UNIVERSIDADE DA BEIRA INTERIOR
Faculdade de Engenharia
Departamento de Informática

Performance Assessment of Real-Time Data Management on Wireless Sensor Networks

Ousmane Diallo

Thesis for obtaining the degree of Doctor of Philosophy in

Computer Science and Engineering

(3rd Cycle Studies)

Supervisor: Prof. Dr. Joel José Puga Coelho Rodrigues (University of Beira Interior)

Co-supervisor: Dr. Mbaye Sene (University of Cheikh Anta Diop of Dakar)

Covilhã, May 2014

Thesis prepared at the *Departmento de Informática* within the Next Generation Networks and Applications Group (NetGNA), *Instituto de Telecomunições*, University of Beira Interior, Covilhã, Portugal, and at the Department of Mathematics and Informatics within the PhD School of Mathematics and Informatics (EDMI), UCAD, Senegal, and submitted to University of Beira Interior in candidature for the Degree of Doctor of Philosophy in Computer Science and Engineering.

NetGNA Next Generation Networks and Applications Group

Networks and Multimedia
Instituto de Telecomunicações,
Portugal

Department of Informatics
University of Beira Interior,
Portugal

University of Cheikh Anta Diop of Dakar,
Senegal

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 and Pest-OE/EEI/LA0008/2013 projects, 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*.

Dedicatory

To my family.

For all the love and support.

To my sisters Racky, Khady and Ndeye Ngoné.

Your memories will be engraved in our heart forever.

May God welcome you in his Heaven!!

Acknowledgments

I would like to thank here those who helped, supported me from near and far during the completion of this thesis research.

First of all I thank the reporters, which devoted their time to read and value my thesis, as well as the members of the jury for questions, criticisms and suggestions during the presentation.

I have to express my gratitude to my supervisor, Prof. Dr. Joel José Puga Coelho Rodrigues, for the confidence he has placed in me by offering me the possibility of pursuing my PhD, and for the atmosphere of serenity, the guidance, and motivation, especially in the most difficult moments. I thank him also for offering me the possibility to come to Portugal to improve and pursue my research work in his Laboratory. I am also thankful to my co-supervisor, Prof. Dr. Mbaye Sene, for having been my supervisor since my Master degree, and for also giving me the opportunity to develop the PhD research in his office. Nevertheless, more than for anything else, I would like to thank them for their great friendship.

I am also thankful to my friend Modou Gueye for all the amazing discussions we had since the year of my Master degree, for all the help and support, and for the long talks that always provide good ideas. Moreover, I would like to thank him for his great friendship.

My recognition is also due to SAPO from Portugal Telecom for offering me the possibility to stay in Portugal during one year of research work under the Sensor Fall Project.

Being part of the *Next Generation Networks and Applications Group* (*NetGNA*) at the *Instituto de Telecomunições*, *UBI*, Covilhã, Portugal, and being part also of the *Network and Performance Group* at the *Department of Mathematics and Informatics*, *UCAD*, Sénégal, I thank my cooperative, talented and creative colleagues for all the support and motivation during this research work. I am particularly grateful to all the friends I made in the research group NetGNA, not only for all the discussions we had as colleagues, but also for the jokes, the lunches, the pool parties, the football matches on Tuesdays, the sleigh ride on the snow on top of Serra da Estrela, and many other moments we had as friends. Particularly, I thank João Amaral for the help in the Portuguese translation and for the print of the document.

I am also grateful to the secretaries and the teaching staff of my two universities, particularly to Patricia from IT and Dulce from UBI. I cannot forget the wife of my supervisor Celina, my medical doctor. You always helped me during all my stays in Portugal.

I am equally thankful to my family, particularly my parents for having been by my side in the hardest moments of this work, always giving me encouragement and support. But, most of all, I would like to thank them for the love they always gave me and for having taught me very good education.

Last, but not least, I have to thank my brothers and sisters Waly, Mass, Wayndé, Mamadou, Anta, Safy, Awa, Bigué and my uncle Issa Gueye for your prayers and my wife Kene Fall for your love on my modest person, your comprehension and patience when I was always absorbed by this hard work, and for your help and support whenever I needed it.

“ Our doubts are traitors, and rob

us of what we could often earn of

good, because we are afraid to try. ”

William Shakespeare

List of Publications

Papers included in the thesis resulting from this 3-year doctoral research programme.

[1] Ousmane Diallo, Joel J. P. C. Rodrigues, and Mbaye Sene, "Real-time Data Management on Wireless Sensor Network: a Survey", in Journal of Network and Computer Applications, Elsevier, ISSN: 1084-8045, Vol. 35, Issue 3, May 2012, pp. 1013-1021, DOI: 10.1016/j.jnca.2011.12.006. ISI Journal Citation Report with an impact factor of 1.467 in 2012, Computer Science, Hardware and Architecture (Q1; 11th place); Computer Science, Software and Engineering (Q1; 24th place).

[2] Ousmane Diallo, Joel J. P. C. Rodrigues, Mbaye Sene, and Jaime Lloret "Distributed Database Management Techniques for Wireless Sensor Networks", IEEE Transactions on Parallel and Distributed Systems, ISSN: 1045-9219, 19 Aug. 2013, IEEE Computer Society, DOI: 10.1109/TPDS.2013.207 (in press). ISI Journal Citation Report with an impact factor of 1.796 in 2012, Computer Science (Q1; 16th place).

[3] Ousmane Diallo, Joel J. P. C. Rodrigues, Mbaye Sene, and Jaime Lloret "Simulation Framework for Real-Time Database on WSNs", in Journal of Network and Computer Applications, Elsevier, ISSN: 1084-8045, Vol. 39, March 2014, pp. 191-201, DOI: 10.1016/j.jnca.2013.07.001. ISI Journal Citation Report with an impact factor of 1.467 in 2012, Computer Science, Hardware and Architecture (Q1; 11th place); Computer Science, Software and Engineering (Q1; 24th place).

[4] Ousmane Diallo, Joel J. P. C. Rodrigues, Mbaye Sene, and Feng Xia, "Real-Time Query Processing Optimization for Wireless Sensor Networks",International Journal of Sensor Networks (IJSNet), InderScience Publishers, ISSN (Online): 1748-1287 - ISSN (Print): 1748-1279 (in press). ISI Journal Citation Report with an impact factor of 1.386 in 2011, Engineering Miscellaneous (Q2); Computer Networks and Communications (Q4).

[5] Ousmane Diallo, Joel J. P. C. Rodrigues, Mbaye Sene, Jianwei Niu, "Real-Time Query Processing Optimization for Cloud-based Wireless Body Area Networks", in Information Sciences, Elsevier, ISSN: 0020-0255 (in press). ISI Journal Citation Report with an impact factor of 3.643 in 2012, Computer science; Information Systems (Q1; 6th place).

Resumo Alargado

O presente resumo alargado, em língua Portuguesa, apresenta o trabalho realizado para a preparação da tese de doutoramento intitulada "*Performance Assessment of Real-Time Data Management on Wireless Sensor Networks*". Começa por apresentar o enquadramento da tese, centrando-se, em seguida nas temáticas mais relevantes a abordar na tese como são os sistemas de bases de dados, os sistemas em tempo real e os sistemas de gestão de bases de dados em tempo-real, considerando também as redes de sensores sem fios e gestão de dados. Depois, define o problema e fixa os objectivos da investigação, inclui o argumento da tese e apresenta as suas principais contribuições para o avanço do estado da arte. O capítulo termina com uma breve discussão das principais conclusões e sugestões para investigação futura.

Enquadramento da Tese

Actualmente, os sistemas baseados em redes de sensores são cada vez mais utilizados em muitas áreas do conhecimento, dando origem a aplicações em diversas áreas, como os mercados financeiros, aplicações para rastreabilidade de movimentos humanos, a monitorização dos fenómenos urbanos e ambientais, monitorização de pacientes, produção mecanizada, controlo de aeronaves e militares, entre outras. Algumas destas aplicações são designadas por aplicações em tempo-real e têm a particularidade de ter que respeitar as restrições lógicas e de consistência impostas pelo sistema, mas também as limitações temporais relativas à velocidade de execução de operações, bem como a relação com os seus prazos. Além disso, estas aplicações devem ser capazes de lidar com grandes quantidades de dados provenientes de sensores, necessários para o seu correcto funcionamento. Assim, a utilização de bases de dados é necessária e indispensável para este tipo de sistemas.

Juntamente com os requisitos inerentes às bases dados tradicionais fornecidos por sistemas de gestão de bases de dados (SGBD), tais como a garantia de integridade e consistência, a capacidade de partilhar dados e possíveis recuperações após falhas, as bases de dados em tempo-real devem também ser capazes de satisfazer as restrições temporais introduzidas por sistemas em tempo-real. Assim, as bases de dados em tempo-real (BD-TR) são essenciais para sistemas em tempo-real com restrições temporais não-negociáveis, tais como aplicações de aviação nas quais os prazos em dados temporais e transações não podem ser perdidos por causa do risco de gerar um acidente. De igual modo, as bases de dados em tempo-real são úteis para sistemas em tempo-real que estão instalados em ambientes imprevisíveis, tais como mercados financeiros e aplicações de acompanhamento de movimentos humanos, onde satisfazer a maioria das restrições temporais significa obter o melhor desempenho do sistema.

A concepção de soluções eficazes para gestão de dados em tempo-real para as redes de sensores sem fios está no centro desta tese e é particularmente difícil na medida em que este tipo de sistemas envolve duas áreas bastante complexas: as bases de dados em tempo-real e as redes de sensores sem fios. No entanto, foram propostas soluções que seguem uma abordagem distribuída ou centralizada, ou algoritmos específicos com modelos formais para definir os níveis de precisão e desempenho que devem ser satisfeitos. O enquadramento da tese é descrito com maior detalhe nas seções seguintes.

Sistemas de Bases de Dados

Um sistema de bases de dados é constituído por um conjunto de dados estruturados denominado por base de dados e uma aplicação de software designada por sistema de gestão dos bases de dados (SGBD). Uma base de dados é um grande conjunto de dados estruturados armazenados num meio permanente. A base de dados é um componente essencial dos sistemas computacionais modernos, como por exemplo, sistemas de gestão de informação (bancos, seguros, contabilidade, gestão de inventário, gestão de vendas, gestão da produção, etc), livrarias electrónicas, bases de dados científicas, etc. Um SGBD é um conjunto de programas que permite a gestão e acesso a uma ou mais bases de dados.

Uma base de dados é normalmente acedida por vários utilizadores com diferentes perspectivas sobre os dados. Para evitar inconsistências causadas pelo acesso múltiplo, um SGBD deve fornecer mecanismos para controlar a execução simultânea de vários programas na base de dados.

O acesso ao conteúdo da base de dados é realizado através de transacções [1]. A transacção, por definição, é uma sequência de acções que transformam a base de dados de um estado consistente para outro estado consistente. Tais acções são principalmente as operações de escrita ou leitura de dados chamados de *grânulos* (tuplos, páginas, ...) e as operações de manipulação (cálculos, testes, ...). Uma base de dados é consistente se é correcta do ponto de vista do utilizador, ou seja, mantém os invariantes da base ou as restrições de integridade. A noção de consistência abrange várias dimensões, tal como descrito em [2]. Do ponto de vista dos pedidos de acesso, trata-se da gestão da execução simultânea de múltiplas transacções, sem que as actualizações de uma transacção sejam visíveis antes da sua validação designada por consistência transacional ou isolamento.

As execuções concorrentes de transacções em sistemas de bases de dados são geridas por protocolos de controlo de concorrência baseados na noção de *serialização* que determina que as transacções podem ser executadas simultaneamente [3-7]. A execução serializável é uma execução entrelaçada das acções de um conjunto de transacções $\{T_1, T_2, ..., T_n\}$, que resulta para cada transacção participante no mesmo resultado que a execução serial de $T_1, T_2, ..., T_n$. Estes protocolos garantem as propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade) [8, 9] de transacções que impõem restrições sobre a ordem de execução de transacções e os direitos de acesso aos dados (restrições de consistência lógica de dados e transacções).

Sistemas em Tempo-Real

Em computação, um sistema é qualificado em tempo-real quando é capaz de controlar um processo físico a uma velocidade apropriada para evolução do processo controlado. Os sistemas de computação em tempo-real diferem de outros sistemas informáticos, tendo em conta as limitações temporais cujo cumprimento é tão importante quanto a precisão do resultado, ou seja, o sistema não deve apenas fornecer resultados precisos, deve entregá-los nos prazos. Hoje em dia, os sistemas de computação em tempo-real estão presentes em diversos sectores, nomeadamente na indústria de transformação, por exemplo, através de sistemas de controlo de processo (fábricas, centrais nucleares), o tratamento dos dados de mercado em tempo-real, em aeronáutica através de sistemas de controlo integrado (aeronaves, satélites), no processamento e encaminhamento de informação (vídeo, dados, controlo remoto, realidade virtual, etc.) [10].

Dois elementos distintos podem ser destacados em sistemas em tempo-real [11]: uma ou mais entidades físicas que constituem o *processo* cujo papel é o de actuar e de detectar, e um controlo de computação chamado *controlador* ou *aplicação em tempo-real*, que é o decisor do processo. O controlador recebe informações do ambiente do processo através de sensores e controla as mudanças de estado do processo através de *actuadores*.

Estes sensores e actuadores são controlados por *tarefas* que são programas sequenciais dedicados ao tratamento de um dos componentes do sistema em tempo-real. Por exemplo, um programa em tempo-real pode ser constituído por um conjunto de *tarefas* tais como a execução periódica de medidas de diferentes grandezas físicas (pressão, temperatura, aceleração, etc), tratamentos com intervalos regulares ou agendados, etc.

Existem três tipos de tarefas em tempo-real, que diferem nas suas características temporais [11]: *tarefas periódicas* que têm actividade regular, as *tarefas aperiódicas* que são activadas de forma aleatória, dependendo, por exemplo, de eventos aleatórios, e as *tarefas cíclicas*. As tarefas aperiódicas têm uma subfamília de tarefas chamada *tarefas esporádicas* para as quais um período mínimo separa duas ocorrências sucessivas do evento desencadeador. Finalmente, *tarefas cíclicas* [12], que são muito próximas das tarefas periódicas, mas com a sublime diferença que a duração entre duas activações sucessivas de uma tarefa periódica é constante, enquanto que pertence a um intervalo $[P_{min}, P_{max}]$ para tarefas cíclicas.

Por forma a verificar se as restrições temporais são satisfeitas, são introduzidas indicações temporais quantitativas nos modelos de tarefas. Assim, por exemplo, considera-se uma tarefa periódica θ_i com base no modelo temporal originalmente introduzido por [13]. Estas indicações temporais são dadas sob a forma de *tempo de libertação* (l_i), *tempo de computação* (C_i), o *prazo* (D_i), e a *periodicidade* (P_i).

- *O tempo de libertação* (l_i) representa o instante de tempo em que a primeira instância de θ_i é ativada e é chamada de *data de despertar*. Quando todas as tarefas têm a mesma data de despertar, existe uma instância de cada tarefa pronta para ser executada;
- O *tempo de computação* (C_i), também chamado de pior tempo de execução ou carga máxima de θ_i, representa o tempo de execução necessário para executar a tarefa θ_i sem interrupção. Esta especificação de tempo é crucial para o sucesso do escalonador das tarefas. Determinar este pior tempo de execução é geralmente muito difícil. De facto, seja por análise dinâmica (medida directa de uma execução) ou análise estática (exploração do código da tarefa) [14, 15], a sua avaliação é dificultada pela presença de instruções condicionais, ciclos não determinísticos [16, 17, 18], ou melhorias em processadores, como a execução especulativa, que também os torna não determinísticos [19, 20, 21]. Assim, este parâmetro é um dos componentes mais importantes para a modelação do sistema de tarefas, uma vez que é o único valor aproximado *a priori*;
- O *prazo* (D_i) determina o tempo alocado para cada instância da tarefa θ_i para a sua execução, isto é, o período de tempo máximo permitido entre a activação e encerramento da instância. Este parâmetro é o mais interessante, em termos de determinação da eficiência e correção do sistema.
- Finalmente, o último parâmetro, a *periodicidade* (P_i), quantifica o intervalo de tempo mínimo entre duas activacções sucessivas da mesma tarefa.

É possível determinar as datas de activação para cada instância da tarefa θ_i: dado $j \in N^+$, então a data de despertar da instância j^{th} da tarefa θ_i é:

$$l_i^j = l_i + (j-1) * P_i \qquad (1)$$

Da mesma forma, sucessivas datas de prazos para cada instância de uma tarefa θ_i com $j \geq 1$ são dados por

$$d_i^j = l_i^j + D_i \qquad (2)$$

A qualidade de serviço esperada para a avaliação de aplicações em tempo-real depende da violação ou não das restrições temporais. Assim, podem-se classificar os sistemas em tempo-real de acordo com a rigidez das restrições temporais que lhes foram impostas:

- *Sistemas em tempo-real suaves* são menos exigentes. O não cumprimento de um prazo não origina uma falha no sistema. Estas violações são toleradas, mas vão causar perturbações que devem ser minimizadas. Neste tipo de sistema, a medida de eficácia é geralmente realizada através de uma análise estatística dos tempos de resposta médios e pode ser encontrado em sistemas de processamento multimédia, tais como o *streaming*, por exemplo.
- *Sistemas em tempo-real firmes* [22] são mais restritivos. Neste tipo de sistema, a não-conformidade com as restrições temporais deve ser evitada, uma vez que acarreta

consequências mais graves para o sistema. No entanto, reconhece-se que tais violações podem ocorrer excepcionalmente.

- *Sistemas em tempo-real rígidos* são os mais graves. O fracasso de restrições temporais pode levar a falhas com consequências potencialmente graves. É, portanto, imperativo que as restrições sejam satisfeitas. Estes sistemas são frequentemente encontrados em sistemas embutidos, tais como aviação, robótica, etc.
- *Sistemas em tempo-real mistos* estão sujeitos às exigências de ambos os sistemas com restrições rígidas para certas tarefas e os sistemas com restrições suaves para as outras. Estes sistemas incluem os mais recentes sistemas de tempo-real.

Características dos Sistemas de Gestão de Bases de Dados em Tempo-Real

Como um SGBD tradicional, um SGBD em tempo-real (SGBD-TR) deve processar transacções e garantir que a consistência lógica dos dados não é violada. No entanto, ao contrário de um SGBD tradicional, um SGBD-TR enfatiza sobre a validade temporal dos dados e das restrições de tempo ou prazos para transacções [23-25].

Dados em tempo-real

Uma das muitas questões levantadas pela concepção de um SGBD-TR é manter a consistência dos dados na base de dados [26, 27]. De facto, o estado actual do ambiente-alvo perceptido pelo sistema do controlador deve refletir, tanto quanto possível, o estado real do ambiente. Este requisito tem um impacto sobre a concepção do SGBD-TR, que deve não só respeitar as restrições de integridade (coerência lógica), mas também respeitar as restrições de consistência temporal dos dados.

Consistência Temporal dos Dados em Tempo-Real

A consistência temporal pode ser medida de duas formas [28]:

- *Consistência absoluta*, que lida com a necessidade de manter a visão que representa o estado do ambiente-alvo de acordo com o estado real do ambiente;
- *Consistência relativa*, que diz respeito a dados provenientes de outros ambientes.

De acordo com [29], estes dois conceitos para os dados em tempo-real podem ser definidos da seguinte forma: um conjunto de dados usado para derivar um novo item de dados constitui um conjunto com *consistência relativa*. Cada conjunto R está associado com um intervalo de validade relativa denotado por R_{ivr}. Dado um item de dados em tempo-real $d \in R$, d tem um estado correcto se:

1. d_{valor} é a consistência lógica, ou seja, satisfaz todas as restrições de integridade.
2. d é temporalmente consistente:
   - ✓ consistência absoluta: $(tempo\_corrente - d_{rotulo\_tempo}) \leq d_{iva}$

- ✓ consistência relativa: $\forall d' \in R$, $|d_{rotulo\_tempo} - d'_{rotulo\_tempo}| \leq R_{ivr}$,

Onde d_{valor} é o valor real dos dados *d*, d_{rotulo_tempo} representa o instante de tempo em que esse valor foi medido, e d_{iva} é o intervalo de validade absoluta do valor de *d*.

Representação de dados em tempo-real

Os dados em tempo-real representam a captação do estado actual e devem reflectir o estado actual do ambiente-alvo. No entanto, o ambiente muda constantemente e os dados são recolhidos em tempos discretos por forma a apresentarem restrições temporais.

Para satisfazer estas restrições temporais, a estrutura dos dados deve incluir os seguintes atributos : (*i*) *rótulo_tempo*, que indica o instante em que a observação relativa aos dados foi feita, e (*ii*) o *intervalo de validade absoluta* (*iva*) que indica o intervalo de tempo a seguir o rótulo de tempo em que os dados são considerados válidos. Estes dados podem ser anotados como d = (d_{valor}, d_{rotulo_tempo}, d_{iva}). Por exemplo, vamos considerar um sistema com dois dados, temperatura e pressão, com $Temperatura_{iva} = 5$, $Pressão_{iva} = 10$, *R = {temperatura, pressão}*, o intervalo de validade relativa $R_{ivr} = 2$. Se o *tempo_corrente* é igual a *100*, então a pessoa tem, de acordo com as duas condições anteriores:

- ✓ a temperatura = (347, 96, 5) e a pressão = (50, 97, 10) têm consistência temporal.
- ✓ no entanto, a temperatura = (347, 96, 5) e a pressão = (50, 92, 10) não são temporalmente consistentes (a consistência relativa não é atingida).

Para a qualidade dos dados, outro atributo que pode ser considerado é a *imprecisão* ou o *erro de dados* (*ED*), que se refere à forma como o estado actual do ambiente-alvo pode ser diferente dos dados medidos [29]. O *erro de dados* na versão de dados *d* é definido por (3).

$$\mathrm{ED(d)} = 100 * \left|\frac{\mathrm{ValorCorrente(d)} - \mathrm{ValorNovo(d)}}{\mathrm{ValorCorrente(d)}}\right| \% \qquad (3)$$

Transacções em Tempo-Real

Os métodos utilizados em SGBD tradicionais não são geralmente adequados para SGBD em tempo-real. No entanto, podem ser usados e adaptados para ter em conta as limitações de SGBD em tempo-real que envolvem a exigência de que as transacções sejam executadas antes dos seus prazos e que estas operações de acesso a dados válidos sejam efetuadas com períodos de validade limitados, o que acarreta implicações na concepção destes sistemas. Em particular, as propriedades *ACID* das transacções das bases de dados tradicionais foram revistas e adaptadas ao contexto em tempo-real [25, 29, 30]. De facto:

- ♦ A propriedade de *atomicidade*, em bases de dados tradicionais, especifica que uma transacção deve completar a sua execução ou não ser executada. No entanto, esta propriedade é relaxada em bases de dados em tempo-real. Na verdade, se a transacção é composta por sub-transacções, essa propriedade só é aplicada às sub-transacções que querem lidar com consistência completa dos dados;

- A propriedade de *consistência* indica que dados inconsistentes podem ser lidos. Como o cumprimento dos prazos das transacções é muitas vezes mais importante do que a exatidão dos resultados, em muitas situações, a correção pode ser trocada por oportunidade;
- A propriedade de *isolamento* especifica que as operações de uma transacção não devem ser visíveis para outras transacções até que esta esteja terminada. Para satisfazer esta propriedade, transacções simultâneas devem ser seriadas. No entanto, para as bases de dados em tempo-real, a execução concorrente das transacções é considerada correcta se as transacções cumprirem os seus prazos com novos dados. Mesmo que essa execução simultânea de transacções não satisfaça os critérios de seriação;
- O conceito de *durabilidade* não tem o mesmo significado num SGBD em tempo-real como num SGBD tradicional. Num SGBD tradicional, esse conceito significa que as alterações feitas numa base de dados se tornam permanentes (persistentes), assim que as transacções de actualização são validadas. No entanto, os dados no SGBD em tempo-real devem refletir o estado actual do ambiente, embora esse estado mude constantemente. Para isso, os sensores actualizam periodicamente o conteúdo da base de dados. Assim, a persistência desses dados é verificada apenas durante o seu período de validade.

