline.

The broadband power line communication (BB-PLC) is widely used for broadband networking applications, such as Internet access, gaming and high definition and 3D video. It operates in the band between 1MHz to 250MHz and having a bitrate ranging from several Mbps to several hundred Mbps. The BB-PLC is mainly based on four protocols: IEEE P1901, HomePlug, universal power line association and high definition power line communication [8]. The HomePlug alliance it is responsible for three protocols compatible with IEEE P1901. The HomePlug AV uses physical and MAC technology that provides 10Mbps for ROBO mode and up to 200Mbps on the adaptive bit-loading mode. On the physical layer, it operates in the frequency range of 2–28MHz, uses windowed orthogonal frequency division multiplexing and turbo convolutional code. On the MAC layer, HomePlug AV provides a quality of service connection oriented, contention-free service on a periodic Time division multiple access (TDMA) allocation, and a connectionless, prioritized contention-based service based on CSMA/CA). The HomePlug AV2 was developed to support high definition 3D and video while maintaining full compatibility with other HomePlug protocols. The HomePlug Green PHY it is similar to the other HomePlug protocols and was designed to support smart grid applications. Only ROBO mode and QPSK modulation it is supported and as a consequence it only support 4,5 and 10 Mbps. The HomePlug Green PHY MAC it is a simplified version of the HomePlug AV MAC. It shares the same CSMA and Priority Resolution mechanisms as HomePlug AV, but it does not support the optional TDMA mechanism.

C. 6LoWPAN

Low bandwidth, low-power resources and the maximum link-layer packet size of 127 bytes are the most relevant characteristics of the IEEE 802.15.4 standard. Implementing

standard IPv6 headers over LoWPAN would result in extremely small payloads for higher-level protocols, more over it requires a neighbor discovery protocol too verbosely based on multicast messages.

The IEEE 802.15.4 standard is mostly accepted as the physical and MAC layer protocol to provide wireless connectivity between IoT nodes. However the IPv6 protocol don't fully match with the IEEE 802.15.4 constraints. For example the minimum IPv6 MTU is 1500 bytes and the IEEE802.15.4 MTU is 127 bytes. Beside to this incompatibility, using standard IPv6 headers would result in extremely small payload for high protocols. To address these issues, the IETF 6LoWPAN-working group was formed to define the support of IPv6 over IEEE 802.15.4 networks. To support IPv6 over IEEE 802.15.4 an additional adaptation layer was introduced between data link and network layers. Like on the IPv6, the 6LoWPAN use stacked headers and therefore only the necessary header types are used. The 6LoWPAN standard defines four header types: i) the dispatch header, ii) the IPv6 header compression header, iii) the fragmentation header and iv) the mesh header. In the simplest case, only the dispatch and compression headers are used. At the beginning of each header, a header type field identifies the header format. Although the standard IPv6 neighbor discovery (ND) protocol should work on 6LoWPANs, the node's resource constraints, the absence of multicast support at layer two, the low duty-cycle and the use of non-transitive links requires a different approach for the ND protocol on 6LoWPANs focused on the efficient use of available energy.

Although 6LoWPAN was originally designed to support IPv6 over IEEE 802.15.4, it can later be adapted to be used on other similar link technologies. A typical 6LoWPAN network it is formed by nodes, routers and edge routers. 6LoWPAN nodes usually perform only sensing and actuation operations. They do not forward datagrams originated on other nodes and destined to other nodes. Routers are intermediate nodes that can be used to forward datagrams to others nodes or routers in the same LoWPAN and are present only in route-over topologies. Edge routers are used to connect the LoWPAN to others networks, for example, the Internet. Typically, nodes and routers have energy and computational resources constraints and only the edge routers are main powered and have more computational resources.

III. System Description

The demonstrator (Fig. 1) it is composed by: (i) IoT nodes (i.e. the environmental sensor, the panel module and the smart plug), (ii) a gateway to connect the IoT to the Internet and (iii) an application server and Internet connected clients.

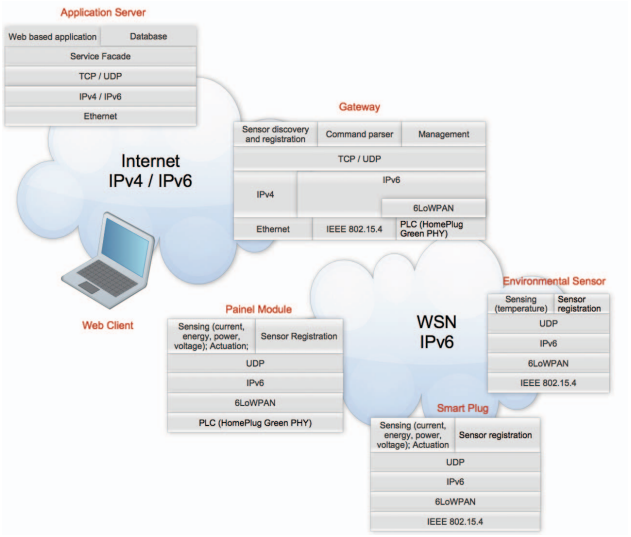


Fig. 1. Demonstrator architecture.

A. IoT nodes

Three different IoT nodes developed in house are used to measure energy, voltage and current and temperature (Fig. 2). Two of them are compatible with IEEE 802.15.4 and the other is compatible with PLC HomePlug Green PHY. All nodes are running Contiki 2.5 and RPL routing protocol [6]. An application compatible with Contiki 2.5 was developed to allow data retrieval, node management and actuation. The smart plug and the environmental sensor compliant with IEEE802.15.4 (Fig. 2) are based on Texas Instruments CC2530 MCU, which provides 8 KB RAM and 256 KB flash. The smart plug (Fig. 2, left) was developed to be inserted between the power outlet and the device to be controlled and it is equipped with voltage and current meters and with one relay that can be used to remotely switch on and off the electric device connected to the power outlet. The environmental sensor (Fig. 2, right) it is equipped with temperature and light transducers and can be used to control home environmental parameters.

The panel module PLC HomePlug Green PHY (Fig. 2, center) compatible node it is based on Texas Instruments MCU MSP 430, which provides 16KB RAM and 128KB flash. The PLC interface is provided by Qualcom QCA 7000 chipset. This node was developed to measure and control more than one device connected to the same distribution circuit and therefore it should be installed near to the utility grid global energy consumption meter. It is equipped with three voltage and current meters, one for each circuit, and with two relays that can be used to control two different circuits.

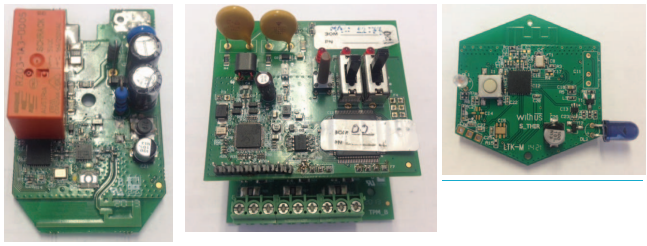


Fig. 2. IEEE 802.15.4 smart plug (left), PLC nodes (center) and IEEE 802.15.4 temperature sensor (right).

B. Gateway

The gateway was also developed in house and uses a LPC 3240 CPU with 64MB RAM and 256MB capacity flash NAND (Fig. 3). It provides four interfaces compatible with Fast Ethernet, IEEE 802.11 g/n, HomePlug Green PHY PLC and IEEE 802.15.4. The Qualcom QCA 7000 and the Texas Instruments chips CC2530 are used to implement, respectively, the PLC and the IEEE 802.15.4 interfaces. The gateway is running embedded Linux and supports both IPv4 and IPv6 stacks. At the application layer three application modules were developed: i) the sensor discovery and registration, ii) the command parser and iii) the management. The sensor discovery and registration it is based on 6LoWPAN neighbor discovery and RPL messages and it is used to maintain the list of the available IoT devices updated and to provide seamless connectivity. The command parser allows the interaction between the application server and the IoT devices. All the requests from the application server are validated before being sent to the IoT nodes. Two main reasons can be evoked to avoid end-to-end connectivity between the IoT network and the Internet connected devices. First, end-to-end connectivity exposes the IoT network to several remote security attacks. Second, IoT devices should be accessed based on its name, location and supported functionalities, therefore only the data and its context is important for the end users not the IoT device IP address. The command parser also translates the IPv6 requests into IPv4 and vice-versa, if needed.