Protocolos de controlo de concorrência

Num SGBD tradicional, o escalonador de seriação é o padrão aceite para manter a consistência da base de dados no caso de acesso simultâneo por várias transacções. Por outras palavras, se as transacções concorrentes são ordenáveis, então a base de dados é mantida num estado consistente após a execução dessas transacções.

Escalonadores com ordenação são difíceis de aplicar num contexto de tempo-real. Na verdade, o critério de seriação é muito rigoroso para a execução das transacções e o acesso aos dados em tempo-real. Portanto, as técnicas de controlo de concorrência das transacções desenvolvidas em SGBD tradicionais não são directamente aplicáveis aos SGBD em tempo-real. De facto, numa base de dados em tempo-real, é necessário manter, para além da consistência lógica dos dados, a sua coerência temporal.

Por outro lado, os algoritmos concebidos para programar tarefas em tempo-real [31] não podem ser aplicados directamente a transacções das bases de dados para integrar restrições temporais. Por um lado, trabalhos no controlo de concorrência das transacções no SGBD em tempo-real dependem de técnicas de controlo de concorrência (CC) tradicionais das transacções num SGBD e, por outro, sobre técnicas de escalonador de tarefas em sistemas de tempo-real. Técnicas de controlo de concorrência de transacções no SGBD em tempo-real são baseadas em duas políticas:

- *Política CC optimista* [32, 33, 34] é uma técnica de tipo curativo. Os algoritmos desta classe permitem que as transacções sejam executadas simultaneamente, assumindo o risco de terem de ser reiniciados se aparecerem inconsistências na base de dados. Esta política é chamada de optimista porque o pressuposto considerado é que há uma baixa probabilidade de que duas transacções entrem em concorrência no mesmo grânulo de dados (dado elementar de uma base de dados acedido por uma transacção). Permite que transacções executem e o controlo é efetuado para assegurar a seriação no final das operações. Neste caso, se duas transacções concorrem por um objecto, uma delas será abortada e reiniciada. Para este protocolo não existe indeterminação no tempo de espera mas o número das transacções que podem ser reiniciadas é grande, o que pode ser um factor negativo em sistemas com restrições de tempo.

- *Política CC Pessimista* é uma técnica de controlo do acesso simultâneo que consiste em bloquear os objectos para leitura por uma transacção e libertar bloqueios somente após a obtenção de todos os "trincos". Os protocolos mais comummente usados para bloqueio são do tipo "Bloqueio em Duas Fases" (2PL) [35]. Os algoritmos desta classe evitam a execução concorrente de transacções quando há potenciais conflitos. Esta política é chamada de pessimista porque o pressuposto subjacente é que qualquer par de transacções que é executado em concorrência é provável que entre em conflito. Por conseguinte, uma das transacções espera até que a outra seja validada. No entanto, a aplicação deste protocolo pode gerar tempos de espera por um período indefinido de bloqueio. Assim, a utilização deste mecanismo pode ser vantajosa para manter a consistência lógica da base de dados, mas em contrapartida pode comprometer as restrições temporais impostas sobre as transacções.

Outras técnicas são propostas para assegurar uma coerência lógica e temporal. Entre elas destaca-se o trabalho em [36], que apresenta uma técnica de controlo de concorrência semântico orientada a objectos, chamada *técnica de bloqueio semântico*. Com esta técnica é possível fornecer consistência lógica e temporal dos dados e transacções definindo os critérios para a negociação entre eles. É também possível expressar a imprecisão resultante da negociação utilizando o conceito de bloqueio semântico para determinar que transacções podem chamar os métodos de um objecto. O bloqueio semântico é controlado em cada objecto individualmente por uma função de compatibilidade (FC), que implementa mecanismos para controlar o acesso simultâneo aos métodos do objecto.

Uma técnica um tanto semelhante pode ser observada em [7], que se baseia no critério de Seriação Epsilon (*SE*) [37]. A *SE* relaxa a gravidade da seriação clássica no processamento das transacções, permitindo uma inconsistência limitada na base de dados. Esta inconsistência limitada é automaticamente mantida pelos algoritmos de controlo de divergência (CD), da mesma maneira que a seriação clássica é gerida pelas técnicas de controlo de concorrência (CC). No capítulo 4, é apresentado um modelo para simular técnicas de bases de dados em tempo-real sobre redes de sensores, que utiliza um algoritmo de CD com duas fases [7] chamado 2PLDC. O 2PLDC é uma extensão do algoritmo de controlo de concorrência 2PL da serialização clássica.

Modelos de Bases de Dados em Tempo-Real

O modelo relacional é o mais utilizado para modelar dados para bases de dados em tempo-real. Entre estes modelos incluem o modelo de Ramamritham [29] em que os dados podem ter restrições temporais absolutas e relativas.

Muitos investigadores argumentam que o modelo orientado a objectos é mais adequado do que o modelo relacional [42] para modelar dados em tempo-real por causa da natureza de diversas aplicações em tempo-real que lidam com objectos complexos do mundo real com limitações de tempo. Por isso, muitos projectos com bases de dados em tempo-real escolheram o modelo orientado a objectos para os seus sistemas [43, 44]. Neste último projecto [44] é apresentado o RTSORAC (*Real-Time Semantic Objects Relationships And Constraints*). O modelo RTSORAC inclui três componentes para modelar as características de uma base de dados em tempo-real orientada a objectos: os objectos, as relações e os métodos. Os objectos representam as entidades do sistema, as relações representam as associações entre os objectos e definem as restrições inter-objectos na base de dados. Os métodos são entidades executáveis que acedem a objectos e relações na base de dados. Um conjunto de restrições para exprimir as restrições lógicas e temporais são definidas para especificar corretamente um objecto. No capítulo 4, o modelo proposto para a simulação de técnicas de base de dados em tempo-real sobre redes de sensores sem fios é baseado no modelo orientado a objectos.

Redes de Sensores sem Fios e Gestão de Dados

Geralmente, uma rede de sensores sem fios tem um grande número de nós distribuídos numa zona de interesse e a comunicando entre eles, de modo a medir uma grandeza física (por exemplo, nível de poluição numa dada área) ou a fazer uma monitoração de eventos (por exemplo, localização de veículos). As redes de sensores sem fios são usadas com diferentes aplicações em muitas áreas e são muito importantes para aplicações que devem ser implementadas em locais hostis para intervenções humanas (por exemplo, a monitorização de vulcões). Cada nó da rede é considerado inteligente e incorpora as seguintes unidades: uma unidade de sensor que fornece uma medida de dados ambientais (tais como temperatura, humidade, pressão, aceleração, som, etc), uma unidade de processamento, uma unidade de armazenamento, uma unidade de comunicação, e uma unidade de energia. A unidade de comunicação geralmente executa a transmissão de dados através de rádio-frequência [45, 46, 47, 48]. No entanto, os recursos de um sensor são geralmente muito limitados, especialmente em termos de armazenamento e energia. O tempo de vida da rede de sensores depende da energia disponível nos nós que compõem a rede [49, 50]. Esta energia disponível é consumida por três actividades [46]: atividade de detecção (aquisição de dados a partir do ambiente), comunicação (envio e recepção de pacotes) que é essencial para formar uma rede de sensores sem fios, e processamento de dados que consiste em algumas operações, aplicadas sobre os dados por sensores inteligentes [51, 52]. As atividades de sensoriamento e processamento são muito menos dispendiosas em consumo de energia do que as atividades de comunicação sem fios [53]. Assim, a conservação de energia deve ser o principal foco de atenção dos algoritmos projetados para redes de sensores.

Assim que os sensores realizem a sua medição, o problema de armazenamento dos dados de consulta surge. De facto, os sensores têm capacidade de armazenamento restringida [54] e a interação contínua entre dispositivos de rede e o ambiente resulta em grandes quantidades de dados. Existem duas abordagens principais para armazenamento de dados e consultas em redes de sensores sem fios [55, 56]: a abordagem distribuída e a abordagem *warehousing*.

1. A *abordagem warehousing* representa um sistema centralizado no qual os sensores funcionam como simples coletores de dados [57]. Os dados recolhidos pelos sensores são enviados periodicamente para uma base de dados central, na qual as consultas dos utilizadores são processadas. Este modelo é o mais utilizado no armazenamento de dados e processamento das consultas. No entanto, tem alguns inconvenientes, tais como eventuais recursos desperdiçados e afunilamento com uma imensa quantidade de dados transmitidos. Além disso, esta abordagem não é adequada para o processamento em tempo-real, uma vez que envolve atraso de tempo para os resultados.
2. A *abordagem distribuída* é a alternativa, onde os sensores são capazes de armazenar, processar localmente e transmitir os dados que produzem [52, 58]. Nesta abordagem, os dados de sensoriamento não são enviados periodicamente para o servidor de base de dados. Eles permanecem nos nós sensores e algumas consultas são distribuídas e avaliadas entre os nós na rede, reduzindo o consumo de energia e a transferência de dados e, portanto, aumentando o tempo de vida da rede. Esta abordagem que consiste em processar os dados dentro dos próprios nós sensores é chamada *processamento em rede* e, além da minimização de energia, pode oferecer várias outras vantagens, tais como o processamento em quase tempo-real, e suporte a consultas instantâneas e consultas longas. Para mais detalhes sobre esta abordagem e as propostas actuais e relevantes, os leitores podem ver Capítulo 3.

Existem três tipos de transacções em redes de sensores: (*i*) *consultas de dados históricos*, que são executadas no servidor, (*ii*) *consultas instantâneas*, que são executadas no sensor inteligente num instante de tempo, e (*iii) consultas longas*, que se referem às consultas executadas no sensor inteligente durante um intervalo de tempo [55, 60].

O âmbito desta tese está limitado à gestão de bases de dados em tempo-real para redes de sensores sem fios. O trabalho de investigação aqui apresentado centra-se no estudo dos desafios em lidar com o armazenamento e consulta de dados em tempo-real para redes de sensores sem fios e sobre as várias técnicas de gestão de bases de dados em tempo-real para redes de sensores sem fios. A limitação energética intrínseca das redes de sensores sem fios, os constrangimentos temporais de dados em tempo-real e a limitação dos métodos baseados em *warehousing* motivaram o desenvolvimento de novas técnicas do processamento dos dados em tempo-real para redes de sensores sem fios que minimizem o esgotamento de energia na rede e são adequados para aplicações em tempo-real. As técnicas propostas na presente tese são baseadas na abordagem distribuída e técnicas de modelação estatísticas, na análise das características temporais dos dados e das operações, particularmente a eficácia temporal de dados e a latência de consulta.

Definição do Problema e Objectivos da Investigação

O problema abordado nesta tese é a descrição das diferentes características e a proposta de soluções inovadoras de técnicas de bases de dados em tempo-real nas redes de sensores sem fios. Assim, como já anteriormente mencionado, em aplicações em tempo-real os dados recolhidos pelas redes de sensores sem fios devem refletir de perto o estado actual do ambiente-alvo. No entanto, o ambiente muda constantemente e os dados são recolhidos em momentos discretos de tempo. Como tal, os dados recolhidos têm uma validade temporal e à medida que o tempo avança tornam-se menos precisos, até que cessam de refletir o estado do ambiente [23, 24]. Neste contexto, a latência e eficiência energética tomam uma importância fundamental devido aos requisitos de tempo-real das tarefas e as limitações de recursos de redes de sensores sem fios.

Na maioria das aplicações em tempo-real, os dados recolhidos pelos sensores são enviados periodicamente para uma estação base central onde as transacções do utilizador em tempo-real são encaminhadas e tratadas, de modo que a latência e eficiência energética podem sofrer drasticamente. Outro factor importante é a obrigação de responder a várias aplicações com necessidades diferentes. Na verdade, existem várias aplicações em tempo-real que são geralmente específicas. Por exemplo, para algumas aplicações em tempo-real, a precisão dos resultados pode ser sacrificada para reduzir a latência de consulta. Portanto, a optimização de um novo processamento da consulta para melhorar tanto a latência da consulta individual com dados válidos e o tempo de vida da rede em relação aos vários tipos de aplicações e os níveis de qualidade especificados são muito importantes neste contexto. Uma técnica de processamento pode não ser eficaz para as diferentes aplicações.

No início dos trabalhos de doutoramento, no entanto, notou-se a não existência de uma ferramenta que tenha em conta as características específicas dos modelos de bases de dados em tempo-real para as redes de sensores sem fios. De facto, em comparação com as medições em protótipos, a simulação é o meio menos dispendioso e mais rápido para explorar muitas soluções deste tipo de sistemas complexos. Portanto, neste contexto, a utilização de um simulador para uma fase de validação antes da implementação da aplicação é muito útil. Num trabalho recente, foi proposto um modelo para um cenário de simulação de tempo-real de gestão de dados em redes de sensores sem fios que usa uma abordagem distribuída (Cap. 4). Este modelo utiliza o protocolo *EDF* (*Earliest Deadline First*) [62] para escalonar transacções e as técnicas de *seriação Epsilon* [37] para permitir que as transacções em conflito para executar simultaneamente de modo a que o escalonador não provoque uma imprecisão que é maior do que a aceite nos dados. O modelo foi implementado utilizando o modelo orientado a objectos.

O objectivo principal desta tese é a proposta de uma solução para optimizar o processamento de consultas em tempo-real no contexto de redes de sensores sem fios. Além disso, serão propostas avaliações do desempenho de técnicas de bases de dados em tempo-real que satisfaçam os requisitos de aplicações em tempo-real com base em redes de sensores sem fios. A nova *framework* deverá permitir oferecer uma ferramenta eficaz de decisão para vários tipos de aplicações de tempo crítico, enquanto a solução de optimização de processamento de consultas proposta deve optimizar o processamento de consultas de utilizador em tempo-real para fornecer tanto processamento de dados em tempo-real como economia de energia. Para atingir os principais objectivos desta tese, os seguintes objectivos intermédios foram definidos:

1. Um dos objectivos desta tese é efectuar a revisão do estado da arte das diversas técnicas de gestão de bases de dados em tempo-real em redes de sensores sem fios, bem como em redes tradicionais. Os trabalhos que expõem as várias técnicas são explorados para a identificação das suas vantagens, limitações e desafios a fim de escolher um conjunto das melhores soluções sugeridas ou fornecer uma nova proposta.
2. O segundo objectivo intermédio é a análise de diferentes técnicas de gestão de bases de dados distribuídas para as redes de sensores sem fios que irão ajudar as propostas. Na verdade, com base na análise da literatura e algumas experiências, em redes de sensores sem fios a técnica de *warehousing* é a mais utilizada para a gestão de dados do sensor, porém a técnica distribuída é a técnica mais eficiente em termos de energia e apropriada em tempo-real.
3. Como não existe uma ferramenta para testar e validar técnicas de bases de dados em tempo-real para as redes de sensores sem fios, um dos objectivos intermédios é propor e construir um novo modelo de gestão de bases de dados em tempo-real para as redes de sensores sem fios. A avaliação do desempenho e qualidade de serviço do modelo são demonstrados.
4. Devido à exigência de tempo-real dos dados e tarefas em aplicações em tempo-real e as limitações de recursos das redes de sensores sem fios, para atingir o objectivo principal é necessário propor uma nova técnica de processamento de consultas em tempo-real, que optimiza a latência e o consumo de energia.

Argumento da Tese

Esta tese propõe uma nova abordagem baseada em técnicas de modelação estatística e uma abordagem distribuída para optimizar o processamento de consultas em tempo-real por forma a minimizar a latência e energia. Além disso, é também proposto um novo modelo para um ambiente de simulação de gestão de dados em redes de sensores sem fios em tempo-real que usa uma abordagem distribuída.

O argumento da tese é o seguinte:

Além das restrições subjacentes à gestão de dados em tempo-real para redes de sensores sem fios, existe a necessidade de criar novas abordagens que economizem energia em cada nó sensor, mantendo um bom desempenho ao nível dos tempos de resposta (incluindo a latência) e apresentação de dados em tempo oportuno. Entretanto, o tratamento dos dados off-line *seguido pela abordagem* Warehousing *incrementa o dispêndio de energia podendo também envolver um elevado tempo de espera para receber os resultados, o que degrada a validade dos dados. Além disso, o uso da validade temporal dos dados a serem acedidos para abortar ou suspender as transacções nos primeiros passos de alguns algoritmos propostos pode levar à diminuição de desempenho quando não há alterações de dados que justifiquem tais operações. Neste sentido, segue-se uma abordagem distribuída combinada com técnicas de modelação estatística com o objectivo de melhorar a longevidade da rede, a latência de consulta de dados, bem como a validade dos dados em sistemas de tempo-real.*

Para apoiar este argumento da tese, foi adoptada a seguinte abordagem de investigação.

O problema e o tema de investigação são estudados partindo da revisão da literatura sobre várias técnicas de gestão de bases de dados em tempo-real em redes de sensores sem fios e em redes tradicionais. As vantagens e limitações de cada técnica proposta na literatura são analisadas, bem como as características e restrições das duas tecnologias subjacentes, as bases de dados em tempo real e as redes de sensores sem fios.

As técnicas de bases de dados em tempo-real tradicionais não são adequadas para redes de sensores e necessitam de ser adaptadas para ter em conta as restrições de recursos das redes de sensores sem fios. Na realidade, as redes de sensores sem fios têm recursos limitados, particularmente em termos de armazenamento e energia, e o tempo de vida da rede de sensores depende da energia disponível nos nós que compõem a rede. A optimização do o consumo de energia deve ser um dos principais focos de atenção em trabalhos relacionados com as redes de sensores sem fios. Assim, a literatura e as experiências analisadas revelam que uma das melhores formas de economizar energia na gestão de dados em redes de sensores sem fios é minimizar a atividade dos nós sensores, principalmente as tarefas de comunicação de dados que levam ao maior consumo de energia na rede. Isto pode ser conseguido com uma abordagem distribuída, em vez da abordagem de *warehousing*. As observações feitas na literatura sobre a abordagem distribuída demonstram que é possível realizar o processamento de consulta em rede combinando com técnicas de redução de dados, tais como a agregação de dados, fusão de pacotes, técnicas de compressão de dados, fusão de dados, e as técnicas baseadas em aproximação, que reduzem o tamanho dos dados transmitidos e evitam enviar periodicamente os dados a partir dos nós de rede para a estação base. Além disso, o processamento de consultas em rede oferece o processamento de consultas praticamente em tempo-real, o que faz com que o processamento de consultas dentro dos próprios dispositivos faça com que os dados mais actuais sejam adquiridos.

A literatura sobre bases de dados em tempo-real revela que o principal propósito de tais sistemas é processar transacções em tempo útil, mantendo ao mesmo tempo a consistência lógica e temporal dos dados. De facto, em aplicações em tempo-real os dados recolhidos pelas redes de sensores sem fios devem refletir tanto quanto possível o estado actual do ambiente-alvo, embora esses dados tenham uma validade temporal e, à medida que o tempo avança, se tornem menos precisos, até que deixem de refletir o estado do ambiente. Neste contexto, minimizar a latência de consulta com dados válidos devolvidos é fundamental. Além disso, a análise de aplicações em tempo-real revela que para algumas aplicações a precisão dos resultados pode ser sacrificada para reduzir a latência [11]. Dessa forma, algumas aplicações podem tolerar a leitura de dados obsoletos até certo ponto. Esta observação permite a realização de estimativas de consultas que utilizam técnicas de modelação estatística que fornecem boa latência de consulta com dados válidos com alguma incerteza que o utilizador/aplicação está disposto a tolerar. Assim, a combinação das técnicas de modelação estatística com a abordagem distribuida permite propor uma nova arquitectura e um algoritmo de processamento de consultas para optimizar o processamento em tempo-real de consultas pelos utilizadores minimizando tanto a latência como o consumo de energia com dados válidos.

Os trabalhos de investigação realizados sobre técnicas de gestão de bases de dados em tempo-real mostram que não existe uma ferramenta para testar e validar técnicas de bases de dados em tempo-real para as redes de sensores sem fios, o que motiva a investigação sobre como modelar e contruir um sistema de simulação para gestão de bases de dados em tempo-real sobre as redes de sensores sem fios. Isto revela que o modelo orientado a objectos é mais adequado que o modelo relacional para modelar dados em tempo-real, devido à natureza de diversas aplicações em tempo-real que lidam com objectos complexos do mundo real, tendo em conta, também,limitações de tempo. Assim, o simulador proposto, e tendo em conta uma abordagem distribuída destacada em investigações anteriores, utiliza modelação orientada a objectos e é implementado em Java. Para demonstrar a validade do modelo foi elaborado um estudo de caso que mostra a execução de transacções em tempo-real e a energia consumida na rede.

Principais Contribuições

Esta secção descreve resumidamente as principais contribuições científicas para o avanço do estado da arte resultantes do trabalho apresentado nesta tese.

A primeira contribuição desta tese é uma descrição detalhada das principais características da gestão de bases de dados em tempo-real em redes de sensores sem fios, bem como nas redes tradicionais. Destacam-se as restrições de concepção na gestão de bases de dados em tempo-real sobre as redes de sensores sem fios e oferece-se uma análise abrangente e crítica das soluções actuais propostas de técnicas de bases de dados em tempo-real sobre as redes de sensores sem fios.

Este estudo é descrito no capítulo 2, que consiste num artigo publicado na revista *Journal of Network and Computer Applications* (JNCA), da Elsevier [63].

A segunda contribuição é um estudo detalhado sobre técnicas de gestão de bases de dados distribuídas para as redes de sensores sem fios. Este trabalho teve como objectivo mostrar como as técnicas de bases de dados distribuídas são adaptadas para redes de sensores sem fios com o objectivo de melhorar a gestão da grande quantidade de dados de sensores de uma forma eficiente, em termos de energia, apresentando e classificando as mais recentes e relevantes propostas de gestão de bases de dados distribuídas nas redes de sensores sem fios. Esta contribuição foi muito útil para as principais contribuições desta tese, que se baseiam numa abordagem distribuída. Além disso, esta contribuição apresenta uma análise abrangente e problemas em aberto para facilitar contribuições posteriores. Este estudo está descrito no capítulo 3, que consiste num artigo aceite para publicação na revista *IEEE Transactions on Parallel and Distributed Systems* (TPDS) [64].

A terceira contribuição consiste na proposta de um sistema de simulação para bases de dados em tempo-real sobre as redes de sensores sem fios. O modelo do simulador é baseado na abordagem distribuída e utiliza o *EDF* para escalonador as transacções e as técnicas de *seriação Epsilon* para permitir que as transacções em conflito possam executar simultaneamente de modo que a sua programação não provoque uma imprecisão que é maior do que a aceite nos dados. O modelo foi implementado utilizando o modelo orientado a objectos. Esta ferramenta, programada em Java, foi construída com o intuito de testar e validar as técnicas de bases de dados em tempo-real para as redes de sensores sem fios. Pode ajudar os profissionais e investigadores para conhecer as diferentes características e os condicionalismos a ter em conta na gestão de bases de dados em tempo-real sobre as redes de sensores sem fios. Este simulador permite aos investigadores também saber quais os componentes que participam na concepção de um modelo de arquitectura para técnicas de gestão de bases de dados em tempo-real para redes de sensores, como construir e facilmente implementar um modelo de arquitetura. Ao usar este simulador, os profissionais e investigadores podem simular um protocolo de base de dados em tempo-real para as redes de sensores sem fios e aproximadamente testar a validade temporal e lógica de dados e transacções antes da implementação com medidas reais. Por último, os investigadores e desenvolvedores podem melhorar o simulador, adicionando a simulação de protocolos ou migrando parte do simulador para outro simulador existente. Este trabalho é descrito no capítulo 4, que consiste num artigo publicado na revista *Journal of Network and Computer Applications* (JNCA), da Elsevier [65].

A quarta contribuição desta tese é a proposta de uma nova optimização de processamento de consultas em tempo-real para as redes de sensores sem fios. Esta proposta combina técnicas de modelação estatística com a abordagem distribuída para fornecer uma nova arquitectura e um algoritmo de processamento de consultas para optimizar o processamento em tempo-real da consultas do utilizador, por forma a minimizar a latência e o consumo de energia com dados válidos. Estes dados válidos está manchados com alguma incerteza (ε) que o utilizador/aplicação está disposto a tolerar. Esta arquitectura proposta e algoritmo de processamento de consultas estão descritas no capítulo 5, que consiste num artigo submetido para publicação numa revista internacional [66].

A quinta e última contribuição desta tese é a extensão e adaptação do trabalho anterior para as *WBANs* (*wireless body area networks*), a fim de propor uma nova optimização de processamento de consultas em tempo real para as WBANs baseadas na nuvem (*cloud*). Esta proposta consiste numa nova arquitectura de WBANs baseada na nuvem com o seu algoritmo de processamento de consultas de base para o armazenamento seguro e eficaz, eficiente em termos de energia e processamento de dados em tempo-real. Assim, esta abordagem combina os serviços de uma WBAN baseados em nuvem com técnicas de modelação estatística para optimizar tanto o acesso à infra-estrutura de armazenamento e processamento de consultas em tempo-real para economizar mais energia, sem descurar as limitações de tempo. Esta arquitectura de WBAN baseada em nuvem e o algoritmo de processamento de consultas são descritos no capítulo 6, que consiste num artigo submetido para publicação numa revista internacional [67].

Conclusões

Esta tese centra-se na avaliação do desempenho da gestão de dados em tempo-real em redes de sensores sem fios. Para tal, é apresentado o trabalho de investigação desenvolvido com o objectivo de oferecer uma nova solução de processamento de consultas que satisfaça os requisitos de aplicações em tempo-real e a limitação de recursos das redes de sensores sem fios. É também apresentada uma nova ferramenta que pode ajudar a simular protocolos de bases de dados em tempo-real para as redes de sensores sem fios para avaliar, o mais próximo da realidade possível, a validade temporal e lógica dos dados e transacções antes da implementação com medidas reais. O trabalho de investigação foi dividido em quatro partes: o estudo das diversas técnicas de gestão de bases de dados em tempo-real em redes de sensores sem fios, bem como em redes tradicionais; a análise dos problemas e das aplicações relevantes como base de apoio a esta problemática; a análise de diferentes técnicas de gestão de bases de dados distribuídas para redes de sensores sem fios; a proposta e construção de um novo modelo de gestão de bases de dados em tempo-real para redes de sensores sem fios; e, finalmente, a proposta de uma nova técnica de processamento de consultas em tempo-real que optimiza a latência e o consumo de energia. Estas quatro etapas de investigação resultaram em contribuições desta tese.

A possibilidade de integração de aplicações em tempo-real em redes de sensores sem fios deu origem a diversos desafios para lidar com o armazenamento de dados e consultas em tempo-real de forma eficiente em termos energéticos. De facto, em aplicações em tempo-real, os dados e as tarefas têm restrições temporais. Além disso, uma vez que nas redes de sensores sem fios a energia é um dos recursos mais cruciais e a interação contínua entre os dispositivos de rede e ambientais resulta em grandes quantidades de dados, é desafiante criar um mecanismo de processamento de consultas que satisfaça tanto as limitações de tempo como as de energia. Por isso, algumas técnicas têm sido propostas pelos investigadores. No entanto, muitas delas são limitadas por causa da arquitectura subjacente utilizada, seja ela *warehousing,* distribuída, ou baseada em algoritmos complexos utilizados que podem ter um impacto negativo em qualquer das restrições temporais ou energéticas.

O objectivo principal desta tese foi propor um novo método de processamento de consultas que satisfaça os requisitos de tempo-real das aplicações (em tempo-real) e a limitação de recursos das redes de sensores sem fios. Os objectivos intermédios foram estabelecidos de forma a dividir o trabalho de investigação necessário para alcançar o objectivo principal. A primeira parte do trabalho de investigação foi descrito no capítulo 2 e consistiu no estudo de várias técnicas de gestão de bases de dados em tempo-real em redes de sensores sem fios, bem como em redes tradicionais e da análise das suas limitações. A revisão da literatura revelou que, após a obtenção da informação, um problema importante de armazenamento de dados e exploração surge, particularmente em sistemas que lidam com restrições de tempo-real.

Na verdade, a construção de mecanismos que satisfaçam os requisitos deste tipo de aplicações é muito difícil. Além disso, as técnicas de gestão de dados utilizadas em bases de dados tradicionais geralmente não são adequadas para redes de sensores devido às suas especificidades. A maioria das propostas baseiam-se na abordagem de *warehousing*, uma abordagem centralizada que requer a actualização periódica da base de dados central, que pode causar, entre outros inconvenientes, atraso no tempo de resposta e um grande desperdício de energia.

Por outro lado, outras propostas seguem uma abordagem distribuída que, com base no seu modo de acesso de dados, aparenta fornecer pouco auxílio quanto às restrições temporais. No entanto, com algum risco pois pode haver uma falha súbita de sensores porque os sensores geralmente operam em ambiente instável. Isto pode levar a um atraso ou ausência de informação que influencia o tempo de análise ou leva mesmo ao bloqueio do sistema.

Outros estudos utilizaram algoritmos complexos, principalmente para satisfazer as limitações temporais, por vezes, com a ausência de conhecimento do mecanismo de gasto de energia. Além disso, nalguns destes algoritmos, a espera ou interrupção das transacções é determinada pela validade temporal dos dados a serem acedidos. No entanto, quando se utilizam restrições temporais directamente, pode-se cometer o erro de esperar ou interromper as transacções demasiadas vezes sem existirem variações que justifiquem estas operações. Isto pode levar à diminuição de desempenho quando se sabe que, com base neste estudo, em sistemas em tempo-real, para algumas aplicações, a precisão dos resultados pode ser sacrificada até certo ponto para reduzir o tempo de resposta.

As diferentes técnicas de gestão de bases de dados distribuídas para as redes de sensores sem fios foram estudadas e exploradas em mais detalhe na segunda parte do trabalho de investigação que foi apresentado no capítulo 3. A revisão da literatura mostrou que esta abordagem é uma alternativa quanto aos inconvenientes da abordagem *warehousing* e motivou as conclusões descritas no trabalho de investigação anterior. Os nós sensores não necessitam de enviar periodicamente os dados recolhidos para a estação base, uma vez que podem permanecer nos nós sensores e algumas consultas são distribuídas e processadas dentro dos nós sensores. Isto reduz a comunicação de dados, minimizando o consumo de energia na rede. Além disso, esta abordagem oferece o processamento de consultas em quase tempo-real, suporte para consultas longas e consultas instantâneas. No entanto, de acordo com o ambiente instável e geralmente operando em condições estremas, podem existir falhas repentinas de sensores, que podem levar à perda de informação ou mesmo ao bloqueio do sistema e influenciam substancialmente a análise de resultados. Além disso, nesta abordagem, o atraso de tempo é dependente da distância e depende da profundidade dos nós que possuem as informações necessárias.

No capítulo 4 propôs-se um sistema de simulação para bases de dados em tempo-real sobre as redes de sensores sem fios. O trabalho de investigação feito nos dois capítulos anteriores, principalmente no capítulo 2, mostra que várias ferramentas de simulação para redes de sensores sem fios têm sido propostas e podem ser divididas em duas classes de acordo com a natureza das restrições especificadas. A primeira classe dos simuladores é orientada à rede e estuda a rede do ponto de vista da comunicação, enquanto que a segunda classe de simuladores é orientada ao nó e enfatiza a função de um único nó com modelos de comunicação simples. No entanto, não existe nenhuma ferramenta específica para testar e validar técnicas de bases de dados em tempo-real para as redes de sensores sem fios.

O modelo do simulador é baseado na arquitectura distribuída tendo em consideração as suas vantagens e usa o protocolo *EDF* para escalonar as transacções e as técnicas de seriação *Epsilon* para permitir que transacções conflituantes executem simultaneamente de modo a que a sua programação não origine uma imprecisão superior à aceite nos dados. O modelo completo foi construído em Java. O modelo orientado a objectos foi escolhido porque, com base no trabalho de investigação anterior, é mais adequado para modelar os objectos complexos do mundo real com limitações de tempo do que o modelo relacional. Depois de executar uma simulação, são mostrados os resultados da execução das transacções em tempo-real, em conformidade com o ficheiro de configuração, a percentagem de energia consumida, assim como a percentagem de energia remanescente. Além disso, o simulador é configurável e os protocolos podem ser alterados conforme necessário.

A quarta parte deste trabalho de investigação, descrita nos capítulo 5 e 6, inclui a proposta de um novo método para optimizar o processamento de consultas em tempo-real sobre redes de sensores sem fios, por forma a minimizar a latência e o consumo energético; e uma nova proposta de optimização do processamento de consultas em tempo-real para *WBANs* baseadas na nuvem. A segunda principal contribuição desta tese foi alcançada através da apresentação de uma nova optimização de processamento de consultas em tempo-real para redes de sensores sem fios. De acordo com o trabalho de investigação anterior, a abordagem distribuída permite realizar o processamento em rede das consultas que diminui as actividades de comunicação de dados que causam um elevado dispêndio de energia na rede. Além disso, suporta o processamento de consultas instantâneas e de longa duração, que são processadas quase em tempo-real. Assim, esta proposta combina técnicas de modelação estatística com a abordagem distribuída para fornecer uma nova arquitectura e um algoritmo de processamento de consultas para optimizar o processamento em tempo-real das consultas do utilizador com dados válidos, minimizando a latência e o consumo energético. Estes dados válidos têm alguma incerteza *(ε)* que o utilizador/aplicação está disposto a tolerar. De facto, o estudo anterior revelou que em sistemas em tempo-real, para algumas aplicações, a precisão dos resultados pode ser sacrificada até certo ponto para reduzir o tempo de resposta. Assim, em vez de enviar periodicamente as leituras do sensor para o servidor da base de dados para processamento *off-line* ou processar a consulta directamente na rede, a abordagem híbrida proposta utiliza técnicas de modelação estatística para realizar um processamento de consulta com base no controlo de admissão que usa a tolerância de erro e o intervalo de confiança como parâmetros de admissão para o sistema. Um novo conceito de rede virtual, composto por sensores lógicos que, por sua vez, são compostos por um modelo probabilístico e memória, é utilizado para aproximar na *gateway* as respostas da consulta de acordo com uma determinada tolerância de erro e intervalo de confiança. Se os dados do sensor dentro da rede virtual não são suficientemente ricos para responder à consulta, o controlador de admissão encaminha a consulta para a rede física. Os resultados experimentais com base no mundo real, bem como em conjuntos de dados sintéticos, mostram que a arquitectura geral proposta prevê, entre outras vantagens, boa latência de consulta individual e dados válidos para aplicações em tempo-real e eficiência energética para as redes de sensores sem fios.

O modelo probabilístico usado no trabalho de investigação anterior é específico para o fenómeno estudado. Porém, a arquitectura e o algoritmo de processamento de consulta é geral. Assim, o último trabalho de investigação desta tese é a extensão e a adaptação do trabalho anterior para *WBANs*. É proposta uma nova arquitectura para *WBANs* baseadas em nuvem, incluindo o algoritmo de processamento de consultas subjacente para o armazenamento seguro e eficiente, o processamento de dados e a eficiência energética e em tempo-real. Em vez de enviar periodicamente as leituras do sensor existente na *WBAN* para o servidor dos profissionais de saúde na nuvem, para processamento *off-line*, envolvendo atraso de tempo e desperdício de energia, a abordagem proposta combina os serviços de uma *WBAN* baseada em nuvem e técnicas de modelação estatística para realizar a optimização de processamento de consulta que usa a tolerância de erro e o intervalo de confiança probabilística como critérios de execução da consulta. Assim, o modelo probabilístico utiliza a informação de dados do paciente armazenada no servidor de dados clínicos na nuvem, para aproximar a resposta da consulta de acordo com uma determinada tolerância de erro e intervalo de confiança. Se as informações de dados do paciente dentro do servidor de base de dados clínica não for suficiente para responder à consulta, o processador de consultas encaminha a consulta para a *WBAN* física. Os resultados experimentais baseados em dados reais, bem como em conjuntos de dados sintéticos, mostram que a arquitectura geral proposta oferece, entre outras vantagens, boa latência de consulta individual e dados válidos para diagnóstico médico em tempo-real e processamento energeticamente eficiente de nós sensores corporais. Desta forma, podemos afirmar que os resultados obtidos são extremamente motivadores para utilização real e novos trabalhos futuros.

Sugestões para Trabalho Futuro

A concepção de um modelo de bases de dados em tempo-real para redes de sensores sem fios é muito complexa devido às suas especificidades. A primeira contribuição desta tese consiste na proposta de um modelo para um sistema de simulação para realizar validações de bases de dados em tempo-real com dados adquiridos a partir da rede de sensores sem fios em estágios iniciais de desenvolvimento de *software*. No entanto, numa rede de sensores sem fios, o acesso ao meio e os ciclos de operação dos nós podem impor um atraso na transmissão de dados. Além disso, este atraso é dependente da distância, ou seja, aumenta a cada "salto" de uma comunicação "multi-salto". No entanto, este modelo baseia-se na abordagem distribuída que envolve o processamento das consultas em rede. Portanto, embora o simulador tenha sido construído principalmente para validar as aplicações com protocolos das bases de dados em tempo-real para redes de sensores sem fios, pode ser melhorado pela adição de simulação de protocolos ou migração de uma parte deste simulador para outro simulador existente com características aceitáveis de protocolos de rede, de disposição, escalonamento de tarefas do sistema operativo, etc, que podem afectar a precisão do simulador.

A construção de uma interface gráfica para o utilizador facilita e acelera a criação da topologia da rede e da composição de módulos básicos. Pode também permitir a visualização rápida dos resultados de simulação e ajudar a rastrear e depurar a simulação em tempo-real. Além disso, ao usar uma interface gráfica, os utilizadores não-especializados podem controlar a simulação mais facilmente. Deste modo, o simulador construído pode ser revisto e melhorado a fim de adicionar uma interface gráfica apropriada.

A segunda maior contribuição desta tese é a proposta de uma nova arquitectura e um algoritmo de processamento de consultas para optimizar o processamento das consultas do utilizador em tempo-real para redes de sensores sem fios reduzindo a latência de consulta e economia de energia. Esta abordagem combina a abordagem distribuída e técnicas de modelação estatística para respostas de consulta aproximadas. A abordagem distribuída envolve processamento em rede que geralmente se baseia noutras técnicas de redução de dados, tais como a agregação de dados, fusão de pacotes, técnicas de compressão de dados, fusão de dados, etc. No entanto, esta proposta não beneficia destas técnicas de redução de dados. Na verdade, as técnicas baseadas em aproximação são úteis em muitas aplicações que não requerem leituras exactas, o que resulta num menor consumo de energia. No entanto, a fim de obter melhores resultados em termos de economia de energia, tanto técnicas de agregação como técnicas de aproximação podem ser usadas em combinação sempre que possível. As redes de sensores sem fios dependem do ambiente específico em que são usadas, deste modo, uma técnica não poderá ser sempre eficiente para qualquer tipo de aplicação. Assim, é desejável ter um sistema de gestão de bases de dados generalizado que permita que as várias aplicações o possam personalizar e adaptar de acordo com as suas necessidades.

Adicionalmente, a correlação é um aspecto importante entre os sensores e a sua vizinhança. Assim, em trabalhos futuros, sugere-se a melhoria da arquitectura proposta e o seu mecanismo de consulta, com o objectivo de ter em conta a correlação entre as leituras do sensor e, eventualmente, outros parâmetros, podem ser investigados, melhorando assim ainda mais a estimativa de leituras e a economia de energia.

Referências

[1] P. K. Eswaran et al., “The notions of consistency and predicate locks in a database system,” *Communications of the ACM*, v. 19, n. 11, p. 624–633, November 1976.

[2] K. Ramamritham and P. K. Chrysanthis, “A taxonomy of correctness criteria in database applications,” *The VLDB Journal*, 5(1) :085–097, 1996.

[3] A. Brayner, “Transaction Management in Multidatabase Systems,” *[S.l.]: Shaker Verlag*, 1999. ISBN 3-8265-6142-2.

[4] P., Bernstein, V., Hadzilacos, N. Goodman, “Concurrency control and recovery in database systems,” *Addison-Wesley*, 1987.

[5] D. J. Ullman, “Principles of Database Systems,” Computer Science Press, Inc., Maryland, 1980.

[6] S. C. Date, “An Introduction to Database Systems,” *Addison-Wesley*, 1985.

[7] E. Bouazizi, B. Sadez, C. Duvallet, “Applicabilité du critère d'epsilon-sérialisabilité dans les SGBD temps-réel,” *Special issue ISDM MCSEA*. 2004.

[8] J. Gray, “The Transaction Concept: Virtues and Limitations,” (Invited Paper). *VLDB* 1981, 144-154.

[9] R. Elmasri, S. Navathe, “Fundamentals of Database Systems,” *2nd Edition, New York: Addison-Wesley Pub. Co.*, 1994.

[10] Jane W. S. Liu, “Real-Time Systems,” *PrenticeHall*, 2000, ISBN 0130996513.

[11] S. Pailler, “Analyse Hors Ligne d’Ordonnançabilité d’ Applications Temps Réel comportant des Tâches Conditionnelles et Sporadiques,” *PhD Thesis*, 2006, 21-30.

[12] C. Hanen and A. Munier, “Cyclic scheduling on parallel processors : An Overview,” *volume Scheduling theory and its applications, P. Chretienne et al., Chap 9. John Wiley & Sons*, 1995.

[13] C.L. Liu and J.W. Layland, “Scheduling algorithms for mutltiprogramming in real-time environnement,” *Journal of the ACM*, 20(1) :46–61, 1973.

[14] P. Puschner and R. Nossal, “Testing the results of static worst case execution time analysis,” *In IEEE Real Time Systems Symposium*, dec 1998.

[15] E. Klingerman and A.D. Stoyenko, “Real time euclid : a language for reliable real time systems,” *IEEE Trans. softw. Eng.*, 12(9) :941–949, 1986.

[16] P. Puschner and C. Koza, “Calculating the maximum execution time of real-time programs,” *Journal of Real-Time Systems*, 1(2) :159–176, 1989.

[17] J. Gustafsson, B. Lisper, N. Bermudo, C. Sandberg, and L. Sjöberg, “A prototype tool for flow analysis of c program,” *In 2nd Intl. Workshop on WCET Analysis 14th Euromicro Conference on Real-Time Systems*, June 2002.

[18] C. Healy, M. Sjödin, V. Rustagi, and D. Whalley, “Bounding loop iterations for timing analysis,” *In 4th IEEE Real-Time Technology and Applications Symposium (RTAS’ 98)*,Discrete Event Systems: Analysis and Control,pages 12–21. IEEE, 1998.

[19] T. Mitra and A. Roychoudhury, “A framework to model branch prediction for wcet analysis,” *In 2nd Intl. Workshop on WCET Analysis 14th Euromicro Conference on Real-Time Systems*, June 2002.

[20] C. Rochange and P. Sainrat, “Difficulties in computing the wcet for processors with speculative execution,” *In 2nd Intl. Workshop on WCET Analysis 14th Euromicro Conference on Real-Time Systems*, June 2002.

[21] A. Colin and I. Puaut, “Worst case execution time analysis for a processor with branch prediction,” *Real-Time Systems*, 18(2): 249–274, May 2000.

[22] M. Hamdaoui and P. Ramanathan. “*A dynamic priority assignment technique for streams with*, 1995.

[23] N. Idoudi, C. Duvallet, B. Sadeg, N. Louati, R. Bouaziz, F. Gargouri, "How to model a real-time database?," *Proceedings of 12th IEEE International Symposium on Object-oriented Real-time distributed Computing (IEEE ISORC'2009)*. Tokyo, Japan: IEEE, 2009a, pp. 321-328.

[24] N. Idoudi, N. Louati, C. Duvallet, R. Bouaziz, B. Sadeg, F. Gargouri, "A Framework to Model Real-Time Databases," *International Journal of Computing & Information Sciences* Vol. 7; January 2009b, pp.1 – 11.

[25] L. C. DiPippo, V.F. Wolfe, "Real-time databases," *Journal of Database Systems Hand-book, Multiscience Press (Citeseer)*, 1997.

[26] S.H. Son and J.W.S. Liu, "Maintaining temporal consistency: pessimistic vs. optimistic concurrency control," *IEEE Transactions on Knowledge and Data Engineering*, vol. 7, n° 5, 1995.

[27] M. Xiong, J.A. Stankovic, K. Ramamritham, D. Towsley and R.M. Sivasankaran, "Maintaining temporal consistency: issues and algorithms," *1st Int. Workshop on RTDBS*. Issues and Applications, p. 1-6, California 1996.

[28] S.H. Son and J.W.S.Liu, "How well can data temporal consistency be maintained ?," *IEEE Symp. on Computer-Aided Control System Design*, 1992.

[29] K. Ramamritham, "Real-time databases," *Journal of Distributed and Parallel Databases (Springer)* 1993; 1(2): 199-226.