Fig. 3. Gateway PCB board.

The management module it used to provide node's resource description (such as the available transducers and actuators).

C. Application server and Internet Clients

The Internet clients interact with the application server retrieve data from the IoT nodes and also perform actuation, such as relay switching The application server runs on a desktop PC with Ubuntu 10.0.4 LTS operating system and provides a REST API to be used by the Android APP, running on the smart phone clients. It also provides a webserver to be used by a laptop client and a MySQL database to store the data retrieved from the LLN nodes. The application server also runs an application that interacts with the gateway in order to retrieve data and information from IoT nodes.

IV. Demonstration and Validation

The following validation experiments were successfully performed over the network presented in the Fig. 4:

- Node discovery and resource discovery mechanism: (i) a new node is connected to the network, (ii) the gateway detects the new node, based on the RPL data structures, and requests its supported functionalities and (iii) the directory service is updated with the new node information.
- Data retrieval: the client application selects, from the list with all available IoT nodes, the one from which it wants to retrieve data; the application server retrieves periodically data from all available IoT nodes.
- Actuation and node configuration: the actuation orders can be performed from the client application and from other IoT nodes. For example, the environmental sensors are configured to actuate over other IoT node's relay if the temperature raise a preconfigured value.

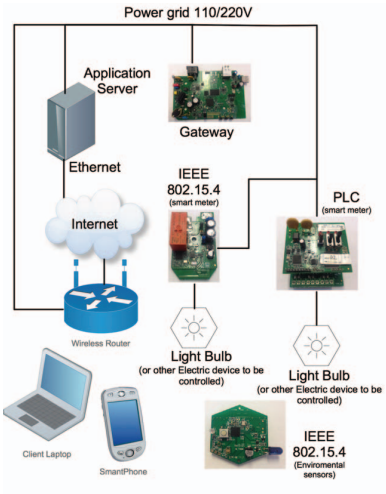


Fig. 4. Gateway PCB board.

V. Conclusion

This paper presents an IoT solution based on IEEE 802.15.4 and PLC HomePlug Green PHY for home energy monitoring. The 6LoWPAN was used as unifying layer to provide connectivity between all IoT devices. The proposed solution includes node and resource discovery mechanisms to provide seamless connectivity and data retrieval and actuation based on IoT device name, location and supported functionalities. A demonstrator was built to validate and evaluate the proposed solution. In this demonstrator the IoT devices were built in house and based on COTS chipsets.

All the proposed mechanisms are based on standards and only open source and freeware software was used on its implementation. As a future work, the proposed solution may be extended to include security and semantic interoperability mechanisms.

ACKNOWLEDGMENTS

This work has been partially supported by the *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal, by the Visiting Professor Program at King Saud University, by National Funding from the FCT - *Fundação para a Ciência e a Tecnologia* through the UID/EEA/50008/2013 Project and by the LOPIX Project financed by QREN funding program and by Withus Innovation and Technology Company.

REFERENCES

[1] IEEE Std 802.15.4-2006. Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs). IEEE Std. 802.15.4-2006, 2006.

[2] IEEE Std P1901.2, Low-Frequency Narrow-Band Power Line Communications Working Group, IEEE P1901.2 Working Group Status Update. IEEE Std P1901.2-2013, 2013,

[3] H. Han-Chuan, L. Chi-Ha, “Internet of Things Architecture Based on Integrated PLC and 3G Communication Networks," 2013 International Conference on Parallel and Distributed Systems, pp. 853-856, 2011 IEEE 17th International Conference on Parallel and Distributed Systems, 2011.

[4] J. Hui, D. Culler, “Extending IP to Low-Power, Wireless Personal Area Networks,” IEEE Internet Computing, vol. 12, no. 4, 2008, pp. 37-45.

[5] N. Kushalnagar, G. Montenegro, C. Schumacher, “IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals,” Internet Engineering Task Force, Request for comments 4919, August 2007.

[6] T. Winter, P. Thubert, A. Brandt et al., “RPL: IPv6 routing protocol for low-power and lossynetworks,” RFC 6550, March 2012.

[7] L. T. Berger, A. Schwager, P. Pagani, and D. Schneider, MIMO Power Line Communications: Narrow and Broadband Standards, EMC, and Advanced Processing, CRC Press, 2013.

[8] S. Galli, A. Scaglione, Z. Wang, "For the Grid and Through the Grid: The Role of Power Line Communications in the Smart Grid," Proceedings of the IEEE - Special Issue on Smart Grid, vol. 99, no. 6, June 2011.

Anexo B:

This appendix is based on the following paper:

A WSN Solution for Light Aircraft Pilot Health Monitoring

L. M. L. Oliveira, J. J. P. C. Rodrigues, B. M. Mação, P. A. Nicolau and L. Zhou, 2012 IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, 2012, pp. 119-124.

DOI: 10.1109/WCNC.2012.6213959

2012 IEEE Wireless Communications and Networking Conference: PHY and Fundamentals

A WSN Solution for Light Aircraft Pilot Health Monitoring

Luís M. L. Oliveira [1, 2], Joel J. P. C. Rodrigues [1], Bruno M. Mação [2], Paulo A. Nicolau [2], and Liang Zhou [3]
[1] Instituto de Telecomunicações, University of Beira Interior, Covilhã, Portugal
[2] Polytechnic Institute of Tomar, Tomar, Portugal
[3] Nanjing University of Posts and Telecommunications, China

loliveira@it.ubi.pt; joeljr@ieee.org; estt8060@ipt.pt; estt11882@ipt.pt; liang.zhou@ieee.org

***Abstract*— Wireless sensor networks can be used to improve both safety critical and unsafety critical aircrafts systems. Using wireless sensor networks can help to increase the number of sensors as well the system redundancy and also helps to reduce the aircraft system weight and complexity, improving the fuel efficiency and maintenance costs. Supporting standard protocols in all wireless sensor nodes simplifies the application development, configuration and maintenance. The wireless sensor network devices can also be used to monitor the physiological pilot's parameters. This paper presents a complete and innovator solution, mainly based on standard protocols, to monitor light aircraft and gliders pilot's physiologic parameters. The proposed system does not interfere with pilot's agility, is simple to install, configure and operate. To evaluate the system, a real testbed was deployed.**

***Keywords*- Wireless sensor networks; All-IP networks; 6LoWPAN; IEEE 802.15.4; WSN and aircraft**

I. INTRODUCTION

Sensor devices are finding applications in many areas, such as medical and environmental monitoring, industrial automation, smart grids, smart cities and urban networks, home and building automation, structural health monitoring, military, automotive and aeronautic applications [1-2]. Recent innovation in micro-electro-mechanical systems (MEMS) and in low-power wireless network technology have created the technical conditions to build multi-functional tiny sensor devices, which can be used to remotely observe and collect data related to physical phenomena of their surrounding environment [1]. Wireless sensor nodes are low-power devices equipped with processor, storage, a power supply, a transceiver, one or more sensors and, in some cases, with actuators. Several types of sensors can be attached to wireless sensor nodes, such as, optical, chemical, thermal, motion and biological. These wireless sensor devices are small and, usually they are cheaper than the regular sensor devices.

The wireless sensor devices can automatically organize themselves to form an ad-hoc and multi-hop network. Wireless sensor networks (WSNs), may comprise hundreds or maybe thousands of wireless low-power sensor node devices, working together to realize a common task. Wireless support, energy and resource constrain, self-organizing, self-optimizing and fault-tolerant are the main characteristics of this type of network [1].

Nowadays, several technologies can be used to support wireless sensor networks, most of them have been specified using standard IEEE 802.15.4 [3] as a physical and link layer technology. However, some network and upper layer protocols are proprietary, such as ZigBee [4] and WirelessHART [5]. Moreover, these proprietary protocols are incompatible with IP, and therefore complex gateways are required to connect these networks to the Internet and new methodologies are necessary to develop node's software.

A new paradigm was needed to enable wireless sensor network devices to accessed from the Internet, independently of the used physical and MAC layers protocols. The application developing process is also simplified and open and there are tools already developed for commissioning, configuring, managing, and debugging can be used or adapted. Originally, the scientific community not considered appropriate the use IP suite protocol in small power and resource-constrained networks, because of the perception that is was to heavy weight. Recently, the industry and the scientific community started to rethink many misconceptions about the use of IP protocol suite in energy and resource constrained nodes [6].