[30] Chagas Lunas Dantas, Lima Eduardo Pereira, Ribeiro Neto Pedro Fernandes, "Real-Time Databases Techniques in Wireless Sensor Networks," *Sixth International Conference on Networking and Services, IEEE*, 2010.p. 182–188.

[31] J.A. Stankovic, M. Spuri, M. Di Natale and G. C. Buttazzo, "Implications of classical scheduling results for real-time systems," *IEEE Computer*, vol. 28, n° 8, p. 16-25, 1995.

[32] J. Lindström, "Integrated and adaptive optimistic concurrency control method for real-time databases," *8th International Conference on Real-Time Computing Systems and Applications (RTCSA'2002)*, University of Helsinki Finland, p. 143–151, 2002.

[33] J. Lindström, "Optimistic Concurrency Control Methods for Real-Time Database," *PhD Thesis*, Department of Computer Science, University of Helsinki Finland, November 2002.

[34] K. Raatikainen, J. Lindström, "Using Real-Time Serializability and Optimistic Concurrency Control in Firm Real-Time Databases," University of Helsinki Finland, March 2002.

[35] D. Nyström et al., "Pessimistic concurrency-control and versioning to support database pointers in real-time databases," *In: Proceedings of the 16th Euromicro Conference on Real-Time Systems (ECRTS04)*. Sicily, Italy: IEEE Computer Society Press, 2004.

[36] L. C. DiPippo, "Semantic real-time object-based concurrency control," *PhD Thesis*, Department of Computer Science and Statistics, University of Rhode Island, 1995.

[37] C. Pu, "Generalized Transactions Processing with Epsilon Serializability," *In Proceedings of 4th International Workshop on High Performance Transactions Systems*, Asimolar, California, 1991.

[38] J.R. Haritsa, M.J. Carey and M. Livny, "Data access scheduling in firm real-time database systems," *J. of RTS*, vol.4, n° 3, p. 203-241, 1992.

[39] B. Purimetla, R. M. Sivasankaran, K. Ramamritham and J. A. Stankovic, "Real- time databases: issues and applications," *Principles of RTS*, ed. Printice Hill, 1994.

[40] G. Noël, "Indexation dans les bases de données capteurs temps réel," *PhD Thesis*, National Institute of Applied Sciences Of Lyon, 2006.

[41] C. Duvallet, Z. Mammeri, B. Sadeg, "Les SGBD temps réel," *In Technique et Science Informatiques, Hermes ed.*, 479-

516, June 1999.

[42] W. Kim, "Object-Oriented Database Systems: Promises, Reality, and Future," *Modern Database Systems*, pp 255-280. Addison-Wesly, 1995.

[43] F. Terrier, A. Lanusse, D. Bras, P. Roux, P. Vanuxeem, "Concurrent object for multitasking," *In L'objet*, v3. pp 179-196, Paris, 1997.

[44] V. F. Wolfe, J. J. Prichard, L. C. Dipippo, J. Black, "Real-Time Database System: Issues and Applications," *chapter The RTSORAC Real-Time-Object-Oriented DataBase Prototype*, pp 279-301. Kluwer Academic Publlishers, 1997.

[45] L. M. L. Oliveira and J. J. P. C. Rodrigues, "Wireless Sensor Networks: a Survey on Environmental Monitoring," *Journal of Communications (JCM)*, Vol 6, No 2, pp. 143-151, April 2011, DOI: 10.4304/jcm.6.2.143-151

[46] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, 38, 393-422, 2002.

[47] F. I. Akyildiz, T. Melodia, and R. K. Chowdhury. "A survey on wireless multimedia sensor networks," *Journal of Computer networks (Elsevier)* 51, no. 4 (March 2007): 921--960.

[48] L.M. Luís Oliveira, F. de Sousa Amaro, and Joel J. P. C. Rodrigues. "Routing and Mobility Approaches in IPv6 over LoWPAN Mesh Networks," *in International Journal of Communication Systems, Wiley*, ISSN: 1074-5351, Vol. 24, Issue 11, November 2011a, pp. 1445-1466, DOI: 10.1002/dac.1228.

[49] K. Lin, M. Chen, S. Zeadally, and J. J. P. C. Rodrigues, "Balancing Energy Consumption with Mobile Agents in Wireless Sensor Networks," *Future Generation Computer Systems*, Vol. 28, Issue 2, pp. 446-456, Feb. 2012, DOI: 10.1016/j.future.2011.03.001,

[50] K. Lin, J. J. P. C. Rodrigues, H. Ge, N. Xiong, and X. Lian "Energy Efficiency QoS Assurance Routing in Wireless Multimedia Sensor Networks," *IEEE Systems Journal*, Vol. 5, Issue 4, pp. 495-505, December 2011, DOI: 10.1109/JSYST.2011.2165599

[51] S. Cho, E. Shih, N. Ickes, and R. Min. "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," *ACM/IEEE MOBICOM. ACM*, 2001. 272– 287.

[52] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TinyDB: An Aquisitional Query Processing System for Sensor Networks," *ACM Transactions on Database Systems (TODS) (ACM)* 30, no. 1 (2005): 122-173.

[53] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG:Atiny aggregation service for ad-hoc sensor networks," *In OSDI.Citeseer*, 2002.

[54] F.A. Silva, T.R.M. Braga, L.B. Ruiz, J.M.S. Nogueira, "Tecnologia de Nós Sensores Sem Fio," *Contrôle & Instrumentação*; agosto 2004;92:76–87.

[55] P. Bonnet, P. Seshadri, "Device Database Systems," *In: Proceedings of the International Conference on Data Engineering ICDE'99*. San Diego, CA: *IEEE Computer Society*; 2000.

[56] P. Bonnet, J. Gehrke, P. Seshadri, "Querying the Physical World," *Journal of Personal Communications, IEEE* 2000;7(5):10–5.

[57] S. Servigne, G. Noel, "Real time and spatiotemporal data indexing for sensor based databases," *Journal of Geospatial Information Technology for Emergency Response (Taylor & Francis Group)* 2008 ; 6: 123-142.

[58] Y. Yao and J. Gehrke, "Query Processing in Sensor Networks," *In First Biennial Conference on Innovative Data Systems Research (CIDR 2003)*, Asilomar, California, January 5-8, 2003.

[60] Neto Pedro Fernandes R, Perkusich Maria LB, Perkusich Angelo, "Real-Time Database for Sensor Networks," *In: Proceedings of the 6th International Conference on Entreprise Information Systems*, 2004. p. 599–603.

[61] M. Holanda, A. Brayner, S.V. Fialho, "Controle de concorrência em bancos de dados para ambientes de computação móvel," *VI Workshop de Comunicação sem Fio e Computação Móvel,* p. 161–170, 2004.

[62] C. Liu, J. Leyland, "Scheduling algorithms for multiprogramming in hard real-time environment," *Journal of the ACM,* 20(1), 1973, pp 46-61.

[63] O. Diallo, J. J.P.C. Rodrigues, and M. Sene, "Real-time Data Management on Wireless Sensor Network: a Survey," *in Journal of Network and Computer Applications, Elsevier*, ISSN: 1084-8045, Vol. 35, Issue 3, May 2012, pp. 1013-1021, DOI: 10.1016/j.jnca.2011.12.006.

[64] O. Diallo, J. J.P.C. Rodrigues, M. Sene, and J. Lloret, "Distributed Database Management Techniques for Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, ISSN: 1045-9219, 19 Aug. 2013, IEEE Computer Society, DOI: 10.1109/TPDS.2013.207 (in press).

[65] O. Diallo, J. J.P.C. Rodrigues, M. Sene, and J. Lloret, "Simulation Framework for Real-Time Database on WSNs," *in Journal of Network and Computer Applications, Elsevier*, ISSN: 1084-8045, Vol. 39, March 2014, pp. 191-201, DOI: 10.1016/j.jnca.2013.07.001.

[66] O. Diallo, J. J.P.C. Rodrigues, M. Sene, and F. Xia, "Real-Time Query Processing Optimization for Wireless Sensor Networks," *International Journal of Sensor Networks (IJSNet), InderScience Publishers*, ISSN (Online): 1748-1287 - ISSN (Print): 1748-1279 (in press).

[67] O. Diallo, J. J.P.C. Rodrigues, M. Sene, J. Niu, "Real-Time Query Processing Optimization for Cloud-based Wireless Body Area Networks," *in Information Sciences, Elsevier*, ISSN: 0020-0255, doi: http://dx.doi.org/10.1016/j.ins.2014.03.081 (in press).

Abstract

Technological advances in recent years have allowed the maturity of Wireless Sensor Networks (WSNs), which aim at performing environmental monitoring and data collection. This sort of network is composed of hundreds, thousands or probably even millions of tiny smart computers known as wireless sensor nodes, which may be battery powered, equipped with sensors, a radio transceiver, a Central Processing Unit (CPU) and some memory. However due to the small size and the requirements of low-cost nodes, these sensor node resources such as processing power, storage and especially energy are very limited.

Once the sensors perform their measurements from the environment, the problem of data storing and querying arises. In fact, the sensors have restricted storage capacity and the on-going interaction between sensors and environment results huge amounts of data. Techniques for data storage and query in WSN can be based on either external storage or local storage. The external storage, called warehousing approach, is a centralized system on which the data gathered by the sensors are periodically sent to a central database server where user queries are processed. The local storage, in the other hand called distributed approach, exploits the capabilities of sensors calculation and the sensors act as local databases. The data is stored in a central database server and in the devices themselves, enabling one to query both.

The WSNs are used in a wide variety of applications, which may perform certain operations on collected sensor data. However, for certain applications, such as real-time applications, the sensor data must closely reflect the current state of the targeted environment. However, the environment changes constantly and the data is collected in discreet moments of time. As such, the collected data has a temporal validity, and as time advances, it becomes less accurate, until it does not reflect the state of the environment any longer. Thus, these applications must query and analyze the data in a bounded time in order to make decisions and to react efficiently, such as industrial automation, aviation, sensors network, and so on. In this context, the design of efficient real-time data management solutions is necessary to deal with both time constraints and energy consumption.

This thesis studies the real-time data management techniques for WSNs. It particularly it focuses on the study of the challenges in handling real-time data storage and query for WSNs and on the efficient real-time data management solutions for WSNs.

First, the main specifications of real-time data management are identified and the available real-time data management solutions for WSNs in the literature are presented. Secondly, in order to provide an energy-efficient real-time data management solution, the techniques used to manage data and queries in WSNs based on the distributed paradigm are deeply studied. In fact, many research works argue that the distributed approach is the most energy-efficient way of managing data and queries in WSNs, instead of performing the warehousing. In addition, this approach can

provide quasi real-time query processing because the most current data will be retrieved from the network.

Thirdly, based on these two studies and considering the complexity of developing, testing, and debugging this kind of complex system, a model for a simulation framework of the real-time databases management on WSN that uses a distributed approach and its implementation are proposed. This will help to explore various solutions of real-time database techniques on WSNs before deployment for economizing money and time. Moreover, one may improve the proposed model by adding the simulation of protocols or place part of this simulator on another available simulator. For validating the model, a case study considering real-time constraints as well as energy constraints is discussed.

Fourth, a new architecture that combines statistical modeling techniques with the distributed approach and a query processing algorithm to optimize the real-time user query processing are proposed. This combination allows performing a query processing algorithm based on admission control that uses the error tolerance and the probabilistic confidence interval as admission parameters. The experiments based on real world data sets as well as synthetic data sets demonstrate that the proposed solution optimizes the real-time query processing to save more energy while meeting low latency.

Keywords: *Cloud computing; Data Collection; Data Reduction Techniques; Distributed Database Management; Distributed Storage; Model; Performance Evaluation; Query Estimation; Query Optimization; Query Techniques; Real-Time Database Management Techniques; Simulator; Warehousing; Wireless Body Area Networks; Wireless Sensor Networks.*

Contents

Chapter 3

Chapter 4

List of Figures

Chapter 5

Real-Time Query Processing Optimization for Wireless Sensor Networks

Chapter 6

Real-Time Query Processing Optimization for Cloud-based Wireless Body Area Networks

List of Tables

Acronyms

2PL	Two Phase Locking
AC	Admission Controller
ACID	Atomicity, Coherency, Isolation, Durability
ADAGA	Adaptive AGgregation Algorithm for sensor networks
ADC	Analog-to-Digital Converter
AP	Access Point
API	Application Programming Interface
AQUA	Approximate Query Answering
ARS	Action and Relay Stations
ATEMU	ATmel EMUlator
AVI	Absolute Validity Interval
BB	Boundary-Based
BBQ	Barbie-Q
BCH	Backup Cluster Head
BM	Backbone Mapping
CBC	Clustering, Balancing, and Compression algorithm
CC	Concurrency Control
CEC	Clustering, Expanding, and Compression algorithm
CF	Compatibility Function
CH	Cluster-Head
CKN	Connected K-Neighborhood
CPU	Central Processing Unit
CQL	Continuous Query Language
DB	Database
DBMS	Databases Management Systems
DBW	Database Warehousing
DC	Divergence Control Algorithms
DC-WSNs	Duty-Cycled Wireless Sensor Networks
DDCRS	Dynamic Data-Centric Routing and Storage
DE	Data Error
DISC	Distributed Information Storage and Collection
DOSA	Distributed and self-Organizing Scheduling Algorithm
DR	Data Replication

DRA	Data Replication Algorithm
DSMS	Data Stream Management System
ECG	Electrocardiogram
EDF	Earliest Deadline First
EHRs	Cloud-based of Electronic Health Records
ENERGY	Energy Efficient Rate Governed Yardstick
ESR	Epsilon-Serializability
ESPDA	Energy-Efficient and Secure Pattern-based Data Aggregation
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSN	Global Sensor Network
GUI	Graphical User Interface
HDRA	Hierarchical Data Replication Algorithm
IP	Internet Protocol
ISO	International Organization for Standardization
J-Sim	Java-based Simulator
SL	Logical Sensors
MAC	Medium Access Control
NSB	Necessary Set-Based
OOB	Obtaining Optimal Backbones
PC	Personal Computer
PCH	Primary Cluster Head
PDA	Personal Digital Assistant
PDF	Probability Density Function
PDT	Pocket Driven Trajectories
PS	Personal Server
QL-RTDB	Query Language for Real-Time Databases
QMF	QoS Management architecture for deadline Miss ratio and data Freshness
QoD	Quality of Data
QoS	Quality of Service
RAM	Random Access Memory
RF	Radio Frequency
RT-DBMS	Real-Time Databases Management Systems
RTP	Real-Time Protocol
RTSORAC	Real-Time Semantic Objects Relationships And Constraints
RTSTREAM	
SDB	Sensor Database
SDS	Similarity Data Storage
SENS	Sensor Environment and Network Simulator

SIA	Secure Information Aggregation
SNEE	Streaming NEtwork Engine
SNQL	Sensor Network Query Language
SQL	Structured Query Language
SR	Serializability
SSB	Sufficient Set-Based
SSDQP	Sun SPOT Distributed Query Processing
StH	Spatiotemporal/Heat
SWN	Stochastic Well-formed Petri Nets
TCP/IP	Transmission Control Protocol/Internet Protocol
TiNA	Temporal Coherency-Aware In-Network Aggregation
UML	Unified Modelling Language
ViFuR-ASN	Virtual Full Replication with Adaptive Segmentation for sensor Networks
WBANs	Wireless Body Area Networks
WBASN	Wireless Body Area Sensor Network
WCE	Worst Case Execution time
WLAN	Wireless Local Area Network
WSNs	Wireless Sensor Networks
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol

Chapter 1

Introduction

This thesis addresses the subject of performance assessment of real-time data management on wireless sensor networks. In fact nowadays, systems based on sensor networks are getting increasingly used in many areas of the knowledge, giving rise to several flavours of applications such as financial market, human motion tracking application, monitoring of urban or environmental phenomena, patients monitoring in hospitals, automated production, military and aircraft control, etc. Some of these applications called real-time applications have the particularity of having to comply with the logical constraints and consistency imposed by the system, but also the temporal constraints related to the speed of execution of operations, as well as the respect of their deadlines. In addition, these applications must be able to handle large amounts of data, coming from sensors, necessary for their correct functioning. Thus, the use of databases is necessary and indispensable for this type of systems. However, unlike traditional databases, real-time databases must be able to also meet temporal constraints introduced by real-time systems, while ensuring the integrity constraints and consistency, the ability to share data, the recoveries after failures, etc., provided by the traditional databases management systems (*DBMS*). Thus, the real-time databases are essential for real-time systems with non-negotiable temporal constraints, such as automotive and aircraft applications where deadlines on temporal data and transactions can not be lost because of the risk of generating a disaster. Similarly, the real-time databases are useful for real-time systems running in unpredictable environments, such as financial market and human motion tracking application, where meeting most of the temporal constraints is the best system performance.

The design of effective solutions for real-time data management for WSNs is at the centre of this thesis and is particularly difficult in the sense that these kinds of systems involve two areas quite complex those of real-time databases and WSNs. However, solutions that rely on either a distributed or centralized approach or specific algorithms with formal models to define levels of accuracy and performance that must be met have been proposed.

This chapter describes in detail the focus and scope of the thesis, defines the addressed problem and objectives. Afterwards, the thesis statement and the main contributions are presented. Finally, the chapter ends by the description of the manuscript organization.

1. Thesis Focus and Scope

Database Systems

A database system is composed of a set of structured data named *database* and a software system named *databases management system* (*DBMS*). A database is a large set of structured data stored on a permanent medium. The database is an essential component of modern computer systems, for example management information systems (banks, insurance, accounting, inventory management, sales management, production management, etc.), electronic libraries, scientist database, etc. A *DBMS* is a set of programs that allows the management and access to a database. It usually hosts several databases, which are intended to software or different themes.

Multiple users with different views of the data usually access a database. To avoid inconsistencies caused by multiple accesses, a *DBMS* must provide mechanisms to control the simultaneous execution of several programs on the database.

Access to the content of the database is performed through transactions [1]. A transaction, by definition, is a sequence of actions that transform the database from one consistent state to another consistent state. Such actions are mainly writing or reading operations of data called *granules* (tuples, pages, ...) and manipulation operations (calculation, test, ...). A database is said to be consistent if it is correct from the point of view of the user, i.e. it maintains the invariants of the base or integrity constraints. The notion of consistency covers many dimensions as described in [2]. From the point of view of access requests, it is to manage the concurrent execution of multiple transactions without that the updates of a transaction are visible before its validation, one talks about transactional consistency or isolation.

Concurrent executions of transactions in database systems are managed by concurrency control protocols based on the notion of *serialization* that determines which transactions can run concurrently [3-7]. A serializable execution is an interlaced execution of the actions of a set of transactions $\{T_1, T_2, ..., T_n\}$, which gives overall and for each participating transaction the same result as a serial execution of $T_1, T_2,.., T_n$. These protocols ensure the ACID (Atomicity, Coherency, Isolation, Durability) properties [8, 9] of transactions that impose restrictions on the transaction execution order and the data access rights (logical consistency constraints of data and transactions).

Real-Time Systems

In computing, one qualifies a real-time system when the system is able to control a physical process at an appropriate speed to the evolution of the controlled process.

The real-time computing systems differ from other computer systems by taking into account temporal constraints which its compliance is as important as the accuracy of the result, i.e. the system must not only deliver accurate results, it must deliver in the deadlines. The real-time computing systems are now present in many sectors, such as in the manufacturing industry, for example, through process control systems (factories, nuclear centrals), treatment of market data in real-time, in aeronautic through embedded control systems (aircraft, satellites), in the processing and routing of information (video, data, remote control, virtual reality, etc.) [10].

Two distinct elements can be highlighted in real-time systems [11]: one or more physical entities constituting the *process*, whose role is to act and to detect, and a computing control named *controller* or *real-time application* that is the decisions (or reactions) maker of the process. The controller receives information from the environment of the process via sensors and controls the changes of state of the process through *actuators*. Figure 1 gives an overview of the interactions between *process* and *controller* of a real-time system.

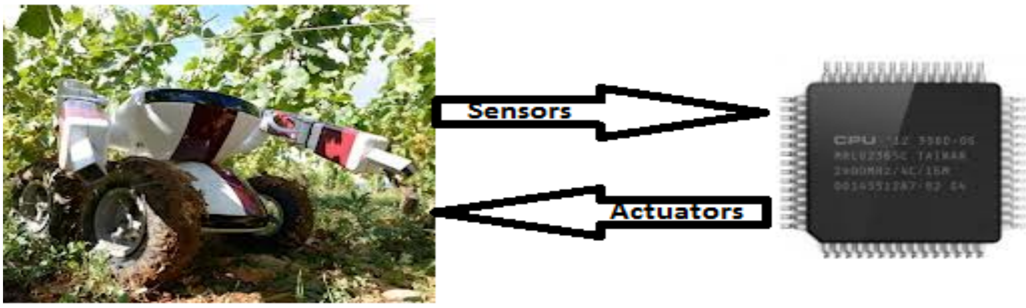


Fig. 1. Interaction process-controller of a real-time system [11].

These sensors and actuators are controlled by *tasks* that are sequential programs dedicated to the treatment of one of the components of the real-time system. For example, a real-time program may consist of a collection of *tasks* such as: periodic execution of measures of different physical quantities (pressure, temperature, acceleration, etc.), treatments at regular intervals or scheduled, etc.

There are three types of real-time tasks that differ in their temporal characteristics [11]: *periodical tasks* that have regular activity, the *aperiodic tasks* are in their side activated randomly depending on for example random event. This type of task has a subfamily of task called *sporadic tasks* for which a minimum period separates two successive occurrences of the triggering event. Finally *cyclic tasks* [12] that are very close to periodic tasks, but at the closer difference that the duration between two successive activations of a periodic task is constant, while it belongs to an interval $[P_{min}, P_{max}]$ for cyclic tasks.

In order to verify that the temporal constraints are met, quantitative temporal indications are introduced in the task models. Thus as an example, given a periodic task θ_i based on the temporal model originally introduced by [13]. These temporal indications are given in the form of *liberation time* (l_i), the *computing time* (C_i), the *deadline* (D_i), and the *periodicity* (P_i).

- The *liberation time* (l_i) represents the instant time on which the first instance of θ_i is activated and is called *Waking date*. When all tasks have the same waking date, so at this moment, there is one instance of each task ready to be run;

- The *computing time* (C_i), also called the worst case execution time (*WCET*) or maximum load of θ_i, represents the execution time required to perform the task θ_i without interruption. This specification of time is crucial to the success of the task scheduler. Determine this worst case execution time is usually very difficult. Indeed, either by dynamic analysis (direct measure of an execution) or static analysis (exploration of the code of the task) [14, 15], its evaluation is made difficult by the presence of conditional instructions, non-deterministic loops [16, 17, 18], or improvements in processors such as speculative execution that also make them indeterminist [19, 20, 21]. Therefore, this parameter is one of the most critical for modelling the tasks system since it is the only a priori an approximate value;
- The *deadline* (D_i) determines the time allocated for each instance of the task θ_i for its execution, i.e. the maximum period of time allowed between the activation and termination of the instance. This parameter is the most interesting in terms of determining the correction and system efficiency.
- Finally, the last parameter, the *periodicity* (P_i), quantifies the minimum time interval between two successive activations of the same task. Figure 2 illustrates the role of each of these above parameters.