A typical commercial/military aircraft are supported by several safety-critical systems, such as aircraft engine control system, aircraft flight control systems and also by nonsafety critical systems, such as structural and engine health monitoring systems, aircraft cabin environmental control system [7]. Wired connections are mainly used to support the communications between the sensors devices and the control management unit. Using wireless sensor devices can help to increase the number of sensors as well the system redundancy [8]. It also helps to reduce the aircraft system weight and complexity, improving the fuel efficiency and maintenance costs [9]. Civilian small dimension aircrafts such as light aircrafts and gliders normally have much simpler safety support systems.

Several physiological effects influence the pilot's performance and therefore the flight safety and low blood

978-1-4673-0437-5/12/$31.00 ©2012 IEEE

oxygen (hypoxia) is perhaps the one with highest negative impact. Moreover, most of these pilots do not know their own physiological limitations to flight and to altitude, not noticing the symptoms of hypoxia when they reach or remain in these environmental conditions. Note that, in most of the light aircrafts the cabin is unpressurized and the air quality is not monitored. In such conditions, besides the low level altitude, those pilots are more exposed to the hypoxia situations than the commercial/military pilots.

This paper presents a complete solution to monitor the pilots' physiological parameters, such as the blood oxygen saturation and body temperature. Environmental parameters are also monitored, for example air temperature and humidity, altitude, air pressure, 3D acceleration and GPS location. Except the blood oxygen sensor, all the others are supported by inexpensive hardware. The acquired data is stored in a database system and can be easily exported to *csv* (comma separated value) file and correlated. Altitude and blood oxygen saturation values are accessible to pilots in real time. A real testbed has been constructed to evaluate the proposed solution and to prove their capabilities.

The rest of this paper is organized as follows. Section II elaborates on the related work, while Section III focuses on the system architecture. Section IV presents the system evaluation and demonstration using a real testbed. Finally, Section V concludes the paper and pinpoints future research work.

II. RELATED TECHONOLOGIES

Solutions to monitor the aircraft systems and structures using wireless sensor networks have already been proposed [7-9]. However, none of the proposed solutions allow the pilot's health monitoring. Besides, the current solutions use proprietary protocols and do not support IP suite protocols in all wireless sensor nodes. The use of standard protocols, such IP suite protocols, reduces the complexity to connect wireless sensor networks to the Internet and simplifies the application developing process.

A. IEEE 802.15.4

The standard IEEE 802.15.4 [3] released in 2003 was the first low-power layer two standard for low power wireless personal area network. IEEE 802.15.4 physical layer provides an interface between the medium access control (MAC) sub-layer and the physical radio channel. Two services are provided, the physical data service and the physical management service. The physical layer is responsible for the activation and deactivation of the radio transceiver, energy detection sensed on the current channel, clear channel assessment for CSMA/CA, channel frequency selection, link quality indication (LQI) for received packets and data transmission and reception. The IEEE 802.15.4 defines three physical operation modes over 27 different channels: 20 kbps at 868 MHz, 40 kbps at 915 MHz, and 250 kbps at 2.4 GHz using direct spread spectrum modulation. A device can use either a 64-bit address or a 16-bit short IEEE address in a IEEE 802.15.4 network. The frame length is limited to 127 bytes because low-power wireless links are used and the sensors have limited buffering capabilities.

Two types of devices were defined, full-function devices (FFD) and reduced-function devices (RFD). In FFD all network functionalities are supported and therefore it can be used in peer-to-peer and multi-hop communications and to measure physical parameters. The reduced-function devices only support a limited set of functionalities and they can only be used to measure physical parameters, to execute basic tasks and do not support multi-hop communications. The FFD and RFD devices organize themselves in a personal area network (PAN) controlled by a PAN coordinator, which has the function of setting up and maintaining the network.

The MAC sub-layer specifies an interface between the service specific convergence sub-layer and the physical layer. Similar to the physical layer, the MAC sub-layer also provides two services, the MAC data service and the MAC management service. The MAC sub-layer is responsible for PAN node association and disassociation, to transmit network beacons if the device is a PAN coordinator; to synchronize to the beacons, to execute CSMA/CA mechanism for channel access, to support the guaranteed time slot mechanism and to provide a reliable link between two nodes.

The IEEE 802.15.4 MAC specifies two operation modes, the synchronous beacon enabled and the asynchronous beaconless. In the beacon-enabled mode, the PAN coordinator periodically broadcasts beacons containing information about the PAN. In the beacon-enabled mode the superframe structure is used and nodes do not need to use contention mechanisms before transmit because one or more time slots can be previously reserved. Synchronization provided by the beacons allows devices to sleep between transmissions, which result in energy efficiency and extended battery lifetime. The beaconless mode does not permit superframe structures; as a consequence guaranteed time slots cannot be reserved. So, only random access methods such as unslotted CSMA/CA can be used to medium access.

Two network topologies can be adopted, the star and the peer-to-peer topology. In a star topology a master-slave network model is used. An FFD device assumes the PAN coordinator role and controls all the networks operations and the other nodes can only with PAN coordinator. Star topology is better suited for small networks. In peer-to-peer topology multi-hop communications can be used to send messages to other nodes outside of its radio range. Peer-to-peer topology supports more topologies, such as mesh or hierarchical cluster. When compared to star topology, peer-to-peer have the capability to provide extension of network coverage without increasing transmit power or receive sensitivity, better reliability via route redundancy, easier network configuration and better device battery life.

B. *6LoWPAN* adaptation layer

The network layer protocol must fulfill the constraints imposed by the IEEE 802.15.4 protocol. However, the standard IPv6 protocol does not fully match with such constraints. For example, the smallest allowed maximum transmission unit (MTU) for an IPv6 packet is 1280 bytes. However, the frame size provided by IEEE 802.15.4 is limited to 127 bytes, of which only between 81 and 102 bytes are available for payload considering link layer overhead. To address these issues the 6LoWPAN-working group defined an additional adaptation layer to be introduced between data link and network layers.

Three main services are provided by the 6LoWPAN adaptation layer [10], packet fragmentation and reassembly, header compression and link layer forwarding when routing decisions are taken at layer two in a multihop network. The 6LoWPAN currently supports three headers types, the fragmentation header, the compression header and a mesh-addressing header. Like IPv6 the 6LoWPAN also uses encapsulation header stack, so only the required headers are used. The 6LoWPAN defines a stateless compression scheme consisting of two parts: the header compression one (HC1) and the header compression two (HC2). HC1 allows compressing the IPv6 header with an original size of 40 bytes into three bytes in the best case. Analogously, the HC2 describes a compression format to reduce the length of the TCP/UDP headers. 6LoWPAN uses the fragmentation header to support the minimum value required by IPv6 for the underlying MTU (which is 1280 bytes). Whenever the payload is too large to fit into a single IEEE 802.15.4 frame, it is fragmented into several packets and the fragmentation header is added to the header stack.

Two RFCs were released, the RFC 4919 [10] and the RFC 4944 [11]. The first document describes the assumptions, problem statement and goals of 6LoWPAN. The second describes the frame format for transmission of IPv6 packets, the method for defined IPv6 link-local addresses and stateless auto configured addresses, an header compression scheme using shared context and the frame delivery process in a link-layer multihop network.

The Neighbor Discovery (ND) is used to discover the neighbor nodes, maintain the reachability information, to do prefix discovery and default routing configuration in a similar way as in IPv6. This protocol also performs address resolution, neighbor unreachable detection and duplicated address detection. The regular IPv6 ND protocol can be used on 6LoWPAN networks with few modifications. However, there are significant challenges to use the current IPv6 ND within LoWPANs. Firstly, ND uses link-local multicast for sending address resolution solicitations, router advertisements and duplicated address detection messages, currently LoWPAN does not have support for multicast communications due to energy conservation. Secondly, IPv6 ND was not designed for non-transitive wireless links. Finally, ND protocol is too verbose and may generate an overhead in the number of transmitted messages. So, the actual protocol is not perfectly adjusted to the 6LoWPAN requirements and may generate an overhead in the number of messages. A new ND protocol is under discussion in IETF 6LoWPAN working group.

C. *Application to WSNs for aircraft systems*

Advances in global position system (GPS), sensor devices and wired and wireless networking have revolutionized the aviation industry [7]. These advance promises to improve the safety, efficiency, transportation capacity and environmental footprint of air transportation.

Current safety-critical systems are supported on wired connections, which are complex, difficult to route, heavy and prone to damage due to wear. Moreover, the wired based data acquisition systems are complex to install, reconfigure and maintain. Also, there are some inaccessible locations and harsh environments that impose restrictions on the use of wired connections. Avionics Full-Duplex Switched Ethernet standard was proposed to reduce the number and the complexity of wired connections [12]. The replacement of current wired based connections with a wireless sensor network, can simultaneously help to increase the number of sensors and their locations, as well as increasing the system redundancy. Moreover, it also offers significant benefits in flexibility, interoperability, weight reduction, reduction in direct and maintenance costs and robustness improvement.

WSN are currently use in inland operations, such as structural and systems monitoring [8]. Before to use wireless sensor networks on safety critical systems during flight it is necessary to ensure that their operability will not be compromised due to interferences phenomena. Also, wireless sensor networks should not interfere with others wireless systems on board, such as aircraft radio communications. The effect of crew and passenger wireless equipment on wireless sensor network operation should be also considered. A very high degree of safety assurance and certification is required before using wireless communications networks for safety critical functions. Several wireless radio frequencies, typically operate in unlicensed spectrum, were already certified to be used on nonsafety critical systems [9]. A specific regulation applied to aircraft wireless systems for safety critical functions does not exist. Besides, research needs to be conducted to study the WSN operation in both safety and unsafety critical functions.

III. System Architecture

The proposed system aims to monitor physical environmental and pilot's physiologic parameters. The measured environmental parameters are: air temperature, atmospheric pressure, air humidity and light intensity in both interior and exterior aircraft. GPS in 3D axis and acceleration in 2D axis are also recorded. Cerebral peripheral oxymeter, ImG and corporal temperature are the physiologic parameters measured. Three different communication interfaces are used to retrieve data from sensors. To sense environmental parameters and pilot's corporal temperature IEEE 802.15.4

sensors are used. The GPS data is retrieved by USB serial data. Finally, Bluetooth interface is used to support communication with oxymeter device. The system architecture can be divided in three different modules, according to communication interfaces used to retrieve data from the sensor devices (figure 1-3). A laptop computer is used to execute the application and to store the measured data.

A Bluetooth sniffer captures the data transmitted by the oxymeter via Bluetooth interface and a text file with the oxymeter sensor readings is created with all recorded values. Next, a temporary file is created to store the last read value. The temporary file is necessary to make the system independent from the periodicity used by the oxymeter sensor and to faster the real time presentation and alert generation related to the oxymetry values. A time stamp is added to the oxymetry last read value. In the next step, two operations are executed, first the value is presented to the screen installed in from of the pilot and an alert is generated if the value is bellow the limit. Second, the value is stored on a buffer and next is copied to a final *csv* file before being stored on the database. The *csv* file is used to make the application more robust.

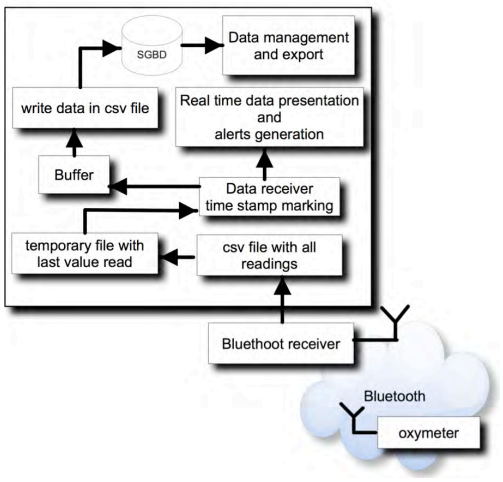


Figure 1. Bluetooth communication architecture module.

To retrieve data from GPS device, a request message is sent to the device using a USB interface. The response is processed and stored in a text file. As previous module, the application transforms the data in to a *csv* standard file (Figure 2). The next steps are similar to the previous module.

Finally, to read the data from IEEE 802.15.4 compliant sensors, IPv6 end-to-end connectivity is used based on client server architecture. The server application is installed on the mote sensors and the client is the main application. A 6LoWPAN gateway is needed to grant the communication between the main application, stored on an IPv6 regular node, and 6LoWPAN nodes (Figure 3). The main application is responsible for data request, data processing and presentation

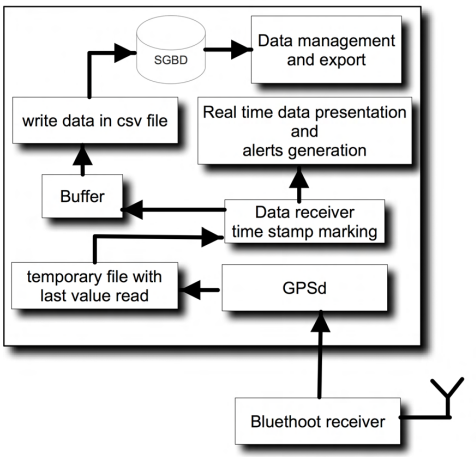


Figure 2. GPS communication architecture module.

and persistent data storage. To achieve this functions the following tasks must be completed: user authentication and validation, sensor connectivity tests, sensor data retrieve and timestamp marking, real-time data presentation and alerts, CSV dumping into database and database queries. To prevent data loss, multithreading architecture and buffer structures were used. A single thread were used to support each reading and writing operation. The reading threads are responsible by gathering the data from de sensor devices, adding the timestamp, publish the real-time data and fill individual buffers. Each writing threads, remove the data from the correspondent buffer and append the data to the sensor individual CSV standard file. The dumper operation is executed when the flight is over and the data acquisition ended.

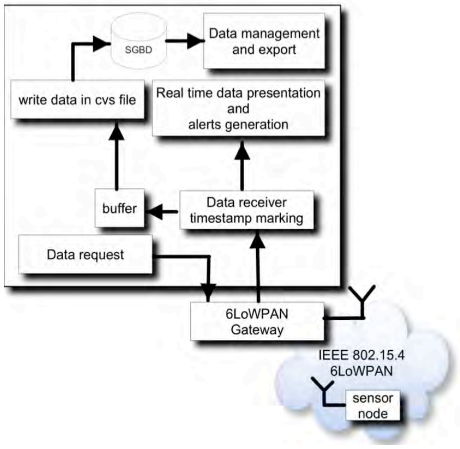


Figure 3. IEEE 802.15.4 communication architecture module.

IV. System Evalutation and Demonstration

In order to evaluate the performance and demonstrate the operation of the proposed system a real testbed has been deployed (Figure 4). This section presents the testbed deployment details and the results obtained.

Three Iris sensors with MTS400 expansion board were used to measure air and body temperature, humidity, atmospheric pressure and 2D acceleration. The sensors use TinyOS 2.1.1 and Blip 1.0 with IPv6 addresses manually assigned. The 6LoWPAN gateway [13] is used to send the requests and to retrieve the data from the Iris sensors and it is supported by the TinyOS IP-driver application compliant with RFC 4944.

Nonin 7600 [14] monitor with two channels were used to measure the peripheral oxygen saturation in the pilot's head. The Bluetooth sniffer minicom installed in the Ubuntu operating system is used to capture the oximetry values and to write it in a *csv* temporary file. The Garmin GPS18 USB was used to record the aircraft's position and the altitude. The GPSd daemon is used to receive, to format and to write in a *csv* text file the GPS data.

Figure 4. Testbed equipment inside the light aircraft.

The main application was developed using java programming language and uses three interfaces, one for each type of communication interface, to request and retrieve data from the measure devices. To generate the timestamp value two java JDK classes were used, the calendar and java.sql.timestamp. During the flight oximetry and altitude data is shown in real time (Figure 5). In fact only these two values were considered critical to the pilot's safety. When the new data is processed by the application the timestamp functions is executed to ensure the data time integrity and the right format to insert into database. At the end of each flight, the dumper read the *csv* files verifies the data integrity and uses java.SQL class to insert the values in the data model supported by MySQL database management system. The *csv* files are only destroyed when dumper module finishes without errors. To execute the software modules and to store the read values a dual core MacBook with an SSD running Ubuntu 10.0.4 LTS edition was used. The main application is also responsible for sensor connectivity verification and database management and export to a *csv* file.