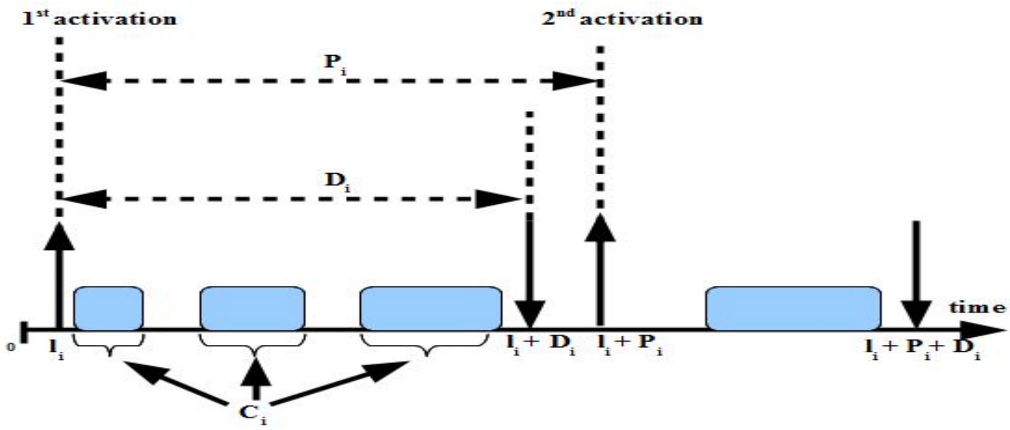


Fig. 2. Modelling of a periodic task θ_i [11].

It is possible to determine the activation dates for each instance of task θ_i: given $j \in N^+$, then the waking date of the j^{th} instance of task θ_i is:

$$l_i^j = l_i + (j-1) * P_i \qquad (1)$$

Similarly, successive dates of deadlines for each instance of a task θ_i are given, with $j \geq 1$, by:

$$d_i^j = l_i^j + D_i \qquad (2)$$

The expected quality of service for the evaluation of real-time applications depends on the violation or not of temporal constraints. Thus, one can classify real-time systems according to the rigidity of temporal constraints imposed on them:

- *Soft real-time systems* are less exacting. Failure to comply with a deadline does not result in system failure. These violations are tolerated, but they will cause disruptions that one has to minimize. In this type of system the measure of effectiveness is generally performed by a statistical analysis of average response times and can be encountered in the multimedia processing systems such as streaming for example.
- *Firm real-time systems* [22] are more restrictive. In this type of system non-compliance with a temporal constraints is to be avoided, since it has more serious consequences for the system. However, it is recognized that such violations can exceptionally happen.
- *Hard real-time systems* are the most severe. Failure of temporal constraints can lead to failures with potentially serious consequences. It is therefore imperative that these constraints are met. These systems are often encountered in the field of embedded systems such as avionics, robotics, etc.
- *Mixed real-time systems* are subject to the requirements of both the systems with hard constraints for certain tasks and those systems with soft constraints for others. These systems include the most current real-time systems.

Characteristics of Real-Time Databases Management Systems

Like a traditional database management system (*DBMS*), a real-time DBMS (*RT-DBMS*) must process transactions and ensure that the logical consistency of the data is not violated. However, unlike a traditional DBMS, a RT-DBMS emphasizes on the temporal validity of the data and the time constraints or deadlines for transactions [23, 24, 25].

Real-Time Data

One of the many issues raised by the design of RT-DBMS is maintaining the consistency of data in the database [26, 27]. Indeed, the current state of the targeted environment as perceived by the controller system must reflect as closely as possible the actual state of that environment. This requirement has an impact on the design of RT-DBMS, which should not only respect the integrity constraints (logical consistency), but also respect the constraints of temporal consistency of data.

Temporal Consistency of Real-Time Data

The temporal consistency can be measured in two ways [28]:

- *Absolute consistency*, which deals with the need to maintain the view representing the state of the targeted environment consistent with the real state of the environment;
- *Relative consistency*, which concerns data derived from other ones.

According to [29], these two concepts for real-time data can be defined as follows: a set of data used to derive a new data item constitutes a *relative consistency* set. Each such set *R* is associated with a relative validity interval denoted by R_{rvi}. Given a real-time data item $d \in R$, d has a correct state if:

1. d_{value} is logical consistency, i.e., satisfies all integrity constraints.
2. d is temporally consistent:
   - ✓ Absolute consistency: $(current\_time - d_{timestamp}) \leq d_{avi}$
   - ✓ Relative consistency: $\forall d' \in R,\ |d_{timestamp} - d'_{timestamp}| \leq R_{rvi},$

Where d_{value} is the actual value of the data d, $d_{timestamp}$ represents the time instant where this value has been measured, and d_{avi} is the absolute validity interval of the value of d.

Representation of Real-Time Data

The real-time data represents the capture of the current state of the targeted environment. This data should reflect the current state of the targeted environment. However, the environment changes constantly and the data are collected in discreet times. So, the collected data have temporal constraints.

To satisfy these temporal constraints, the structure of the data must include these attributes: (*i*) *timestamp*, which indicates the instant when the observation relating the data was made; and (*ii*) *absolute validity interval* (*avi*) that denotes the time interval following the timestamp during which the data are considered valid. This data can be noted as $d = (d_{value}, d_{timestamp}, d_{avi})$. For exemplification purpose, let's consider a system with two data: temperature and pressure with $Temperature_{avi} = 5$, $Pressure_{avi} = 10$, $R = \{temperature, pressure\}$, the *relative validity interval* $R_{rvi} = 2$. If the *current_time* is equal to *100*, then one has with respect to the two conditions in the previous sub-paragraphs:

- ✓ the temperature = (347, 96, 5) and pressure = (50, 97, 10) have temporal consistency.
- ✓ however, the temperature = (347, 96, 5) and pressure = (50, 92, 10) are not temporally consistent (the relative consistency is not met).

For the data quality, another attribute can be considered; the *imprecision* or *data error (DE)*, which refers to how the current state of the targeted environment may differ from the measured data [29]. The *data error* on a data version *d* is defined by:

$$\mathrm{DE}(d) = 100 * \left|\frac{\mathrm{CurrentValue(d)} - \mathrm{UpdateValue(d)}}{\mathrm{CurrentValue(d)}}\right| \% \qquad (3)$$

Real-Time Transactions

The methods used in traditional DBMS are generally not suitable for real-time DBMS. However, they can be used and adapted to take into account the constraints of real-time DBMS which involve requiring that transactions are executed before their deadlines and that these transactions access to valid data with limited validity periods. This has implications for how to design these systems. In particular, the *ACID* properties of transactions of traditional databases were reviewed and adapted to the real-time context [25, 29, 30]. In fact:

- The *atomicity* property, in traditional databases, specifies that a transaction should either complete its execution or is not executed at all. However, this property is relaxed in real-time databases. In fact, if the transaction is composed of sub-transactions, this property is only applied to the sub-transactions that want to deal with completely data consistency;
- The *coherency* property indicates that inconsistent data can be read. As the respect of transaction's deadlines is often more important than the correctness of results, in many situations, correctness can be traded for timeliness;
- The *isolation* property specifies that the operations of a transaction should not be visible to other transactions until it is finished. To meet this property, concurrent transactions must be serialized. However for real-time databases, the concurrent execution of transactions is considered correct if the transactions meet their deadlines with new data. Even if this concurrent execution of transactions does not meet the serialization criteria;
- The concept of *durability* has not the same meaning in the real-time DBMS as in traditional DBMS. In traditional DBMS, this concept means that changes made to a database become permanent (persistent) as soon as the update transactions are validated. However, data in real-time DBMS should reflect the current state of environment, while this state changes constantly. For that, sensors periodically update the content of the database. So, the persistence of such data is verified only during their period of validity.

Concurrency Control Protocols

In traditional DBMS, the serializable schedule is the accepted standard for maintaining database consistency in case of concurrent access by multiple transactions. In other words, if concurrent transactions are serializable, then the database is maintained in a consistent state after the execution of these transactions.

Serializable schedules are difficult to apply in a real-time context. In fact, the criterion of serializability is too strict for the execution of transactions and access to real-time data. Therefore, techniques for concurrency control of transactions developed in traditional DBMS are not directly applicable to real-time DBMS. Indeed, in a real-time database it is necessary to maintain, in addition to the logical consistency of the data, their temporal coherence.

On the other hand, the algorithms designed for scheduling real-time tasks [31] cannot be applied directly to transactions of databases to integrate temporal constraints. Works on the concurrency control of transactions in real-time DBMS rely on traditional techniques of transactions concurrency control (CC) in a DBMS on one hand, and on scheduling techniques of tasks in real-time systems on the other hand. Techniques for concurrency control of transactions in real-time DBMS are based on two policies:

- *Optimistic CC policy* [32, 33, 34] is a technique of type curative. The algorithms of this class allow transactions to concurrently run, taking the risk of having to restart them if inconsistencies appear in the database. This policy is called optimistic because the assumption considered is that there is a low probability that two transactions enter in concurrence on the same data granule (elementary data of a database accessed by a transaction). One allows transactions to run and a control is performed to ensure the serializability at the end of transactions. In this case, if two transactions are concurrent on an object, one of them will be aborted and restarted. For this protocol there is no indeterminate of the waiting time, but the number of transactions that can be restarted is big, this can be a negative factor in systems with time constraints.
- *Pessimistic CC policy* is a technique of control of concurrent access consisting of locking objects as and when access by a transaction and to release locks only after obtaining all the locks. The most commonly used protocols for locking is Two Phase Locking (2PL) [35]. The algorithms of this class avoid any concurrent execution of transactions when there are potential conflicts. This policy is called pessimistic because the underlying assumption is that any pair of transactions that runs in concurrence is likely to come into conflict. One of the transactions waits therefore until the validation of the other. However, the application of this protocol can generate waiting times for an indefinite period of blockage. Thus, the use of this mechanism may be advantageous to maintain the logical consistency of the database, but against party may compromise the temporal constraints imposed on transactions.

Other techniques to ensure a logical and temporal coherence are proposed. Among them one can be noted the work in [36], which presents an object-oriented semantic concurrence control technique called *semantic locking technique*. With this technique it is possible to provide both logical and temporal consistency of data and transactions by defining criteria for negotiation between them. It is also possible to express the imprecision resulting from the negotiation using the concept of semantic locking to determine which transactions can call the methods of an object. The semantic locking is controlled in each object individually by a compatibility function (*CF*) that implements mechanisms to control concurrent access to object methods.

A technique somewhat similar can be noted in [7] which is based on the epsilon-serializability (*ESR*) criterion [37]. The *ESR* relaxes the severity of the classic serializability (*SR)* in the transaction processing by allowing a limited inconsistency in the database. This limited inconsistency is automatically maintained by the divergence control algorithms (*DC*), in the same way that the *SR* is managed by the concurrency control (*CC*) techniques. In the chapter 4, a model to simulate real-time database techniques on WSNs that uses a *DC* algorithm with two phases [7] named *2PLDC* is presented. *2PLDC* is an extension of the algorithm of *2PL* concurrency control of the classic serializability.

Classification of Real-Time Transactions

Transactions in a real-time DBMS can be classified either according to their real-time constraints, or their type of data access:

1. Classification according to real-time constraints

According to the consequences of deadlines failure of transactions on the application and its environment, transactions in a real-time DBMS are classified into three categories [29, 38, 39]:

- *Hard deadline transactions* (strict and critical deadlines): These are transactions of which the non-respect of their deadlines can lead to serious consequences for the system or controlled environment. For example [40], in a monitoring system of the core of a nuclear central, the transactions have strict and critical deadlines. Indeed, if a sensor for monitoring this nuclear central were to no longer meet the intervals of measures taking, taking one every 5 minutes instead of every second, the consequences could be here devastating.
- *Firm deadline transactions* (strict and non-critical deadlines): if a transaction misses its deadline, it becomes unnecessary for the system. It is thus aborted as soon as it misses its deadline. As an example [41], given a real-time application of the control of an assembly line consisting of conveyors, assembly stations and robots. In this application, robots with cameras are used to acquire the characteristics of objects (transported by conveyors) marching in front their fields of vision. If the acquisition operation is not completed before the disappearance of the object of the field of view of the robot, it is aborted and restarted to take into account the next object.
- *Soft deadline transactions* (non-strict deadlines): if a transaction misses its deadline, the system may not abort it immediately. Indeed, the usefulness of the data only decreases. Multimedia applications are a good example of a system handling transactions of type soft [41]. Thus, in a multimedia system, one sets deadlines for the sound and image reception in order to ensure proper synchronization of these two parameters at the time of their presentation to the end user. If the image and the sound come on two different channels, there may be time-lag between the receipt of the sound data and the image data. When an image arrives a little bit late, it can always be presented to the user if the time-lag from the sound is not too important.

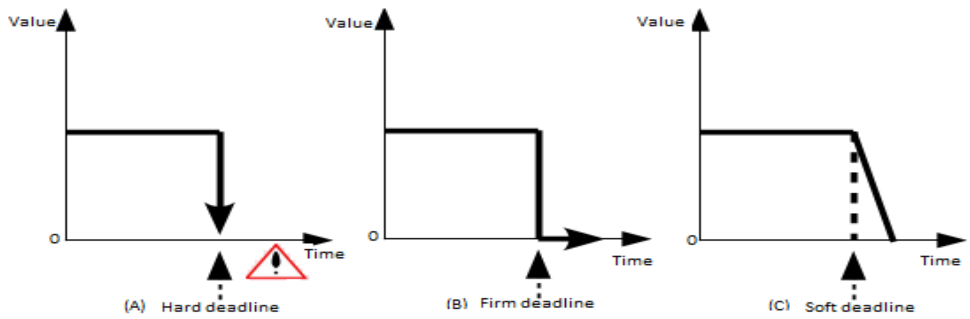


Fig. 3. Illustration of various types of real-time constraints offering different quality of services.

2. Classification according to the type of data access (read/write)

Two types of transaction for the real-time DBMS were proposed according to the type of data access [23, 24]: *update transactions* and *user transactions*. Update transactions, also called sensors transactions, are write transactions that obtain the current state of the targeted environment and periodically update the real-time sensor data in the database. This kind of transactions can be also executed sporadically to update the derived data in order to reflect the state of the real world. Derived data are the data computed using sensor data. For example, the lane of an aircraft is derived from the location and the altitude data items. User transactions, representing user queries, arrive aperiodically. They may read or write non real-time data, but only read real-time data.

Real-Time Database Models

The relational model is the most used to model data for real-time databases. Among these models include the Ramamritham model [29] where the data can have absolute and relative temporal constraints. The definitions given in the sub-section 1.3 for real-time data and transactions comply with this model.

Many researchers argue that the object-oriented model is more suitable than the relational model [42] to model real-time data, because of the nature of several real-time applications, which handle complex real-world objects with time constraints. Therefore, many projects on real-time databases have chosen the object-oriented model for their systems [43, 44]. In this last project [44], the RTSORAC (Real-Time Semantic Objects Relationships And Constraints) is presented. The RTSORAC model includes three components to model the characteristics of an object-oriented real-time database: the *objects*, the *relations* and the *methods*. The objects represent the entities of the system; the relations represent the associations between the objects and define the inter-objects constraints in the database. The methods are executable entities accessing objects and relations in the database. A set of constraints for expressing the logical and temporal constraints is defined to specify correctly an object. In the chapter 4, the proposed model for simulating real-time database techniques on WSNs is based on the object-oriented model.

WSNs and Data Management

Generally a WSN has a large number of nodes distributed on an interest area and communicating between them so as to measure a physical quantity (e.g., pollution level in a given area) or to do an event monitoring (e.g., vehicles tracking). WSNs are used with different applications in many areas and are very important for applications that should be deployed in places hostile to human interventions (e.g., volcano monitoring). Each network node is considered smart and embeds these units: a sensor unit, which provides a measure of environmental data (such as temperature, humidity, pressure, acceleration, sound, etc.), a processing unit, a storage unit, a communication unit, and an energy unit. The communication unit usually performs data transmission by means of radio [45, 46, 47, 48]. However, the resources of a sensor are generally very limited, particularly in terms of storage and energy. The sensor network lifetime depends on the available energy in the nodes composing the network [49, 50]. This available energy is consumed by three activities [46]: sensing activity (data acquisition from the environment), communication (sending and receiving packets), which is essential to form a WSN, and data processing, which consists in some operations applied over data by smart sensors [51, 52]. However, the sensing and processing activities are much less expensive in energy consumption than the wireless communication activities [53]. Thus, energy conservation should be the main point attention of algorithms designed for sensor networks.

Once the sensors perform their measurement, the problem of data storing and querying arises. Indeed, the sensors have restricted storage capacity [54] and the on-going interaction between network devices and environment results in huge amounts of data. There are two main approaches to data storage and querying in WSNs [55, 56]: the distributed approach and the warehousing approach.

1. The *warehousing approach* is a centralized system, the sensors act as simply data collectors [57]. The data gathered by sensors are periodically sent to a central database where user queries are processed. This model is the most used one in data storage and query processing. However, it has some drawbacks, such as eventual wasting resources and bottleneck with an immense amount of transmitted data. Moreover, this approach is unsuitable to real-time processing because it involves time delay for the results.

2. The *distributed approach* is the alternative, where sensors are able to store, locally process and transmit the data they produce [52, 58]. In this approach, the sensed data are not periodically sent to the database server. They remain in the sensor nodes and some queries are distributed and evaluated among the nodes into the network. This reduces the energy consumption and data transfer and, thus increases the network lifetime. This approach that consists to process the data inside the sensor nodes themselves is called *in-network processing* and in addition of energy minimization, it can offer several other advantages such as quasi real-time query processing, long-running queries and instant queries support. For more details about this approach and the current and relevant proposals, readers can see Chapter 3.

There are three sorts of transactions in sensor networks: (i) *historical data queries*, which are run against the server; (ii) *instant queries*, which are run against a device in an instant of time; and (iii) *long-running queries*, which refer to queries run against a device during a time interval [55, 60].

System Architecture

The possible integration of RT-DBMS with sensor networks ensures that the actual data collected by the sensors are used by transactions temporally valid. The figure 4 illustrates the architecture of the system, with respect to the definition of a real-time system mainly consisting of the process and the controller (in the sub-section 1.2).

The controller is composed of a database server computer, called here database warehousing (*DBW*), sensors and a user interface, while the process represents the targeted environment and its different physical entities. The DBW stores data forming the historical data obtained from the targeted environment via sensors and consists of a software with RT-DBMS features and a database represented respectively by RT-DBMS and DBW in the figure. The RT-DBMS [61] consists of four components: the *transactions manager*, the *scheduler*, the *recovery manager* and the *cache manager*. Through the transactions manager, transactions are received and transmitted by the scheduler that performs the concurrent execution control of transactions. The cache manager keeps the cache moves data from volatile storage unit to the stable unit. The recovery manager is responsible for ensuring that the database contains all the effects of a committed transaction and no effects of an aborted transaction [61]. Each smart sensor is able to sense, process and transmit information about the targeted environment. So, these sensors act as local databases, named *sensor database* (*SDB*) composed of a DBMS and a database (DB). The controller Receives information from the environment of the process via sensors and controls the changes of state of the process.

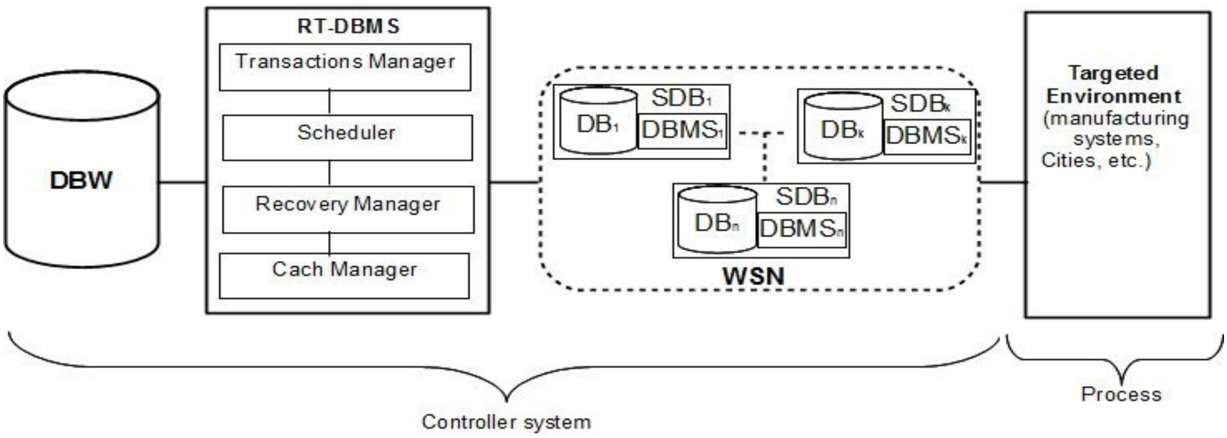


Fig. 4. System architecture.

The scope of this thesis stretches over the real-time databases management for WSNs. The research work presented in this thesis focuses on the study of challenges in handling real-time data storage and query for WSNs, and on the various techniques of real-time databases management for WSNs. The intrinsic energy limitation of WSNs, the temporal constraints of real-time data, and the limitation of warehousing based approaches motivated the design of new real-time data processing techniques for WSNs that minimize the energy depletion in the network and are suitable for real-time applications. The techniques proposed in this research work are based on a distributed approach, statistical modelling techniques, and the analysis of the temporal features of data and transactions, particularly the temporal validity of the data and the query latency.

2. Problem Definition and Research Objectives

The problem addressed in this thesis is the description of the different characteristics and the proposal of novel solutions of real-time database techniques in WSNs. Thus, as already mentioned previously, in real-time applications the data collected by WSNs must closely reflect the current state of the targeted environment. However, the environment changes constantly and the data is collected in discreet moments of time. As such, the collected data has a temporal validity, and as time advances, it becomes less accurate, until it does not reflect the state of the environment any longer [23, 24]. In this context, latency and energy-efficiency become fundamentally important due to the real-time requirements of the tasks and the resource limitations of WSNs.

In most real-time application, the data collected by sensors are periodically sent to a central base station where real-time user transactions are routed and processed, so latency and energy-efficiency can suffer drastically. Another important factor is the obligation to reply to various applications with different needs. In fact, there are several flavours of real-time applications that are generally specific. For instance, for some real-time applications, the accuracy of results may be sacrificed to reduce the query latency. Therefore, a new query processing optimization to improve both the individual query latency with valid data and the network lifetime with respect to various types of applications and specified quality levels is very important in this context. One processing technique could not be efficient for the different applications.