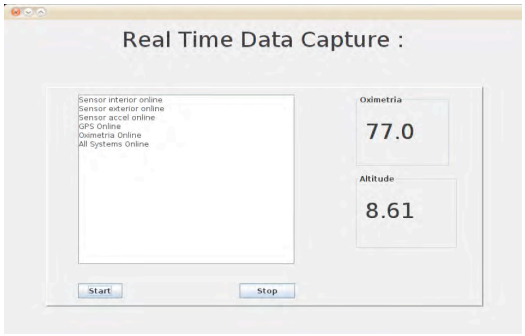


Figure 5. Real time data capture application user's interface.

During the flights no RF interferences were detected in the proposed monitoring system or in the aircraft communications systems.

V. Conclusion and Future Work

Wireless sensor networks are potential driver to improve both safety critical and unsafety critical systems. However future research need to be conducted to study WSN operation in such conditions. Also, new wireless aircraft certification regulations need to be developed to address the various security and safety threats. Supporting standard protocols, such as IP suite protocols in all WSN nodes permits simultaneously facilitate the application development and their connection to the Internet. The system proposed in the current paper presents a complete and innovator solution, mainly based on standard protocols, to monitor light aircraft and gliders pilot's physiologic parameters. The system is simple to use and to setup and the results obtained are promising. Three different communication interfaces were used to interact with measure devices, because GPS and oximetry device does not support 6LoWPAN. Based on the obtained results, new research can be made to evaluate pilots' cognitive function during flight.

As future work, it is intended to measure new physiologic parameters such as, electroencephalogram with auditory evoked potentials. Embed the sensors in the pilot's flying suit, support IP protocol suit in all sensors and study the radio performance interference in commercial aircrafts it is also addressed as future work.

Acknowledgments

This work has been partially supported by the *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal, Nonin Medical Inc., and by National Funding from the FCT – *Fundação para a Ciência e a Tecnologia* through the PEst-OE/EEI/LA0008/2011 Project.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, "Wireless sensor networks: A survey," Computer Networks, vol. 38, no. 4, 2002, pp. 393 – 422.

[2] K. Sohraby, D. Minoli, T. Znati, "Wireless Sensor Networks: Technology," Protocols and Applications, Hoboken: Wiley-Interscience, 2007.

[3] IEEE Std 802.15.4-2006. Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifica-tions for low-rate wireless personal area networks (LR-WPANs). IEEE Std. 802.15.4-2006, 2006.

[4] ZigBee Alliance, "ZigBee specification: ZigBee document 053474r013 Version 1.1," ZigBee Aliance, December. 2006, http://www.zigbee.org, accessed in January 2012.

[5] J. Song, H. Song, A. Mok, D. Chen, M. Lucas, M. Nixon, W. Pratt, "WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control," Proceedings of the 2008 IEEE Real-Time and Embedded Technology and Applications Symposium, St. Louis, United States, 2008, pp. 377-386.

[6] J. Hui, D. Culler, "Extending IP to Low-Power, Wireless Personal Area Networks," IEEE Internet Computing, vol. 12, no. 4, 2008, pp. 37-45.

[7] W. Wilson, G. Atkinson, "Wireless sensing opportunities for aerospace applications," Sensors and Transducers Journal, vol. 94, no. 7, 2008, pp. 83 – 90.

[8] D. Goldsmith, E. Gaura, J. Brusey, "Wireless sensor networks for aerospace applications-thermal monitoring for a gas turbine engine," Proceedings of Nanotech Conference and Expo. Boca Raton, FL: CRC Press-Taylor & Francis Group, 2009, pp. 507 – 512.

[9] H. Bai, M. Atiquzzaman, D. Lilja, "Wireless sensor network for aircraft health monitoring," Broadband Networks, 2004. BroadNets 2004. Proceedings. First International Conference on, 25-29 Oct. 2004, pp 748-750, doi: 10.1109/BROADNETS.2004.92.

[10] N. Kushalnagar, G. Montenegro, C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," Internet Engineering Task Force, Request for comments 4919, August 2007.

[11] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," Internet Engineering Task Force, Request for comments 4944, September 2007.

[12] W. Zhitao, L. Tangqi, W. Xuewang, N. Huang, "The buffer size assignment of AFDX based on network calculus," Reliability, Maintainability and Safety (ICRMS), 2011 9th International Conference on, 12-15 June 2011, pp.1319-1323, doi: 10.1109/ICRMS.2011.5979474.

[13] L.M.L. Oliveira, J.J.P.C. Rodrigues, B.M. Macao, P.A. Nicolau, Lei Wang; Lei Shu, "End-to-end connectivity IPv6 over wireless sensor networks," Ubiquitous and Future Networks (ICUFN), 2011 Third International Conference on, 15-17 June 2011, pp. 1-6 doi: 10.1109/ICUFN.2011.5949126

[14] Nonin 7600 cerebral Oximetry, URL: http://www.nonin.com/Model7600. Accessed in January 2012.

Anexo C:

This appendix is based on the following paper:

End-to-end connectivity IPv6 over wireless sensor networks

L. M. L. Oliveira, J. J. P. C. Rodrigues, B. M. Mação, P. A. Nicolau, Lei Wang and Lei Shu,

2011 Third International Conference on Ubiquitous and Future Networks (ICUFN), Dalian, 2011, pp. 1-6.

DOI: 10.1109/ICUFN.2011.5949126

End-to-End Connectivity IPv6 over Wireless Sensor Networks

Luís M. L. Oliveira [1, 2], Joel J. P. C. Rodrigues [1], Bruno M. Mação [2], Paulo A. Nicolau [2], Lei (Ray) Wang [3], and Lei Shu [4]

[1]Instituto de Telecomunicações, University of Beira Interior, Covilhã, Portugal

[2]Polytechnical Institute of Tomar, Tomar, Portugal

[3]Dalian University of Technology, China

[4]Department of Multimedia Engineering, Osaka University, Japan

loliveira@it.ubi.pt; joeljr@ieee.org; {estt8060; estt11882}@ipt.pt; {lei.wang; lei.shu}@ieee.org

***Abstract*— It is foreseeable that in the near future any object will have an Internet connection – this is the Internet of Things vision. All these objects will be able to exchange and process information and all information will be accessible from Internet. Some of these objects, with wireless support, are characterized by small size, power constrains and small computing resources. Connecting such devices to the Internet is considered simultaneously the biggest challenge and a great opportunity for the Internet grow. To achieve the Internet of things vision is necessary to support IPv6 protocol suite in all objects. Supporting IPv6 simplifies the task of connecting objects to the Internet and the application developing process is also simplified and open. However, support IP suite protocol in low power and computing resource-constrained objects is an important challenge. This paper presents a complete solution to connect the smart objects to the Internet. Security measures were also one of the concerns in this solution. The paper also presents the design and deployment of a laboratory testbed to prove the proposed solution operation.**

***Keywords*— Internet of things; WSN; Wireless Sensor Networks; IPv6; 6LoWPAN; Testbed.**

I. INTRODUCTION

During the 20th century two important digital revolutions happened. First, the use of computers is widespread and their use is fundamental in all quotidian life aspects. Second, the Internet interconnected the computers, changing how people work, think, and interact with each other's. Connecting smart objects to the Internet will be on of the biggest digital revolution in 21th century [1]. The smart objects are characterized by small size, power constrained, small computing and storage resources and, some of them, with reduced radio ranges and throughput, also designated as low power over wireless personal area networks (LoWPAN). Wireless sensor network (WSN) is a subtype of smart objects, where the devices can interact with their environment by sensing and/or controlling some physical parameters. The low size, the low cost, and the wireless communication and sensing capabilities make these devices appropriate for monitoring purposes. A smart network may be comprised by hundreds, or maybe thousands, of smart objects working together to accomplish a common task. Nowadays, there is a tendency to transform several quotidian objects in smart objects, realizing a vision of ambient networks where many different devices will collect and process information from many different sources to both control physical processes and interact with human users. Self-organizing, fault-tolerance, and self-optimizing are the main characteristics of such networks [2].

In a near future, users can access the information collected by smart objects from the Internet, using regular devices and applications. Nowadays, there are several technologies that can be used to realize the Internet of Things vision [1]. Most of the solutions have been specified using the standard IEEE 802.15.4 [3] as link layer technology and some of them proprietary, such as ZigBee [4] and WirelessHART [5]. Moreover, these solutions are not compatible with IP protocol and, therefore, complex gateway systems are required to connect the ZigBee networks to the Internet. Such gateways are hard to manage because updates are required whenever new functionalities are introduced.

A new paradigm was needed to enable smart objects to be accessed from the Internet where all the embedded devices and networks are natively IP-enabled and Internet connected, independently of the used physical and MAC layers protocols. The application developing process is also simplified, and open and tools already developed for commissioning, configuring, managing, and debugging can be used or adapted. Originally, the scientific community not considered appropriate the use IP suite protocol in small power and resource-constrained networks, because of the perception that is was to heavy weight. Recently, the industry and the scientific community start to rethink many misconceptions about the use of IP in all nodes [6]. Support TCP/IP stack in all devices is crucial to realize the Internet of Things vision.

The growth of Internet of Things is hard to estimate for two main reasons: firstly, embedded systems are expected to have significant impact in several military and civil applications and, secondly, the growth is not directly dependent of human users increase. So, the Internet of Things is considered simultaneously a biggest challenge and an enormous opportunity for the Internet grow [1]. The IPv6 have enough address space to connect all smart devices, however it has not designed to be used in low-power and resource constrained objects. To address these constrains, 6LoWPAN adaptation layer was defined to be used between data link layer and network layer [7].

Auto-configuration of network nodes is a required feature in wireless sensor network when managing large networks. Stateless address auto-configuration mechanisms provided by the IPv6 neighbor discovery protocol are used in the node configuration. When started up, a node sends a router solicitation request and the router will respond with a router advertisement message including configuration parameters for the current network (i.e. the network prefix parameters and the router link local address). This paper presents a solution to connect smart objects to the Internet, using end-to-end IPv6 connections. The solution uses a gateway that acts as a router; additionally it performs traffic filtering (i.e. only relevant traffic is forwarded form the Internet to the LoWPAN), IP and UDP header compression and packet fragmentation and reassembly. A laboratory testbed has been constructed to evaluate the proposed solution and to prove their capabilities.

The remainder of this paper is organized as follows. Section II presents the related work, while Section III focuses on the system architecture. Section IV presents the system evaluation and demonstration using a testbed. Finally, Section V concludes the paper and pinpoints future research work.

II. Related Work

Solutions to connect smart objects to the Internet have already been proposed [7][8][9], however such solution does not allow IPv6 end-to-end connectivity between the Internet and the smart objects. In such solutions, a proxy is used to provide to remotely configure and retrieve data from smart objects. The use of standard protocols, such IP suite protocols, reduces the complexity to connect smart object network to the Internet and simplifies the application developing process.

A. 6LoWPAN

The IEEE 802.15.4 protocol is widely accepted as the PHY and MAC layer protocol for LoWPANs. Nevertheless, the network layer protocol must comply with the constraints imposed by the IEEE 802.15.4 protocol. The properties of the standard IPv6 protocol do not fully match with such constraints. To address this issue, the IETF created the 6LoWPAN-working group to define the support of IPv6 over IEEE 802.15.4 LoWPAN networks. To support IPv6 over IEEE 802.15.4 an additional adaptation layer was introduced between data link and network layers, as specified in Figure 1.

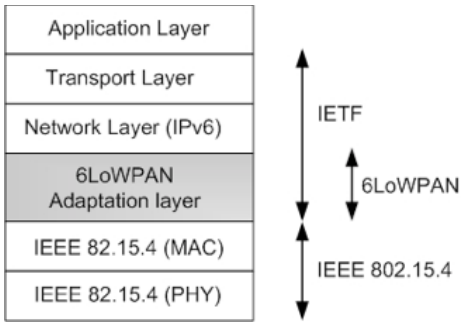


Figure 1 - 6LoWPAN layered architecture.

The 6LoWPAN working group focused on the following items: *i*) to define limited extensions to IPv6 neighbor discovery protocol [10] tailored for low-power networks; *ii*) to describe mechanisms allowing compression of 6LoWPAN headers, to reduce the header overhead; *iii*) to define the "6LoWPAN Architecture" describing the design and implementation of 6LoWPAN networks; *iv*) to define 6LoWPAN routing requirements describing 6LoWPAN specific requirements on routing protocols used in 6LoWPANs; *v*) to produce use cases for 6LoWPAN defining, for a small set of applications with sufficiently unique requirements, how 6LoWPANs can solve those requirements, and which protocols and configuration variants can be used for these scenarios; and *vi*) to define the threat model of 6LoWPANs documenting security mechanisms issues.

Two RFCs were released, the RFC 4919 [7] and the RFC 4944 [30]. The first document describes the assumptions, problem statement, and goals of 6LoWPAN. The second describes *i*) the frame format for transmission of IPv6 packets, *ii*) the method for defining IPv6 link-local addresses and stateless auto configured addresses, *iii*) an header compression scheme using shared context, and *iv*) the frame delivery process in a link-layer in IEEE 802.15.4 mesh network. Compression mechanisms for IPv6 datagrams in 6LoWPAN networks, design and applications spaces for 6LoWPANs, 6LowPAN neighbor discovery protocol and problem statement and requirements for 6LoWPAN routing are under open discussion.

B. Internet of Things approaches

The term Internet of Things [1] describes a vision in which networks and embedded devices are omnipresent in our lives and provide relevant content and information whatever the location of the user. Sensors and actuators will play a relevant role in realize this vision.

There are several trends to take into consideration when thinking about the Internet of Things paradigm. These include ZigBee, machine-to-machine communications, the Future Internet, Web of Things, and wireless sensor networks.

As above-mentioned, Zigbee is an industry protocol specification made by ZigBee Alliance. ZigBee starts in 2003 in conjunction with IEEE802.15.4, and specifies a vertical protocol stack. The ZigBee mainly uses IEEE 802.15.4 features, adding ad-hoc networking, services discovery and application protocol profiles. The ZigBee has been successful in multivendor applications such as home automation. However, the ZigBee have several gaps, such as [6]: only support IEEE 802.15.4 data link protocol; provides a limited set of profile applications; and have Internet integration and scalability limitations. Recently ZigBee Alliance announces that will start to integrate IETF standards.

Machine-to-machine communications (M2M) encompasses remote monitoring and control machines over Internet. Personal health monitoring, intelligent tracking and tracing in the supply chain, smart utility industrial wireless automation and ambient assisted living, are examples of M2M applications. Many component-level standards already exists,

addressing various radio interfaces, different meshed or routed networking choices, or offering a choice of identity schemes. Each is optimized for a particular application scenario. Standardization is necessary to permit interoperable multi-vendor solutions. 6LoWPAN can be considered a natural extension to M2M.

Future Internet is a term used to describe research works dedicated to the Internet protocols, services and architectures that will be used in the next 10-20 years. There are several European projects focus in the Future Internet research, namely, EU 4WARD [11], SENSEI [12] and Euro-NF Network of Excellence [13]. The 4WARD project does not address the embedded systems Internet integration. The second specializes in Internet integration embedded systems.

While the Internet of Things paradigm addresses the way of connecting globally the devices, the Web of things regard as the integration of the embedded systems into the Web. Each embedded device should be available using web protocols, such as hypertext transfer protocol.

The term wireless sensor networks (WSN) was appeared in the middle of 90's [2] and have been a subject under intensive research in the past few years. Several factors have contributed to this, mostly because the enormous potential for application of WSNs in almost every aspect of day-to-day life is the predominant one. These networks have been developed to be isolated from the Internet, using proprietary solutions instead of standard solutions. More recently the importance of standards motivated the WSN community to become involved in IETF groups and in IPSO alliance.

There is a strong trend of convergence towards an Internet-based solution to connect all Internet of Things solutions to the Internet and 6LoWPAN protocol may be the convergence solution [6].

C. *Connectivity models*

Three different LoWPAN architecture types were defined. The ad-hoc LoWPAN, with no infrastructure, simple LoWPAN, with one edge router and Extended LoWPAN with multiple edge routers.

Three main models can be used to connect the low power and resource-constrained networks to the Internet. In this first deployment model the low power and resource-constrained network are not connected to the Internet. In fact, there are several scenarios that do not require any connectivity with the Internet, for example the smart grid applications. Smart grid networks are used to monitoring the power generation networks, the automation and control devices, smart metering and building and home energy management. These networks can also use the IP protocol suite in all nodes but due security and privacy reasons in most of he cases are completely disconnected from the public Internet. In this case, supporting IP suite in all devices continues to be advantageous as described in the introduction, although assigning global IP addresses to all devices is not a requirement.