At the beginning of this research work, the no existence of a tool that takes into account the specific characteristics of the real-time databases models for WSNs was noticed. In fact, compared to testbed measurements, the simulation is the least expensive and fastest means to explore many solutions of this kind of complex systems. Therefore in this context, the use of a simulator for a validation phase before implementation and deploying is proved to be very useful. In recent work, a model for a simulation framework of the real-time databases management on WSNs that uses a distributed approach has been proposed, see the Chapter 4 for more details. This model uses the *Earliest Deadline First* (*EDF*) protocol [62] to schedule transactions and the *Epsilon Serialisability techniques* [37] to allow conflicting transactions to execute simultaneously in way that their scheduling does not lead an imprecision that is higher than the one accepted in the data. The model has been implemented by using the object-oriented model.

The main objective of this thesis is the proposal of a solution to optimize the real-time query processing in the context of WSNs. Moreover, it will be question to propose and evaluate the performances of a framework of real-time database techniques that meets the requirements of real-time applications based on WSNs. The new framework should allow offering an effective tool of decision for various types of time-critical applications; while the proposed query processing optimization solution should optimize the real-time user query processing for providing both real-time data processing and energy saving. To achieve the main objectives of this thesis, the following intermediate objectives were defined:

1. One of the objectives of this thesis is to study the state of the art of various techniques of real-time database management in wireless sensor networks as well as in traditional networks. The papers that expose the various techniques will be explored for identifying their advantages, limitations, and challenges in order to choose best solutions suggested for a coupling or provide a new proposal. The specifications of relevant applications as base support of the problematic will be also studied.
2. The second intermediate objective is the analysis of different distributed database management techniques for wireless sensor networks that will help for the proposals. In fact, based on the literature analysis and experiments, in WSNs, the warehousing technique is the most used one for managing sensor data, however, the distributed technique is the most energy-efficient and real-time appropriate technique.
3. Since there is no tool for experimenting and validating real-time database techniques for WSNs, one of the intermediate objectives is to propose and create a new real-time database management model for wireless sensor networks. The performances evaluation and the quality of service (QoS) of the model will be demonstrated.
4. Due to the real-time requirement of the data and tasks in real-time applications and the resource limitations of WSNs, to achieve the main goal it is necessary to propose a new real-time query processing technique that optimizes both the latency and the energy consumption. Its performance should be evaluated in in different scenarios.

3. Thesis Statement

This thesis proposes a new approach based on statistical modelling techniques and a distributed approach to optimize the real-time query processing for both latency and energy minimization. Additionally, it proposes a new model for a simulation framework of the real-time databases management on WSNs that uses a distributed approach. The thesis statement is the following:

The underlying constraints of real-time data management for WSNs impose the need to provide new approaches that save energy with good individual query latency, as well as with valid data. However, the off-line data treatment of the warehousing approach paradigm encourages the energy depletion and can involve time delay for the results that degrades also the data validity. Moreover, the use of the temporal validity of the data to be accessed to abort or make wait the *transactions in first steps of some proposed algorithms can lead to performance slowdown when there are no data variations that justify these operations. The distributed approach combining with statistical modelling techniques may be used to improve the network lifetime, the individual query latency, and the data validity.*

In order to support this thesis statement, the following research approach was adopted.

The problem is studied and the state of the art on various techniques of real-time databases management in wireless sensor networks as well as in traditional networks is explored. The advantages and limitations of each technique proposed in other research works are analysed, as well as the characteristics and design constraints of the two underlying field, the real-time databases and the WSNs.

Traditional real-time database techniques are not suitable for WSNs; they need to be adapted in order to take into account the resource constraints of WSNs. In fact, WSNs are very resources constrained, particularly in terms of storage and energy, and the sensor network lifetime depends on the available energy in the nodes composing the network. Therefore, optimizing the energy consumption should be one of the main points attention in works related to WSNs. Thus, the literature and experiments analysed reveal that one of the best ways to save energy in WSNs data management is to minimize the activities of sensor nodes, mainly the data communication tasks that lead to the most big energy depletion in the network. This can be achieved with the distributed approach, instead of the warehousing approach. The observations made on the distributed approach literature show that it allows performing in-network query processing (which can be combining with data reduction techniques), which reduces the size of transmitted data and avoid to periodically send data from the network nodes to a base station. Moreover, the in-network query processing provides quasi real-time query processing, thus query processing inside the devices themselves means that the most current data will be acquired.

The real-time database techniques literature reveals that the main purpose of such system is to process transactions on time, while maintaining logical and temporal consistency of data. In fact, in real-time applications the data collected by WSNs must closely reflect the current state of the targeted environment and yet this data has a temporal validity, and as time advances, it becomes less accurate, until it does not reflect the state of the environment any longer. In this context, minimizing the query latency with valid data returned is fundamental. Moreover, the analysis of real-time applications reveals that for some applications the accuracy of the results may be sacrificed to reduce the latency [11]. Thus, some applications can tolerate reading stale data under some limits. This observation enables performing query estimation that uses statistical modelling techniques that provide good query latency with valid data under some uncertainty the user/application is willing to tolerate.

The combination of statistical modelling techniques with the distributed approach enables providing a new architecture and a query processing algorithm to optimize the real-time user query processing for both latency and energy minimization with valid data. Hence, a statistical module is used to statistically analyse historical sensor data and define the data error accordingly. If a query is submitted in the system and if the statistical module concludes that the data does not vary a lot along the time and the data error is smaller than the defined threshold (ε), then it can return the result, adding (for example) an estimate of the error given by the 95% confidence interval, otherwise it would query the network without waiting or aborting. This allows real-time data processing in energy-efficient way.

The researches performed on real-time database management techniques show that there is no tool for experimenting and validating real-time database techniques for WSNs. This motivates the research on how to model and easy implement a simulation framework for real-time databases management on WSNs. This reveals that the object-oriented model is more suitable than the relational model-to-model real-time data, because of the nature of several real-time applications, which handle complex real-world objects with time constraints. So, the proposed simulator, based on the distributed approach highlighted in previous researches, uses object-oriented modeling and is implemented in Java. In order to demonstrate the validity of the model, a case study shows the execution of real-time transactions and the energy consumed in the network.

4. Main Contributions

This section summarizes in five points the main scientific contributions of this thesis.

The first contribution of this thesis is a deep review of the state of the art addressing the main characteristics of real-time databases management in WSNs as well as in traditional networks. It highlights the design constraints of real-time databases management on WSNs and provides a comprehensive analysis and review of the current solutions proposed on real-time database techniques on WSNs. This study is described in chapter 2, which consists of an article published in Journal of Network and Computer Applications (JNCA), from Elsevier [63].

The second contribution of this thesis is a detailed literature review on distributed database management techniques for WSNs. This work aimed to show how distributed database techniques are adapted to wireless sensor networks in order to improve the management of the great amount of sensed data in an energy-efficient way by presenting and classifying the most recent and relevant proposals of distributed database management on WSNs. This was very useful for the main contributions of this thesis, which are based on the distributed approach. Moreover, it presents a comprehensive analysis and open issues to facilitate further contributions. This study is described in chapter 3, which consists of an article accepted for publication in IEEE Transactions on Parallel and Distributed Systems (TPDS) [64].

The third contribution of this thesis consists of a proposal of a simulation framework for real-time database on WSNs. The model of the simulator is based on the distributed approach and uses the *Earliest Deadline First* (*EDF*) protocol to schedule transactions and the *Epsilon Serialisability techniques* to allow conflicting transactions to execute simultaneously in way that their scheduling doesn't lead an imprecision that is higher than the one accepted in the data. The model has been implemented by using the object-oriented model. This tool based on object-oriented modelling and implemented in Java was mostly built to experiment and validate the real-time database techniques for WSNs. It can help practitioners and researchers to know the different characteristics and constraints to take into account in real-time databases management on WSN. Second, this simulator allows also to researchers to know what components take part in the design of a model of architecture for real-time database management techniques for WSNs, how to build and easy implement one model of architecture. Third by using this simulator, practitioners and researchers can simulate one real-time database protocol for WSNs and approximately test the temporal and logical validity of data and transactions before deployment with real measures. Fourth, researchers and developers may improve the simulator by adding the simulation of protocols or place part of this simulator on another existing simulator. This work is described in chapter 4, which consists of an article published in Journal of Network and Computer Applications (JNCA), from Elsevier [65].

The fourth contribution of this thesis is the proposal of a new real-time query processing optimization for WSNs. This proposal combines statistical modelling techniques with the distributed approach to provide a new architecture and a query processing algorithm for optimizing the real-time user query processing for both latency and energy minimization with valid data. This valid data is stained of some uncertainty *(ε)* the user/application is willing to tolerate. This proposed architecture and query processing algorithm are described in chapter 5, which consists of an article accepted for publication in Iternational Journal of Sensor Networks (IJSNet) [66].

The fifth and last contribution of this thesis is the extension and the adaption of the previous work to the area of wireless body area Networks (WBANs) in order to propose a new real-time query processing optimization for cloud-based wireless body area Networks (WBANs). This proposal consists of a new architecture of cloud-based WBAN with its underlying query processing algorithm for secure and powerful storage, energy-efficient and real-time data processing. Thus, this proposed approach combines a cloud-based WBAN services with statistical modelling techniques to optimize both the access of storage infrastructure and the real-time query processing to save more energy while meeting time constraints. This proposed cloud-based WBAN architecture and query processing algorithm are described in chapter 6, which consists of an article published in Information Sciences journal (INS), from Elsevier [67].

5. Manuscript Organization

This manuscript is structured in seven main chapters. Each of the chapters represents an article published in or submitted to an international journal, with the exception of the first and the last chapters, which represent the main introduction and conclusions and future work. To keep consistency in the entire document, the Introduction chapter includes its own list of references, as it is done in each article. For the same reason, the lists of tables and figures are presented according to their belonging to a chapter and the list of acronyms is established with respect to the first occurrence of an acronym in a chapter. The structuring of the main chapters of this thesis can be summarized as follows.

Chapter 1 defines the context of this thesis by describing the focus and scope of the thesis in detail and defining the addressed problem and the objectives to be reached, as well as the thesis statement and the adopted approach for solving the problem. Moreover, this chapter includes a summary of the main contributions and the description of the manuscript organization.

Chapter 2 provides an introduction to real-time databases management on WSNs by presenting the motivation and a brief background for real-time databases management in traditional networks as well as in WSNs, and focusing on the current solutions proposed on real-time database techniques on WSNs. Moreover, a summary of the different proposed solutions as well as a detailed discussion are provided.

Chapter 3 provides a detailed study on distributed database management techniques for WSNs. It describes the motivation and a brief background, which highlights the main approaches of data management and the essential methods to design a distributed data management system in WSNs, and presents and classifies the most recent and relevant proposals of distributed databases management for WSNs. Moreover, the summary of the different proposals and their features as well as a detailed discussion are provided.

Chapter 4 follows the work described in the two previous chapters, but focuses on the proposal of a simulation framework for real-time database on WSNs. It presents the first main contribution of this thesis, which is based on the two previous chapters to provide a tool for mainly testing and validating the real-time database techniques for WSNs.

Chapter 5 provides a brief summary of the main points of the two previous chapters, but focuses on the proposal of a new method to optimize the real-time query processing on WSNs for both latency and energy minimization. This proposal includes an architecture and a query processing algorithm, which are mainly based on statistical modelling techniques and the distributed approach.

Chapter 6 provides a new proposal of real-time query processing optimization for cloud-based wireless body area networks (WBANs). The work in this chapter extends and adapts the previous work in the fifth chapter to the area of WBANs in order to propose a new architecture of cloud-based WBAN with its underlying query processing algorithm for secure and powerful storage, energy-efficient and real-time data processing.

Chapter 7 highlights the most important conclusions and contributions of this thesis and pinpoints directions for further research works.

References

[1] P. K. Eswaran et al., “The notions of consistency and predicate locks in a database system,” *Communications of the ACM*, v. 19, n. 11, p. 624–633, November 1976.

[2] K. Ramamritham and P. K. Chrysanthis, “A taxonomy of correctness criteria in database applications,” *The VLDB Journal*, 5(1) :085–097, 1996.

[3] A. Brayner, “Transaction Management in Multidatabase Systems,” *[S.l.]*: *Shaker Verlag*, 1999. ISBN 3-8265-6142-2.

[4] P., Bernstein, V., Hadzilacos, N. Goodman, “Concurrency control and recovery in database systems,” *Addison-Wesley*, 1987.

[5] D. J. Ullman, “Principles of Database Systems,” Computer Science Press, Inc., Maryland, 1980.

[6] S. C. Date, “An Introduction to Database Systems,” *Addison-Wesley*, 1985.

[7] E. Bouazizi, B. Sadez, C. Duvallet, “Applicabilité du critère d'epsilon-sérialisabilité dans les SGBD temps-réel,” *Special issue ISDM MCSEA*. 2004.

[8] J. Gray, “The Transaction Concept: Virtues and Limitations,” (Invited Paper). *VLDB* 1981, 144-154.

[9] R. Elmasri, S. Navathe, “Fundamentals of Database Systems,” *2nd Edition, New York: Addison-Wesley Pub. Co.*, 1994.

[10] Jane W. S. Liu, “Real-Time Systems,” *PrenticeHall*, 2000, ISBN 0130996513.

[11] S. Pailler, “Analyse Hors Ligne d’Ordonnançabilité d’ Applications Temps Réel comportant des Tâches Conditionnelles et Sporadiques,” *PhD Thesis*, 2006, 21-30.

[12] C. Hanen and A. Munier, “Cyclic scheduling on parallel processors : An Overview,” *volume Scheduling theory and its applications, P. Chretienne et al., Chap 9. John Wiley & Sons*, 1995.

[13] C.L. Liu and J.W. Layland, “Scheduling algorithms for mutltiprogramming in real-time environnement,” *Journal of the ACM*, 20(1) :46–61, 1973.

[14] P. Puschner and R. Nossal, “Testing the results of static worst case execution time analysis,” *In IEEE Real Time Systems Symposium*, dec 1998.

[15] E. Klingerman and A.D. Stoyenko, “Real time euclid : a language for reliable real time systems,” *IEEE Trans. softw. Eng.*, 12(9) :941–949, 1986.

[16] P. Puschner and C. Koza, “Calculating the maximum execution time of real-time programs,” *Journal of Real-Time Systems*, 1(2) :159–176, 1989.

[17] J. Gustafsson, B. Lisper, N. Bermudo, C. Sandberg, and L. Sjöberg, “A prototype tool for flow analysis of c program,” *In 2nd Intl. Workshop on WCET Analysis 14th Euromicro Conference on Real-Time Systems*, June 2002.

[18] C. Healy, M. Sjödin, V. Rustagi, and D. Whalley, “Bounding loop iterations for timing analysis,” *In 4th IEEE Real-Time Technology and Applications Symposium (RTAS’ 98)*,Discrete Event Systems: Analysis and Control,pages 12–21. IEEE, 1998.

[19] T. Mitra and A. Roychoudhury, “A framework to model branch prediction for wcet analysis,” *In 2nd Intl. Workshop on WCET Analysis 14th Euromicro Conference on Real-Time Systems*, June 2002.

[20] C. Rochange and P. Sainrat, “Difficulties in computing the wcet for processors with speculative execution,” *In 2nd Intl. Workshop on WCET Analysis 14th Euromicro Conference on Real-Time Systems*, June 2002.

[21] A. Colin and I. Puaut, “Worst case execution time analysis for a processor with branch prediction,” *Real-Time Systems*, 18(2): 249–274, May 2000.

[22] M. Hamdaoui and P. Ramanathan. “*A dynamic priority assignment technique for streams with*, 1995.

[23] N. Idoudi, C. Duvallet, B. Sadeg, N. Louati, R. Bouaziz, F. Gargouri, “How to model a real-time database?,” *Proceedings of 12th IEEE International Symposium on Object-oriented Real-time distributed Computing (IEEE ISORC'2009)*. Tokyo, Japan: IEEE, 2009a, pp. 321-328.

[24] N. Idoudi, N. Louati, C. Duvallet, R. Bouaziz, B. Sadeg, F. Gargouri, “A Framework to Model Real-Time Databases,” *International Journal of Computing & Information Sciences* Vol. 7; January 2009b, pp.1 – 11.

[25] L. C. DiPippo, V.F. Wolfe, “Real-time databases,” *Journal of Database Systems Hand-book, Multiscience Press (Citeseer)*, 1997.

[26] S.H. Son and J.W.S. Liu, “Maintaining temporal consistency: pessimistic vs. optimistic concurrency control,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 7, n° 5, 1995.

[27] M. Xiong, J.A. Stankovic, K. Ramamritham, D. Towsley and R.M. Sivasankaran, “Maintaining temporal consistency: issues and algorithms,” *1st Int. Workshop on RTDBS*. Issues and Applications, p. 1-6, California 1996.

[28] S.H. Son and J.W.S.Liu, “How well can data temporal consistency be maintained ?,” *IEEE Symp. on Computer-Aided Control System Design*, 1992.

[29] K. Ramamritham, “Real-time databases,” *Journal of Distributed and Parallel Databases (Springer)* 1993; 1(2): 199-226.

[30] Chagas Lunas Dantas, Lima Eduardo Pereira, Ribeiro Neto Pedro Fernandes, “Real-Time Databases Techniques in Wireless Sensor Networks,” *Sixth International Conference on Networking and Services, IEEE*, 2010.p. 182–188.

[31] J.A. Stankovic, M. Spuri, M. Di Natale and G. C. Buttazzo, “Implications of classical scheduling results for real-time systems,” *IEEE Computer*, vol. 28, n° 8, p. 16-25, 1995.

[32] J. Lindström, “Integrated and adaptive optimistic concurrency control method for real-time databases,” *8th International Conference on Real-Time Computing Systems and Applications (RTCSA’2002)*, University of Helsinki Finland, p. 143–151, 2002.

[33] J. Lindström, “Optimistic Concurrency Control Methods for Real-Time Database,” *PhD Thesis*, Department of Computer Science, University of Helsinki Finland, November 2002.

[34] K. Raatikainen, J. Lindström, “Using Real-Time Serializability and Optimistic Concurrency Control in Firm Real-Time Databases,” University of Helsinki Finland, March 2002.

[35] D. Nyström et al., “Pessimistic concurrency-control and versioning to support database pointers in real-time databases,” *In: Proceedings of the 16th Euromicro Conference on Real-Time Systems (ECRTS04)*. Sicily, Italy: IEEE Computer Society Press, 2004.

[36] L. C. DiPippo, “Semantic real-time object-based concurrency control,” *PhD Thesis*, Department of Computer Science and Statistics, University of Rhode Island, 1995.

[37] C. Pu, “Generalized Transactions Processing with Epsilon Serializability,” *In Proceedings of 4th International Workshop on High Performance Transactions Systems*, Asimolar, California, 1991.

[38] J.R. Haritsa, M.J. Carey and M. Livny, “Data access scheduling in firm real-time database systems,” *J. of RTS*, vol.4, n° 3, p. 203-241, 1992.

[39] B. Purimetla, R. M. Sivasankaran, K. Ramamritham and J. A. Stankovic, “Real- time databases: issues and applications,” *Principles of RTS*, ed. Printice Hill, 1994.

[40] G. Noël, “Indexation dans les bases de données capteurs temps réel,” *PhD Thesis*, National Institute of Applied Sciences Of Lyon, 2006.

[41] C. Duvallet, Z. Mammeri, B. Sadeg, “Les SGBD temps réel,” *In Technique et Science Informatiques, Hermes ed.*, 479-

516, June 1999.

[42] W. Kim, “Object-Oriented Database Systems: Promises, Reality, and Future,” *Modern Database Systems*, pp 255-280. Addison-Wesly, 1995.

[43] F. Terrier, A. Lanusse, D. Bras, P. Roux, P. Vanuxeem, “Concurrent object for multitasking,” *In L'objet*, v3. pp 179-196, Paris, 1997.

[44] V. F. Wolfe, J. J. Prichard, L. C. Dipippo, J. Black, “Real-Time Database System: Issues and Applications,” *chapter The RTSORAC Real-Time-Object-Oriented DataBase Prototype*, pp 279-301. Kluwer Academic Publlishers, 1997.

[45] L. M. L. Oliveira and J. J. P. C. Rodrigues, “Wireless Sensor Networks: a Survey on Environmental Monitoring,” *Journal of Communications (JCM)*, Vol 6, No 2, pp. 143-151, April 2011, DOI: 10.4304/jcm.6.2.143-151

[46] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer networks*, 38, 393-422, 2002.

[47] F. I. Akyildiz, T. Melodia, and R. K. Chowdhury. “A survey on wireless multimedia sensor networks,” *Journal of Computer networks (Elsevier)* 51, no. 4 (March 2007): 921--960.

[48] L.M. Luís Oliveira, F. de Sousa Amaro, and Joel J. P. C. Rodrigues. “Routing and Mobility Approaches in IPv6 over LoWPAN Mesh Networks,” *in International Journal of Communication Systems, Wiley*, ISSN: 1074-5351, Vol. 24, Issue 11, November 2011a, pp. 1445-1466, DOI: 10.1002/dac.1228.

[49] K. Lin, M. Chen, S. Zeadally, and J. J. P. C. Rodrigues, “Balancing Energy Consumption with Mobile Agents in Wireless Sensor Networks,” *Future Generation Computer Systems*, Vol. 28, Issue 2, pp. 446-456, Feb. 2012, DOI: 10.1016/j.future.2011.03.001,

[50] K. Lin, J. J. P. C. Rodrigues, H. Ge, N. Xiong, and X. Lian “Energy Efficiency QoS Assurance Routing in Wireless Multimedia Sensor Networks,” *IEEE Systems Journal*, Vol. 5, Issue 4, pp. 495-505, December 2011, DOI: 10.1109/JSYST.2011.2165599

[51] S. Cho, E. Shih, N. Ickes, and R. Min. “Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks,” *ACM/IEEE MOBICOM. ACM*, 2001. 272– 287.

[52] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “TinyDB: An Aquisitional Query Processing System for Sensor Networks,” *ACM Transactions on Database Systems (TODS) (ACM)* 30, no. 1 (2005): 122-173.

[53] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “TAG:Atiny aggregation service for ad-hoc sensor networks,” *In OSDI.Citeseer*, 2002.

[54] F.A. Silva, T.R.M. Braga, L.B. Ruiz, J.M.S. Nogueira, “Tecnologia de Nós Sensores Sem Fio,” *Contrôle & Instrumentação*; agosto 2004;92:76–87.

[55] P. Bonnet, P. Seshadri, “Device Database Systems,” *In: Proceedings of the International Conference on Data Engineering ICDE’99*. San Diego, CA: *IEEE Computer Society*; 2000.

[56] P. Bonnet, J. Gehrke, P. Seshadri, "Querying the Physical World," *Journal of Personal Communications, IEEE* 2000;7(5):10–5.

[57] S. Servigne, G. Noel, "Real time and spatiotemporal data indexing for sensor based databases," *Journal of Geospatial Information Technology for Emergency Response (Taylor & Francis Group)* 2008 ; 6: 123-142.

[58] Y. Yao and J. Gehrke, "Query Processing in Sensor Networks," *In First Biennial Conference on Innovative Data Systems Research (CIDR 2003)*, Asilomar, California, January 5-8, 2003.