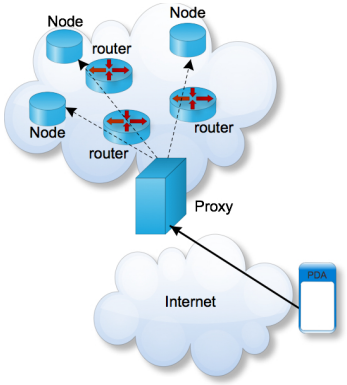


Figure 2 - Connecting the smart objects using a proxy device.

In the second model (Figure 2), a proxy device is used to connect the smart network to the Internet. Internet user will have access to the information provided by smart objects, such as environmental data, using the proxy device. The proxy can act as a server that collects data from the smart objects. This connectivity model can be used to preserve scarce resources on such networks and increase scalability, although does not provide end-to-connectivity. Supporting more than one point of connection between the smart object network and the Internet could be no possible if the proxy uses stateful translation mechanisms. This connectivity model is similar to the previous model. So, the support of IP suite protocol continues to represent a benefit, but assigning IP global addresses to all devices is optional. The second model can be considered an intermediate model between the first model and the smart object full integration in the public Internet.

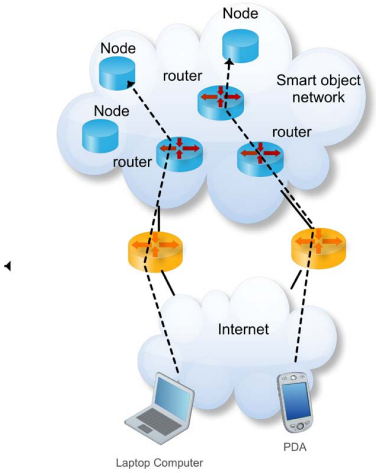


Figure 3 - Extended Internet.

In the third model (Figure 3), the smart object networks are considered as an extension to the Internet. This connectivity model can be used in a near future to support services provided by smart cities, where the citizens can use the Internet to make quotidian decisions based on environmental data such as air

quality, temperature and real-time transportation information provided by the smart objects. All of these networks will make use of the IP protocol suite and one or more router could be used, for redundancy and scalability proposes, to connect these networks to the Internet. In such model, the IP end-to-end connectivity is required and at least one IP global address must be assigned per device.

III. System Architecture

The proposed architecture (Figure 4) is constituted by clients with IPv6 support and connected to the Internet, a gateway to connect the smart objects to the Internet and smart objects compliant with 6LoWPAN and IEEE 802.15.4 protocols. This architecture enables IPv6 end-to-end connectivity between IPv6 nodes and 6LoWPAN nodes. Several layer two protocols can be used to connect the gateway to the Internet, such as IEEE 802.11 a/b/g/n, Ethernet and UMTS/GPRS.

The IPv6 clients can use two different types of connections to retrieve the data from smart objects. In the first type, IPv6 end-to-end connections are used. A multiplatform support application was developed and installed on the clients to permit direct interaction with the smart objects. This application was developed in java and all messages are sent using UDP transport protocol. In fact, the use of TCP transport protocol instead TCP represents a benefit, because the UDP is much simpler then TCP protocol, moreover in the current state only UDP header can be compressed. The packets sent by the IPv6 clients are received by any gateway interface connected to the Internet and processed in the network layer. In the network layer two different operations are performed. First, the packets are inspected by a dynamic packet filter firewall and only relevant packets are permitted. The use of firewalls in this stage avoids some types of denial of service attacks. The permitted packets are forwarded to the routing engine. The packets intended to the smart objects network are then forwarded to the tap virtual interface and then sent to the 6LoWPAN adaptation layer. The 6LoWPAN adaptation layer is responsible for the packet fragmentation and reassembly, in order to support the IPv6 minimum MTU, and for IP and UDP header compression. In the next step, the packet is transmitted to the LoWPAN through the IEEE 802.15.4 interface, connected to the gateway via USB interface. This type of connection is the preferable, because simple gateways can be used and the clients can interact directly with the smart objects. In the second type of connections, the interaction between the IPv6 clients and smart objects is done through a proxy. The clients can retrieve the data accessing a webserver installed in the gateway. In this type of connection, http protocol is used to access the data. Periodically, the logger application installed in the gateway retrieve the data from the smart objects and stores the values in a CVS (comma-separated value) file. The logger and dumper applications are synchronized, so after the logger application ends, the dumper start processing the CVS file in order to put the values in the database. The database is used by the webserver to generate the dynamic webpages requested by the clients. In order to protect the gateway from security attacks, only the http traffic is permitted, all other traffic destined to the gateway is denied by the dynamic packet filter firewall. The second type of connection, does not permit end-to-end connectivity, as a consequence the gateway is more complex when compared with the gateway required in the first type. To support new functionalities is necessary to make changes in the gateway and in the software installed in the smart objects. The second type of connection is used when it is not possible to support end-to-end connectivity between the Internet clients and the smart objects.

Supporting node auto configuration in the smart object network is a requirement. In the proposed system, network auto configuration is supported using 6LoWPAN neighbor discovery protocol. Periodically, the gateway sends a router advertisement message to the smart object network, announcing the prefix and the link local default gateway address. The smart nodes uses the 64 bit announced prefix and the node identifier to generate an IPv6 global address. Configuring an IPv6 global address is an obligation when Internet IPv6 end-to-end connectivity is required.

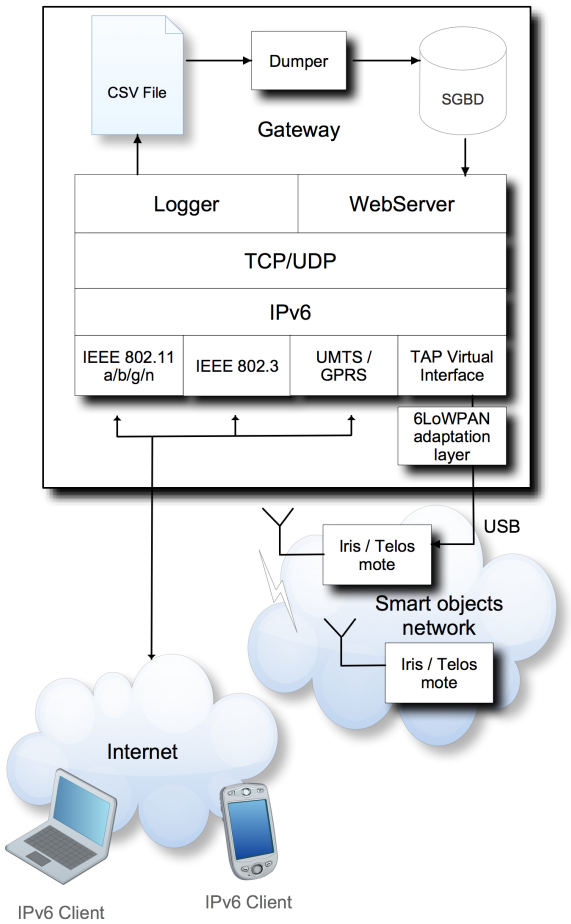


Figure 4 – System architecture.

IV. Performance Evaluation and Demonstration

In order to evaluate the performance and demonstrate the operation of the proposed system a laboratory testbed has been deployed (Figure 5). Five TelosB and five Iris motes, a gateway and a client compose this testbed. This section presents the testbed deployment details and the results obtained.

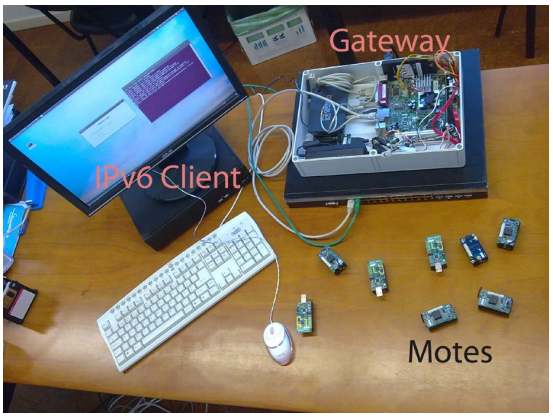


Figure 5 – Laboratory testbed.