[60] Neto Pedro Fernandes R, Perkusich Maria LB, Perkusich Angelo, "Real-Time Database for Sensor Networks," *In: Proceedings of the 6th International Conference on Entreprise Information Systems*, 2004. p. 599–603.

[61] M. Holanda, A. Brayner, S.V. Fialho, "Controle de concorrência em bancos de dados para ambientes de computação móvel," *VI Workshop de Comunicação sem Fio e Computação Móvel,* p. 161–170, 2004.

[62] C. Liu, J. Leyland, "Scheduling algorithms for multiprogramming in hard real-time environment," *Journal of the ACM,* 20 (1), 1973, pp 46-61.

[63] O. Diallo, J. J.P.C. Rodrigues, and M. Sene, "Real-time Data Management on Wireless Sensor Network: a Survey," *in Journal of Network and Computer Applications, Elsevier*, ISSN: 1084-8045, Vol. 35, Issue 3, May 2012, pp. 1013-1021, DOI: 10.1016/j.jnca.2011.12.006.

[64] O. Diallo, J. J.P.C. Rodrigues, M. Sene, and J. Lloret, "Distributed Database Management Techniques for Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, ISSN: 1045-9219, 19 Aug. 2013, IEEE Computer Society, DOI: 10.1109/TPDS.2013.207 (in press).

[65] O. Diallo, J. J.P.C. Rodrigues, M. Sene, and J. Lloret, "Simulation Framework for Real-Time Database on WSNs," *in Journal of Network and Computer Applications, Elsevier*, ISSN: 1084-8045, Vol. 39, March 2014, pp. 191-201, DOI: 10.1016/j.jnca.2013.07.001.

[66] O. Diallo, J. J.P.C. Rodrigues, M. Sene, and F. Xia, "Real-Time Query Processing Optimization for Wireless Sensor Networks," *International Journal of Sensor Networks (IJSNet), InderScience Publishers*, ISSN (Online): 1748-1287 - ISSN (Print): 1748-1279 (in press).

[67] O. Diallo, J. J.P.C. Rodrigues, M. Sene, J. Niu, "Real-Time Query Processing Optimization for Cloud-based Wireless Body Area Networks," in Information Sciences (2014), Elsevier, ISSN: 0020-0255, doi: http://dx.doi.org/10.1016/j.ins. 2014.03.081 (in press).

Chapter 2

Real-Time Data Management on Wireless Sensor Networks: A Survey

This chapter consists of the following article:

Real-Time Data Management on Wireless Sensor Networks: A Survey

O. Diallo, J. J. Rodrigues, and M. Sene

Journal of Network and Computer Applications, Elsevier, ISSN: 1084-8045, 35 (3), pp. 1013-1021, May 2012, http://dx.doi.org/10.1016/j.jnca.2011.12.006

According to 2012 Journal Citation Reports published by Thomson Reuters in 2013, this journal scored ISI journal performance metrics as follows:

ISI Impact Factor (2012): 1.467

ISI 5-Years Impact Factor (2012): 1.251

Journal of Network and Computer Applications 35 (2012) 1013–1021

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Real-Time Data Management on Wireless Sensor Networks: A Survey

Ousmane Diallo[a,b], Joel J.P.C. Rodrigues[b,*], Mbaye Sene[a]

[a]Department of Mathematics and Informatics, UCAD, Senegal

[b]Instituto de Telecomunicações, University of Beira Interior, Portugal

ARTICLE INFO

Article history :
Received 22 July 2011
Accepted 5 December 2011
Available online 14 December 2011

Keywords :
Wireless sensor networks
Real-time database management
Data collection
Survey

ABSTRACT

In the recent past, search in sensor systems focused on node hardware constraints and very limited energy resources. But nowadays, that new applications need data processing with temporal constraints in their tasks; then one of the new challenges faced by wireless sensor networks (WSNs) is handling real-time storage and querying the data they process. Two main approaches for storage and querying data are generally considered warehousing and distributed. The warehousing approach stores data in a central database and then queries may be performed to it. In a distributed approach, sensor devices are considered as local databases and data are managed locally. The data collected by sensors must represent the current state of the environment; for this reason they are subject to logic and time constraints. Then, this paper identifies the main specifications of real-time data management and presents the available real-time data management solutions for WSNs, in order to discuss them and identify some open issues and provide guidelines for further contributions.

1. Introduction

Wireless Sensor Networks (WSNs) may be defined as a set of smart devices, called sensors, which are able to sense and transmit information about the environment on which they are deployed. These devices collect information for users interested in monitoring and controlling a given phenomenon and transfer them to a collected point called sink node. The latter makes the information available to a gateway where the users can access via Internet. So as to obtain information, users use applications that communicate with the network through queries (Callaway, 2004; Sacks et al., 2003; Baronti et al., 2007). An illustration of a WSN may be found in Fig. 1.

This sort of network generally has a large number of nodes communicating and distributed on a given area to measure a physical quantity or an event monitoring. Each network node is considered intelligent and is equipped with an acquisition module, which provides a measure of environmental data (such as temperature, humidity, pressure, acceleration, sound, etc.), processing capacity, storage, communication, and energy. However, these resources are generally very limited, especially those of storage and energy, and the sensor nodes energy consumption is sometimes not negligible (Akyildiz et al., 2002, 2007)

Sensors can be used and placed everywhere they make information omnipresent. Consequently, systems based on sensor networks are more and more used in many areas providing then various types of WSNs (Mendes and Rodrigues, 2010). These numerous WSNs have allowed the development of many applications, which are generally connected to databases treating the amount of data collected from sensors. However, the processing time becomes increasingly critical for certain applications. These applications must query and analyze the data more quickly in order to make decisions and to react as soon as possible.

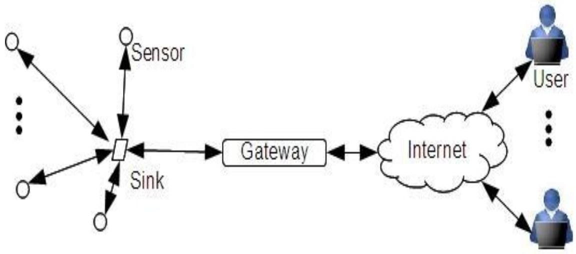


Fig. 1. Illustration of a wireless sensor network architecture.

*Corresponding author. Tel.: +351275242081; fax: +351275319899.
E-mail addresses: ousmane.diallo@it.ubi.pt (O. Diallo),
joeljr@ieee.org (J.J.P.C. Rodrigues), mbaye.sene@ucad.sn (M. Sene).
1084-8045/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2011.12.006

O. Diallo et al. / Journal of Network and Computer Applications 35 (2012) 1013–1021

Some examples of the most popular applications are the following: the control of network traffic (Cranor et al., 2002), transactional analysis (web, banking or telecommunication transactions) (Cortes et al., 2000), human motion tracking application (Chen and Ferreira, 2009), the tracking of actions on dynamic Web pages (Zhu and Shasha, 2002; Chen et al., 2000), monitoring of urban or environmental phenomena (Mainwaring et al., 2002; Ulmer et al., 2003), and the sensors data management (Arasu et al., 2003).

Once the sensors perform their measurement, the problem of data storing and querying arises. Indeed, the sensors have restricted storage capacity (Silva et al., 2004) and the ongoing interaction between network devices and environment results in huge amounts of data.

There are two main approaches for data storage and querying in WSN: distributed and warehousing (Elnahrawy and Nath, 2004). In the first approach, researches recommend a distributed evaluation of requests on sensor networks, and aim at exploiting the capacities of calculation of sensors (Neto et al., 2008). The objective is to locally calculate in order to limit sending of messages, reducing thus the energy consumption. In the second approach, warehousing, one has a centralized system. Collected data from sensors (in stream) are sent to a central database server, in which user requests are processed. Note that this technique generates large data flows.

The data collected by the WSN must closely reflect the current state of the targeted environment. However, the environment changes constantly and the data are collected in discreet times. So, the collected data have temporal validity, as time advances they become less and less accurate, until the time where they do not reflect the state of the environment (Idoudi, 2009a, 2009b). It is fundamental that responses to application queries ensure that returned data comply with logic and temporal constraints. In this context, real-time data management on WSNs is necessary to deal with those constraints. The main goal of real-time data management is to ensure temporal consistence data and process transactions within real-time constraints. The most relevant proposals of real-time data management on WSNs are going to be exposed along this work. This survey aims to study how real-time databases have been integrated in wireless sensor networks in order to satisfy real-time constraints of specifics critics applications connected to those sorts of networks. Moreover, a discussion and open issues on real-time data management on wireless sensor networks will be identified in order to make further contributions.

The remainder of this paper is organized as follows. Section 2 presents the various basic concepts and architectures used on real-time data storage and querying in WSNs while Section 3 exposes research contributions on real-time data management in WSNs. Section 4 discusses the techniques used on the studied approaches and proposes some research issues. Finally, Section 5 concludes the paper.

2. Background

The first purpose of a real-time system is the respect of the temporal constraints. For a database, this does not mean that the transactions execution must be fast, but that transactions must run in well-defined time intervals (fixed time) (Buffenoir, 2006; Lam and Kuo, 2001).

Like a traditional database management system (DBMS), a real-time DBMS (RT-DBMS) must process transactions and ensure that the logical consistency of the data is not violated. However, unlike a traditional DBMS, a RT-DBMS emphasizes on the temporal validity of the data and the time constraints or deadlines for transactions (Idoudi, 2009a, 2009b; DiPippo and Wolfe, 1997).

The main purpose of a RT-DBMS is to process transactions on time, while maintaining logical and temporal consistency of data. The temporal consistency expresses the need to maintain consistency between the current state of the targeted environment and the state as reflect by the database contents. The temporal consistency can be measured in two ways (Ramamritham, 1993):

- *Absolute consistency*, which deals with the need to maintain the view representing the state of the targeted environment consistent with the real state of the environment
- *Relative consistency*, which concerns data derived from other ones.

To satisfy these temporal constraints, the structure of the data must include these attributes: *(i)* timestamp, which indicates the instant when the observation relating the data was made; and *(ii)* absolute validity interval *(avi)* that denotes the time interval following the timestamp during which the data are considered valid. Another attribute can be considered; the imprecision, which refers to how the current state of the targeted environment may differ from the measured data (Ramamritham, 1993). The transactions also must have these attributes: *(i)* liberation time that represent the moment on which all the resources for the transaction processing is available; *(ii)* computing time that indicates the execution time needed for the transaction; and *(iii)* maximum time, which indicates the maximum time limit for the transaction execution and the periodicity that refers to the frequency with which the transaction happens (Chagas et al., 2010).

To satisfy the logical consistency of the data, transactions must be processed with ACID (Atomicity, Consistency, Isolation and Durability) properties. But unlike the conventional databases, in real-time databases these properties are relaxed. Firstly, the atomicity may be relaxed. It is only applied to the sub-transaction that wants to deal with completely data consistency. Secondly, since timeliness is more important than correctness, in many situations, correctness can be traded for timeliness. Thirdly, the isolation allows transactions to communicate with others to better perform control functions. Finally, in real-time databases, not all data must be permanent and some of them are temporal (DiPippo and Wolfe, 1997; Chagas et al., 2010; Ramamritham, 1993).

According to Ramamritham (1993), a set of data used to derive a new data item constitutes a relative consistency set. Each set R is associated with a relative validity interval denoted by R_{rvi}. Assume that, given a data item $d \in R$; d has a correct state if:

1. d_{value} is logical consistency, i.e., satisfies all integrity constraint.
2. d is temporally consistent:

- Absolute consistency: $(\text{current\_time}-d_{timestamp}) \leq d_{avi}$
- Relative consistency: $\forall\, d' \in R,\ |d_{timestamp} - d'_{timestamp}| \leq R_{rvi}$

According to Stankovic et al. (1999), a real-time system can be seen as a *controlling system* and a *controlled system*. In an automated manufacturing system, for example, the controlling system is composed by a computer and human interfaces that manage and coordinate the activities, while the controlled system is the manufacturing system with its robots, assembling stations, parts, and conveyers. Then, the controlled system represents the targeted environment. The controlling system interacts with its environment according to the data available, for example, provided by various sensors about the environment. The data collected by the WSN and perceived by the controlling system must closely reflect the current state of the targeted environment (Idoudi, 2009a, 2009b). However, the environment changes constantly and there are usually delay throughout the process of collecting, storing, and exploiting of the information characterizing the environment. This delay can generate huge inconsistent values that occurs bad prevision. This way, the real-time data management for WNSs can be a solution, since it provides several tools to deal with the temporal restrictions of data and transactions.

As the evolution of the traditional databases management systems, the sensor databases try to create an abstraction between the end-users and the sensor nodes thus allowing the users to only concentrate on the data that they need to be collected rather than bothering with the complexities of mechanisms deciding how to extract data from a network (Bonnet et al., 2001; Madden et al., 2002). This evolution of sensor databases has seen born various data storage and query management techniques designed specifically for WSNs. There are two main approaches to data storage and query in WSNs (Bonnet and Sheshadri, 2000; Bonnet et al., 2000): warehousing and distributed.

The warehousing approach (as illustrated in Fig. 2) is a centralized system (Bonnet et al., 2001). The data gathered by the sensors are sent to a central database where user queries are processed. In this case, the sensors act as simply collectors. The warehousing approach is the most used one in query processing. However, it has some drawbacks, such as, the huge number of generated data can easily create a bottleneck on the central server; the huge amount of information transferring waste the resources. The instructions processing is much less expensive than the wireless data transmission. Indeed, it is shown that power consumption to transmit a bit is equivalent to that consumed to carry out approximately 800 instructions (Madden et al., 2002). Moreover, it is clear that this approach is not adequate for no-historical queries because it involves time delay for the results.

In the other hand, the distributed approach (as illustrated in Fig. 3) exploits the capabilities of sensors calculation and the sensors act as local databases (Bonnet and Sheshadri, 2000). It aims to locally calculate and limit of sending messages in order to save the energy consumption.

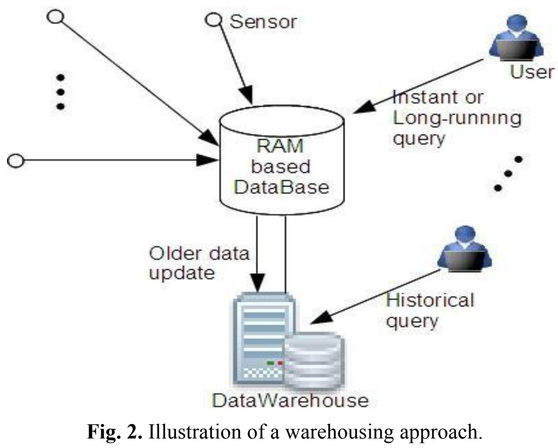


Fig. 2. Illustration of a warehousing approach.

O. Diallo et al. / Journal of Network and Computer Applications 35 (2012) 1013–1021

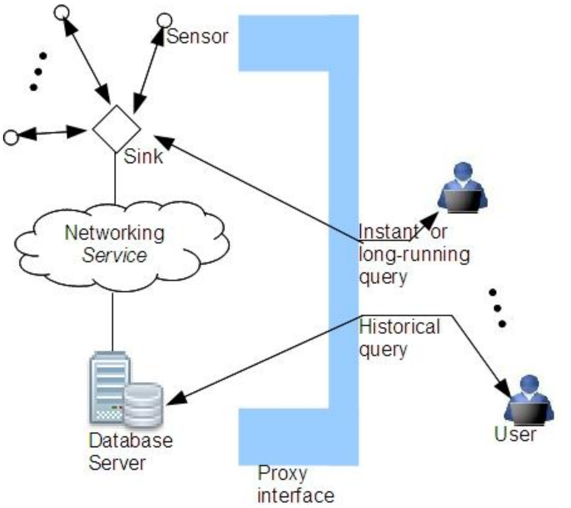


Fig. 3. Illustration of a distributed approach where sensor devices are part of the database.

In this approach, the data are stored in a central database server and in the devices themselves, enabling one to query both. Here, the devices act as part of the database. This approach can offer several advantages; on sensors, the processing of queries is done on quasi real-time; it supports long-running queries and instant queries; and the distributed approach increase the lifetime of the network. Although there is a significant advance in ubiquitous computing, the conditions for using sensor networks still restrict the use of data storage systems located within sensors. Indeed, the improvement of the sensors (operating time, improvement of calculation, and memory capabilities) does not cancel the risks of failure related to natural conditions.

There are three sorts of queries in sensor networks: *(i)* historical data queries, which are run against the server; *(ii)* instant queries, which are run against a device in an instant of time; and *(iii)* long queries, which refer to queries run against a device during a time interval (Bonnet and Sheshadri, 2000; Neto et al. (2004).

3. Current solutions of real-time database techniques for WSNs

The real-time database data must represent the current state of the environment on which they were captured. This is particularly important especially for applications that monitor areas of risk. In this context, a good data structure for real-time access is very important. Thus, under volcanic monitoring, Nöel et al. (2004a, 2004b), Noel et al. (2005a) have proposed a new sensor database spatiotemporal access method, named the Po-tree. Indeed, Po-Tree is an indexing system for centralized spatiotemporal databases, working on data from a fixed sensor network with real-time constraints. It distinguishes two characteristics of the data: spatial and temporal. The Po-tree structure is designed as suit: the spatial characteristic indexed by a Kd-tree (Bentley, 1975) represents the first sub-tree and the temporal characteristic indexed through modified B+-tree represents the second sub-tree. Each sensor is linked through this two linked sub-trees. In this tree, entries take the form *<left-pointer; point-position; link-to-temporal-tree; right-pointer>*, representing thus

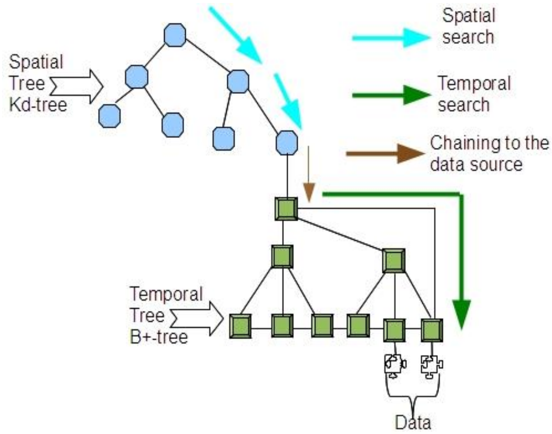


Fig. 4. Illustration of the spatiotemporal query with Po-Tree.

the spatial location, the position in the tree and a link to a temporal sub-tree. After the record of the measurement times a link is created to the recent measured values. In order to accelerate query processing of the most recent data; a direct link between the root and the last node, where the most recent data are usually recorded, is created. Queries are sent to the root of the spatial sub-tree that supports querying of different temporal sub-trees and provides the results. Indeed, a query is divided into two phases: if a query arrives, a first search is performed in the spatial sub-tree to determine the sub-trees of sensors concerned by the query. Subsequently, the spatial sub-tree forwards the query, having only temporal dimensions, to the sub-structures of the sensors involved and collects the results (ID sensor, measurements and time markers) to present to the user (see Fig. 4).

In real-time databases, in addition to logic constraints, data must also be consistent with the time constraints. So, it is necessary to efficiently store new arriving information into databases, to fast search of data and to manage the memory to avoid overload that can lead several miss deadline of transactions (Kang et al., 2004). This is particularly important for applications that need to query in real-time and access to the most recent data. To deal with these challenges, Firstly, Noel et al. (2005b), Noel and Servigne (2005) have proposed a real-time spatiotemporal indexing scheme for agile sensors, named the PasTree. The PasTree is an improved version of the PoTree. The structure is based to a multiversion spatial sub-tree; which keep tracks of the sensor location changes. Thus, each node keeps information about the past positions sensors and the actual ones. Moreover, PasTree includes two other access structures; the sensor tree for referencing sensor ID and the temporal object tree for each recorded sensor, allowing thus queries based on sensor identifiers or spatiotemporal characteristics. This proposal includes an interface located above the sub-trees of access and enables centralized management of queries. This interface allows hiding the user the multi-criterion access. If a query arrives, it redirects it depending on the type of access (spatial or identifiers criteria).

Secondly, Nöel and Servigne (2006) have also proposed a third indexing method, named StH, for real-time management of main memory. The StH, spatiotemporal/Heat, is an indexing system for centralized spatiotemporal databases in RAM memory, working on data from a agile sensor network and taking into account the memory saturation of the database. The StH further allows multi-criteria access to the data. Indeed, it is based, in party, in the construction of the PasTree: spatial sub-tree and identifiers sub-tree. In addition, it incorporates new sub-structures in order to prevent from the memory saturation of the database. Thus, in addition to spatial and identifiers sub-trees, each sensor is associated with a dedicated sensor sub-structure called staircase because of its shape. This dedicated sub-structure contains all information relative to a sensor (id, history of positions and links to the measures). This index allows the resolution of queries based on spatial criteria or sensor identifiers. The management of the memory saturation is done by using a heat-function associated with each staircase, i.e., each sensor. Thus, the coldest data are regularly transferred from the sensor stairs to the data warehouse. This transfer is triggered after the account of a maximum number of data input by the sensor. Less hot data correspond to the less important data the selection of these less important data is made by some criteria, for example by priority.

Servigne and Noel (2008) have shown the complete structure PoTree, PasTree and StH but this time it is increasing the StH linking the database (into memory hand) to a data warehouse. They emphase the aspects of transfer of less important data toward a data warehouse, to release the database. Thus, they based on a warehousing approach where after collecting, data are sent toward a central point (in main-memory databases) for real-time querying and analysis. For long-running querying and free database memory, data are sent toward a disk or secondary-memory (warehousing).

The data acquired by the sensors must represent the current status of the targeted phenomenon. However, this status is subject to constant changes. Thus, in (Bouju et al., 2009). Noel et al. (2005b), the authors attempt to take into consideration the location of the sensors; which can be fixed, agile and mobile. Based on the UML formalism, they define a spatiotemporal sensor data model for real-time storing according to the position (fixed, agile, mobile) of the sensors. Moreover, they emphasis of the importance of metadata modeling and present a formalism of object issued by the sensors.

The data stream management systems meet the operating requirements of applications to query data stream generated continuously flows (Golab and Özsu, 2003). Generally, these types of applications are subject to a huge amount of stream data, triggers, imprecise data, and real-time requirements. To deal with this challenge, in (Abadi et al., 2003; Carney et al., 2002), the authors have proposed a system that provides real-time data stream processing capabilities with a real-time metric the average latency of data tuples, named Aurora. Aurora is a data-flow system that processes incoming stream according to the requirements of the applications (see the aurora architecture in Fig. 5). It includes a set of operators for satisfying the stream processing requirements. Each operator consumes data input, performs operations, and produces results in a continuous manner. Among these operators we can note: windowed operators, filter operator, etc. Aurora can process continual queries in real-time processing according to QoS specifications, but unlike the relational approach, in Aurora query building is procedural, i.e., a query is created using a graphical interface where one place boxes that represent transactions and arrows that represent the data stream.