A. Internet IPv6 clients

A multiplatform IPv6 client application has been developed to retrieve data and to perform actuation on the smart objects. The application was developed using Java IDE NetBeans. The IPv6 client application (Figure 6) can be used to retrieve the temperature and humidity sensed by the motes. The actuation has been simulated changing the motes LEDs states. UDP protocol is used to transport the requests and the retrieved data. The IPv6 application has been tested with success in MacOS, Windows 7, and Ubuntu 10.0.4. In this testebed the client (regular PC) is connected to the gateway using a Ethernet connection.

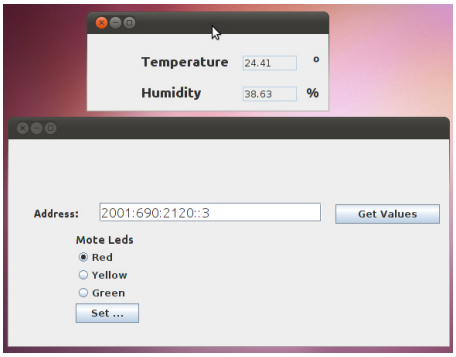


Figure 6 – IPv6 client application.

B. Gateway

The 6LoWPAN gateway is a platform developed to be installed on Ubuntu 10.0.4 freeware and open source OS. It has multiple communication interfaces including IEEE 802.15.4, Ethernet, IEEE 802.11a/b/g and GPRS class 3 modem (Figure 7). The gateway connects to the smart object network through an IEEE 802.15.4 base station, which is represented by a TelosB note connect to an USB port. In the testbed presented in this paper, only GPRS interface has not used, because the IPv6 protocol in unsupported in all available ISPs.

An Intel desktop board D945GCLF with an integrated Intel Atom processor 1.6 GHz has been used to be the motherboard of the gateway. An SSD Corsair drive with 64 GB capacity was used to store the data retrieved from the sensors, the developed gateway applications and the operating system. To run all the software 1GB DIMM module was installed.

The application IP-driver compliant with RFC 4944, provided by TinyOS 2.1, act as the 6LoWPAN adaptation layer in the gateway. A tap (i.e. layer two virtual interface) is used to connect the gateway to the adaptation layer. In the gateway, a version of the IPv6 client software is used to act as a logger. The logger starts automatically with the operating system and runs as a service. The user can configure the frequency used to retrieve de data from the motes. The retrieved data is stored in a CSV file compliant with the following format: *IPv6_node_addrress*; *timestamp*,*temperature*,*voltage*. The dumper was developed in C++ language and starts when dumper application ends. The dumper is responsible to insert the retrieve values in the database. The database was implemented in MySQL 5.5.8 SGBD. The webserver was developed in PHP 5.3.5 and runs over Apache 2.2.17 web server. Periodically or when it receives a router solicitation the *radvd* application send a router advertisement message through the IEEE 802.15.4 interface.

IPtables firewall distributed with Ubuntu 10.0.4 (ip6table) was used permit only UDP port 61616 traffic from the Internet to the smart object network to prevent some type of denial of service attacks. The IPtables is also used to rate limit the traffic from the Internet to the smart network. In our experiment only ten packets per second is permitted. This value can be adjusted.

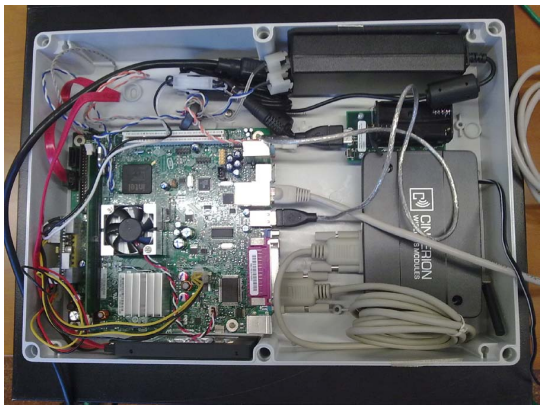


Figure 7 – Gateway hardware implementation.

C. Smart objects

Five IRIS and five TelosB motes, with 6LoWPAN support and equipped with temperature, light and humidity transducers have been used to deploy a single hop smart object network. Address auto-configuration, using router advertisement messages, has been used to configure in all nodes an IPv6 global address with the prefix 2001:690:2120::/64. When the motes are turned on, a router solicitation message is sent to the gateway in order to anticipate the router advertisement message.

TinyOS 2.1 Blip [14] implementation was used to support 6LoWPAN in all smart nodes and a customized version of UDPEcho TinyOS application was used to retrieve de data and to act the motes LEDs. In fact, the regular UDPEcho version does not permit actuating in the mote's LEDs nor reading the values measured by the mote's transducers. So, it was necessary to modify the UDPEcho to allow those operations.

V. Conclusions and Future Work

Smart objects network deployment is far behind what it should be expected. Mainly, because it is hard to deploy new applications and it is difficult to connect this networks to the Internet. Supporting IP suite in all smart objects permits simultaneously facilitate the application development and the connection to the Internet. The system proposed in the current paper presents a complete solution to support IPv6 end-to-end communications between smart objects and all IPv6 nodes. Security mechanisms were used to prevent some denial of service attacks. A laboratory testbed was deployed, using low cost hardware and freeware and open source software.

As a future work, it is intended to support different mote´s operating systems, such as Contiki. Improve the mote's auto-configuration is also a priority. In the current stage only network auto-configuration is possible. In the future authors expected to support service discovery and announcement, and software installation on the fly. Performance evaluation of the current solution will be studied on real testbed deployments.

Acknowledgments

Part of this work has been supported by the Instituto de Telecomunicações, Next Generation Networks and Applications Group (NetGNA), Portugal, in the framework of BodySens and EcoSense Projects, and by the Euro-NF Network of Excellence from the Seventh Framework Programme of the EU, in the framework of the PADU Project.

Lei Wang's work is partially supported by Natural Science Foundation of China under Grant No. 61070181, and Natural Science Foundation of Liaoning Province (China) under Grant No. 20102021.

Lei Shu's research work in this paper was supported by Grant-in-Aid for Scientific Research (S)(21220002) of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] N. Gershenfeld, R. Krikorian, D. Cohen, "The Internet of Things," Scientific American, vol. 291, no. 1, 2004, pp. 76-81.

[2] L. Oliveira, A. Sousa, J. Rodrigues, "Routing and Mobility Approaches in IPv6 over LoWPAN Mesh Networks", International Journal of Communication Systems, Wiley, ISSN: 1074-5351, DOI: 10.1002/dac.1228 (in press).

[3] IEEE Std 802.15.4-2006. Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs). IEEE Std. 802.15.4-2006, 2006.

[4] ZigBee Alliance, "ZigBee specification: ZigBee document 053474r013 Version 1.1," ZigBee Aliance, December. 2006, http://www.zigbee.org, accessed in January 2011.

[5] D. Chen, M. Nixon, A. Mok, "WirelessHART: Real-Time Mesh Network for Industrial Automation," Elsevier, 2010. ISBN 978-1441960467.

[6] J. Hui, D. Culler, "Extending IP to Low-Power, Wireless Personal Area Networks," IEEE Internet Computing, vol. 12, no. 4, 2008, pp. 37-45.

[7] N. Kushalnagar, G. Montenegro, C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," Internet Engineering Task Force, Request for comments 4919, August 2007.

[8] M. Harvan, J. Schönwälder, "TinyOS Motes on the Internet: IPv6 over 802.15.4 (6lowpan)", PIK - Praxis der Informationsverarbeitung und Kommunikation, 2008, 4(31): 244-251.

[9] Baronti P, Pillai P, Chook V, Chessa S, Gotta A, Hu Y. Wireless sensor networks: a survey on the state of the art and the 802.15.4 and ZigBee standards. Computer Communications Wired/Wireless Internet Communications 2007; 30(7):1655–1695.

[10] Z. Shelby, P. Thubert, J. Hui, S. Chakrabarti, C. Bormann, E. Nordmark, "6LoWPAN Neighbor Discovery," Internet Draft draft-ietf-6lowpan-nd-15, 2011, working progress.

[11] 4Ward project, URL: http://www.4ward-project.eu, accessed in January 2011.

[12] SENSEI project, URL: http://www.sensei-project.eu/, accessed in January 2011.

[13] NoE Euro-NF Network of Excellence, URL: http://www.euronf.org, accessed in January 2011.

[14] TinyOS, "Blip tutorial," Aug. 2010. [Online]. Available: http://docs.tinyos.net/index.php/BLIP Tutorial.