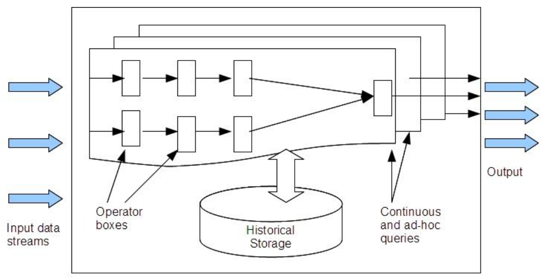


Fig. 5. Architecture of Aurora.

Moreover, the evaluation is made tuple by tuple, unlike the approach of traditional databases where queries are evaluated on relations. To deal with the QoS, Aurora uses a load shedder mechanism that, if it detects an overload situation, flash load till the system find a good performance. Moreover, it uses several utility functions in QoS and associates QoS curves with outputs from stream processing to support continuous timing requirements.

Considering the real-time approach, the main idea is to answer queries in their time constraints; real-time applications need predictable responses. Queries should be process in accord with their deadline. So, RTSTREAM (Wei et al., 2006) try to comply with this constraint in huge amount of sensor data stream. RTSTREAM is a real-time data stream management system (DSMS) that supports a real-time data stream query model, named PQuery. The PQuery model improves upon the Continuous Query Language (CQL) (Arasu et al., 2006) as it allows applications to specify the query frequencies and deadlines; which in turn is used to deal with the queries time constraint. Instead of always-reinitiated query instance by the incoming data stream (like in others continuous query model), PQuery model does not allow the change of the input during the course of the instance execution, i.e. if a query is registered, its instances are periodically initiated by the DSMS system and a newly arrived data stream are processed by the next query instance. Moreover, RTSTREAM includes an overload protection mechanism named data admission for better deal with query deadline miss ratio. The data admission uses sampling and shedding strategies to reduce the incoming data volume during periods of high load. This system produces periodic responses in real time without interrupting the query instance execution, but it does not provide the latest result of the incoming data stream because of the incoming data stream reported to next query instance.

The real time context involves both taking into account the time constraints that a good quality of information provided (Ramamritham, 1993). Thus, the definition of metadata in real time context for spatiotemporal sensor data can be a good support, particularly in decision making-process. There is a standard that represents the structure of the spatial metadata: The ISO 19115 standard (ISO19115, 2006; Servigne et al., 2006) but this standard does not include the dynamic objects and real-time aspect. To meet this need, researchers in (Gutierrez and Servigne, 2007; Gutierrez et al., 2007) attempt to gather the basic features of spatiotemporal data and metadata to better assist in the management quality of real-time spatiotemporal data. Thus, referred to the nature of the sensors; their characteristics and their positions, they define a sensor (fixed, mobile or agile) spatiotemporal data model in real-time context. According to the sensor data model and the future function of the metadata, they expose a generic metadata that can be use by any real-time spatiotemporal application and in the other hand a specific metadata for specific application. They also define two metrics that can help to differentiate data to metadata: the data semantic; characterized by their final use and the data dynami-city characterize by the variation of values on time.

The definition of spatiotemporal metadata in real-time is well appreciated for good quality of service in real-time context, but must be accompanied by good update management, extraction and exploitation policy, especially in dynamic and real-time context. Sensor networks generate large amounts of data very variable and the access time is critical, especially in the context of real time.

Unlike the researches that focus in warehousing approach, in real-time databases for sensor networks (Neto et al., 2004), the authors use a distributed approach to propose an integration between real-time database technology and sensor network systems. Thus, they try to perform both logical and timing verification for the data and transactions. In their model, sensors acquire data; store and periodically they performe updates to the server. Thus, the long-run queries and instant queries can now run against the sensors that are programmed to perform the temporal constraints of data and transactions while the historical queries will be run against the server. For handling the execution concurrence of the transactions in conflict, they propose an in-network layer that deals with the serializability property. Moreover, unlike the other proposals that use programming models to assess their performance, their proposed architecture (sensor network, database server and their components) is modeled by a Colored Petri nets formalism in order to analyze the system performances. They try to demonstrate that their model is consistent with the logical and temporal constraints of data and transactions in the context of real-time data management.

Like a conventional database management systems, a real-time database management system must process transactions while ensuring that the database consistency is not violated. Thus, Chagas et al. (2010) present some real-time database management techniques to deal with this challenge. It uses a distributed approach where the network devices act as database and periodically send the acquired data to the database server. To deal with time constraint, the devices and the server include a program that process transaction while handling the time constraints of the data and transactions. The

PostgreSQL DBMS is used to store data provided by the sensors and the Query Language for Real-Time Databases (QL-RTDB) (Leite, 2005) is used to access to both the server and the devices via an application interface. To deal with the eventual concurrent transactions an algorithm that takes into in consideration the time constraints.

Due to the fact that the real-time applications are subject to a very huge amount of data to process and are temporal constraints many transactions may miss their deadline, degrading then the performances. In these applications, it is also important to access fresh data that effectively reflect the current status of the targeted environment. Considering these problems, Kang et al. (2004) have proposed a real-time main memory database architecture, named QMF (QoS management architecture for deadline Miss ratio and data Freshness) to improve the quality of service (QoS). This proposal attempts to balance the deadline miss ratio compared with data freshness considering the applications requirements. Indeed, this model allows specifying the desired miss ratio and data freshness for a specific application. To deal with the miss ratio, QMF uses a feedback controller. It includes a controller that periodically measures the miss ratio, calculates the error miss ratio, i.e, the difference between the values of miss ratio desired and the actual measured value and react to correct the error. The QMF also includes a freshness manager that updates more or less sensor data on demand according to the miss ratio control messages and the workload. Moreover, QMF performs control admission to incoming transactions to decrease the overload situations. In order to balance potentially conflicting miss ratio and freshness requirements, it uses a flexible method. Thus, it uses a range of quality of data (QoD) that is compared to the sensor data in order to accept their freshness, if necessary. This can be relaxed the update periods of sensor data and decrease the update workload in case of overload. Moreover, the freshness of sensor data is maintained according to flexible validity intervals.

Cross-layer solution states that parameters of two or more layers can be retrieved and/or changed in order to achieve an optimization objective (Mendes and Rodrigues, 2010). The cross-layering concept has been first proposed for TCP/IP networks, when wireless links were deployed (Srivastava and Motani, 2005).

Many optimization solutions have been based on layers. In this context and in the case of a human motion tracking application (Chen and Ferreira, 2009), Gonçalves et al. (2009) have proposed an architecture based on layers in order to optimize the delay of the sensor data dissemination. Thus, this architecture deals with real-time data dissemination over Internet. It is a four tiers architecture that divides the data dissemination process into four different layers: the WSN Layer for the data acquisition, the Data Layer for the presentation, the Transport Layer for the transmission and the Client Layer for the consumption (see Fig. 6). The WSN Layer collects the data, some times with low level processing, and sends it to the Data Layer. The Data Layer acts as a gateway that interconnects the WSN with others networks. When the Data Layer receives the data from the WSN Layer, it encodes it, in XML form for example, and sends it to the Transport Layer. The Transport Layer, on its round, with time constraints, establishes a single or multiple data transport channel and forwards the data to the Client layer. Finally, the Client Layer receives the data packets and makes some processes in order to improve the service. The Transport Layer use two specifics transport protocols: the Extensible Messaging and Presence Protocol (XMPP) and the Real-Time Protocol (RTP) (Schulzrinne et al., 2003). To deal with real-time transmission constraints, the RTP is used for communication between the Data Layer and the Transport Layer and between the Transport Layer and the Client Layer, in that it provides an improved of the delivery arrival time with UDP protocol. The Jingle XMPP extension is used to establish sessions between the Data Layer and the Transport Layer or between Client Layer and the Transport Layer.

Replication (Plattner and Alonso, 2004) allows managing multiple copies that differ at a given time, but eventually converge to the same values (Gardarin and Gardarin, 1997). The motivations to make a replication are essentially improved performance and increased data availability. Thus, in order to improve the access time to sensor data at any node and to ensure scalable propagation of updates over the network, Mathiason et al. (2008) have proposed a communication scheme in sensor networks using a distributed real-time database with Virtual Full Replication. This scheme is called virtual full replication with adaptive segmentation for sensor networks (ViFuR-ASN). It based on a two-tier approach composed by sensor tier nodes and database tier nodes, where the database tier has more powerful nodes and greater energy supplies. As depicted in Fig. 7, the sensor nodes connect to a suitable database node, forming thus a sub-network, reach the database node by multi-hops and update theirs corresponding readings (sensor data objects) in the database. Each database node has a virtual fully replicated database (an image of full replicated to the database), which stores the data objects belong to the sensor nodes connected to it. The clients in moving can select the database node they want to connect in order to access to sensor data objects. Thus, the search space of client nodes for sensor data is thus limited to the number of database nodes. In the database tier, the database nodes replicate the updates in the same manner. The database nodes are

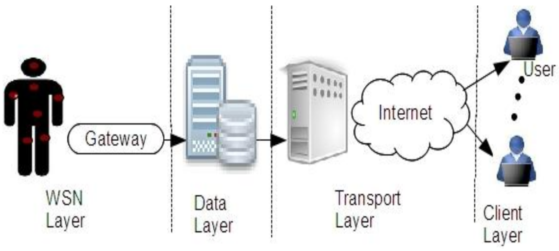


Fig. 6. Illustration of the system architecture.

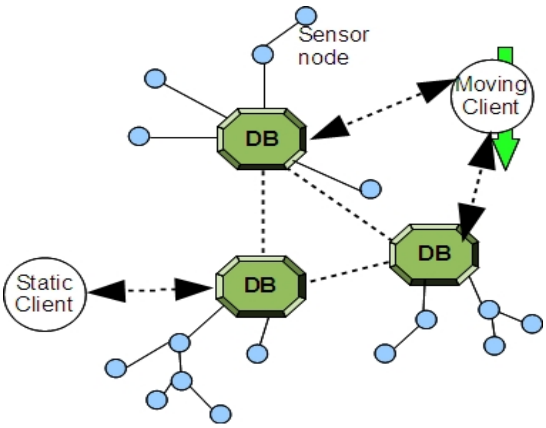


Fig. 7. Illustration of the architecture, using a distributed real-time database with Virtual Full Replication (ViFuR-ASN)

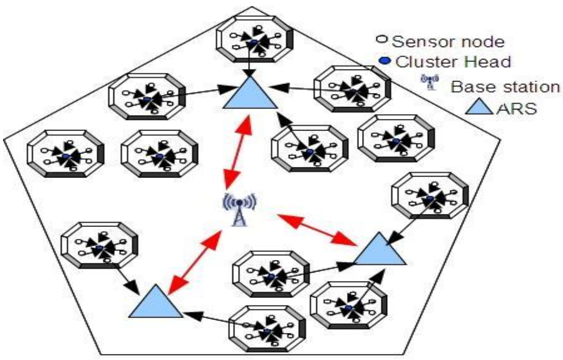


Fig. 8. Illustration of the different components.

composed, among other important modules, the Segmentation Manager that manages the allocations of replicas in the database; the Updater Logger that manage to send local updates to other nodes with the same replicas of the updates data objects; and the Transaction Manager, which executes the transactions. Each sensor object has unique identifier and the location of each sensor is stored as a separate database, allowing thus spatial queries. To deal with timely execution of transactions and the network latency, ViFuR-ASN adopts the same independent updates, virtual full replication, conflict detection and resolution approaches of DeeDS (Andler et al., 2007), but unlike it, ViFuR-ASN uses locally replication that makes values always available and reduces thus the eventual delay. This also allows predictable local access time.

Gupta and Dave (2008) have proposed an architecture based on clusters in order to handle a reliable and real-time placement and dissemination of data in WSNs. As depicted in the Fig. 8, the clusters and cluster-heads (CH) are formed in arbitrary manner and these roles are alternated to better distribute energy use in the network. In each cluster, the sensors communicate data to their CH that aggregates data and reduces thus the size of data to be transmitted to the base station via Action and Relay Stations (ARS) to prevent excessive dissipation of energy. Also, to timely perform actions, the ARS sends data to sink nodes via other ARS. For improving the storage and access of the data, the data centric storage mechanism is used. Also, the data stored in a base station is replicated in the adjacent base stations in order to increase the availability of data and to speed the query processing. For interaction in the network, the queries used are SQL-like queries.

The following tables (Tables 1 and 2) provide a summary to get an overview of the proposed solutions for real-time database management on WSNs.

4. Open issues

The real-time data management for WSNs can be seen as all the necessary resources to meet requirements (storage and exploitation) of real-time applications working on data from sensors. These real-time applications differ from other applications by taking into account time constraints, compliance with which is as important as the accuracy of the result. In other words, these applications should not only deliver accurate results, but within the deadlines. Building mechanisms to meet these kinds of applications is no small matter. Indeed, the techniques previously used in traditional databases are difficult to apply in sensor systems because of their limited resources. Although efforts have been made to deal with these constraints, there are still many challenges that must be addressed to reach best solution. Some identified challenges are given below:

O. Diallo et al. / Journal of Network and Computer Applications 35 (2012) 1013–1021

Table 1
Summary to various kinds of proposals and their features (part 1).

Name of Project/ Mechanism/ Authors	Problematics					
	Type of approach	Basic concepts	Queries expression	Queries evaluation/ metrics	Types of application	Platform
PoTree, 2004	Centralized approach	Real-time spatiotemporal data indexing for fixed sensors	Spatiotemporal queries: spatial criteria for designating a sensor or zone; interval or time instant for the temporal aspect	Real-time update; query on most recent data in RAM Memory	Monitoring applications	Simulation
PasTree, 2005	Centralized approach	Real-time spatiotemporal data indexing for agile sensors	Spatiotemporal queries: ID or spatial criteria for designating a sensor or zone; interval or time instant for the temporal aspect (multi-criteria queries)	Real-time update; query on most récent data in RAM Memory	Monitoring applications	Simulation
StH (SpatioTem-poral-Heat índex), 2006	Centralized approach: derived from PasTree	Real-time spatiotemporal data indexing for agile sensors; management of memory saturation	Spatiotemporal queries: ID or spatial criteria for designating a sensor or zone; interval or time instant for the temporal aspect (multi-criteria queries)	Real-time update; query on most récent data in RAM Memory; Memory saturation	Monitoring applications	Simulation
Bouju et al., 2009	Centralized approach	Spatiotemporal sensor data model for real-time storing	–	–	–	–
Aurora, 2002, 2003	Centralized approach	Data Stream management system	Procedural queries	Average latency	Monitoring applications	–
RTSTREAM, 2006	Centralized approach	Real-time data stream management to deal with deadline of periodic queries over data stream	PQuery: SQL-like language	Deadlines of queries	Real-time application based on data streams	Simulation
Gutierrez and Servigne, 2007; Gutierrez et al., 2007	Centralized approach	Real-time metadata for real-time management of spatiotemporal data	–	Real-time spatiotemporal metadata	Monitoring applications	–
Neto et al., 2004	Distributed approach	In-network processing, concurrency control	Message sending	Logical and timing constraints for data and transactions	–	Analytic simulation
Chagas et al., 2010	Distributed approach	In-network processing, algorithms to negotiate between logical and temporal consistency	SQL-like language	In-networking processing; Logical and timing constraints for data and transactions	Real-time applications	Simulation

Table 2
Summary to various kinds of proposals and their features (part 2).

Name of Project/ Mechanism/ Authors	Problematics					
	Type of approach	Basic concepts	Queries expression	Queries evaluation/metrics	Types of application	Platform
QMF, 2004	Centralized approach	Transactions exécution within their deadlines using fresh data	–	Deadline miss ratio and data freshness metrics	Real-time application	Simulation
Gonçalves et al., 2009	Distributed approach	Real-time sensor data dissemination over Internet based on a four tiers architecture	Text Messages sending	Services applying ; minimizing the transmission delay	Human motion tracking application	–
Mathiason et al., 2008	Distributed approach	communication scheme in sensor networks using a distributed real-time database with Virtual Full Replication	–	In-netwroking exécution ; minimizing the delay of latency ; disponibility; Scalability	–	Simulation
Gupta and Dave, 2008	Distributed approach	Real-time in-networking data storage (replication) and dissemination	SQL-like queries	In-networking processing; Real-time data placement, availability, energy saving	Real-time application	–

First, in these real-time systems, the time constraint is very delicate especially for long-running queries and instant-queries. Most of the proposals are based on the warehousing approach, which, among other drawbacks, can cause a delay on the response times. On the other hand, others proposals focus on the distributed approach that seems, based on its access mode (direct access to sensor nodes), provide little help on the time constraint, however with some risk because there may be a sudden failure of sensors (according to the unstable environment). This may lead to a delay or lack of information that influences the analysis time or even on the blocking of the system. Hence, one can opt for a hybrid approach which allowed one or the other solution depending on the requirements of the application that is the source of the query. This solution will take into account both the time constraints that the risk of failures. Moreover, the connection of the wireless sensor network and Internet will probably cause to the sensor system to be subject to a huge amount of queries. So, system must take account to the scalability.

In data management systems, the inclusion of meta-data that gives the characteristics of the data is essential. Although some proposals include it in their solutions it is not enough. Thus, these solutions may be revised and improved for best met-adata management in order to have fast and accurate routing.

Another thing to consider is the management of simultaneous queries. Indeed, systems connected to a database are most often subject to numerous queries especially for consultation. Hence, a good policy of multiple queries optimizing would be a good deal.

Finally, in real-time systems, for some applications, the accuracy of results may be sacrificed to reduce the response time. Thus, one can think of an approach that allows for an approximation of the queries response.

5. Conclusion and future work

Sensor systems offer new opportunities for building applications that have information and control in areas previously very difficult or even inaccessible. After the acquisition of the information, a major problem of data storage and exploitation arises, particularly for systems that deal with real-time. The researchers have more contributed in the construction of the specific building which is not easy because the data management techniques used in traditional databases are not generally suitable for sensor networks because of their specificities.

This work has reviewed the various solutions for managing sensor data in real-time. To this end, a survey of the oldest to the most recent proposals in this area has been done and it is particularly interested in various stages of data storage, processing and optimization of real-time queries required by the real-time applications.

Many proposals are based on the warehousing approach, considering all treatments in centralized database. Other proposals use a distributed approach, considering the sensor nodes as part of the database. Both have advantages and disadvantages, as depicted earlier.

At the end of this analysis, one can say that a solution that optimizes the real-time query processing while ensuring data availability in case of failure is very important in the management of sensor data in real-time. Thus, a hybrid approach can help to deal with these two challenges. Furthermore, for the time being, there is no proposal that has built-in facilities for approximate query answering. The design of a framework that takes into account these specificities can well meet the requirements to real-time applications based on WSNs.

In future works, we will use a high level modeling such as Stochastic Well-formed Petri Nets (SWN) to study and analyze this wireless sensor network. SWN is a powerful and accurate tool for modeling complex systems with concurrency, synchronization and cooperation. Refined performance indices of WSN (real-time query processing, availability and response time of sensors) can be obtained by simulation or by exact computing.

Acknowledgments

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 a Tecnologia through the PEst-OE/ EEI/LA0008/2011 Project.

References

Abadi DJ, Carney D, Çetintemel Ugur, Cherniack M, Convey C, Lee S, et al. Aurora: a new model and architecture for data stream management. *The VLDB Journal (Springer)* 2003;12(2):120-39.

Akyildiz FI, Melodia T, Chowdhury RK. A survey on wireless multimedia sensor networks*. Journal of Computer networks (Elsevier)* 2007;51(4):921–60.

Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: a survey. *Computer networks (Elsevier)* 2002;38:393–422.

Andler FS, Brohede M, Gustavsson S. DeeDS NG: Architecture, design, and sample application scenario. *Handbook of Real-Time and Embedded Systems', CRC Press*. To appear in June; June 2007.

Arasu A, Babu S, Widom J. The cql continuous query language: Semantic foundations and query execution. *The International Journal on Very Large Data Bases (VLDB) (Springer-Verlag New York, Inc.)* 2006;15(12):121–42.

Arasu A, Babcock B, Babu S, Datar M, Ito K, Motwani R, et al. STREAM: The Stanford Stream Data Manager. *IEEE Data Engineering Bulletin. Citeseer* 2003:19–26.

Baronti P, Pillai P, Chook CVW, Chessa S, Gotta A, Hu FY. Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards. *Journal of Computer communications (Elsevier)* 2007;30(7):1655–95.

Bentley LJ. Multidimensional binary search trees in database application. *Journal of IEEE Transaction on software engineering (IEEE)* 1975:4.

Bonnet Ph., Seshadri P. Device Database Systems. *In: Proceedings of the International Conference on Data Engineering ICDE'99. San Diego, CA: IEEE Computer Society*; 2000.

Bonnet Ph, Gehrke J, Seshadri P. Querying the Physical World. *Journal of Personal Communications, IEEE* 2000;7(5):10-5.

Bonnet Philippe, Gehrke Johannes, Seshadri Praveen. Towards sensor database systems. *In MDM'01: Proceedings of the Second International Conference on Mobile Data Management. London*, UK: Springer-Verlag; 2001. p. 3–14.

Bouju A, Bertrand F, Servigne S, Devogele T, Ray C, Martin H. Gestion de données spatio-temporelles au sein de bases de données capteurs. *Conférence sur les Technologies de l'Information, de la Communication et de la Géolocalisation dans les Systémes de Transports*. 2009.

Buffenoir Christophe. Petite introduction théorique au temps réel. *GNU Linux Magazine France* 2006; 82: 76-79.

Callaway EH. Wireless sensor networks: architectures and protocols. *CRC Press*; 2004.

Carney D, Centintemel U, Cherniack M, Convey C, Lee S, Seidman G, et al. Monitoring streams-a new class of data management applications. *In: Proceedings of the 28th international conference on Very Large Data Bases. VLDB Endowment*, 2002. p. 215–26.

Chagas Lunas Dantas, Lima Eduardo Pereira, Ribeiro Neto Pedro Fernandes. Real-Time Databases Techniques in Wireless Sensor Networks. *Sixth International Conference on Networking and Services. IEEE*, 2010. p. 182–7.

Chen J, Ferreira L. Human Motion Tracking using Wireless Sensor Networks. *Technical Report HURRAY-TR-090407, available online from /www.cister. isep.ipp.ptS*, 2009.

O. Diallo et al. / Journal of Network and Computer Applications 35 (2012) 1013–1021

Chen J, DeWitt David J, Tian F, Wang Y. NiagaraCQ: A scalable continuous query system for internet databases. *Proceedings of the 2000 ACM SIGMOD international conference on Management of data. New York: ACM Press*; 2000. p. 379–90.

Cortes C, Fisher K, Pregibon D, Rogers A. Hancock: a language for extracting signatures from data streams. *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. New York: ACM Press*; 2000. p. 9–17.

Cranor C, Gao Y, Johnson T, Shkapenyuk V, Spatscheck O. Gigascope: High performance network monitoring with an SQL interface. *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*. New York: ACM Press; 2002. p. 623.

DiPippo LC, Wolfe VF. Real-time databases. *Journal of Database Systems Handbook, Multiscience Press (Citeseer)* 1997.

Elnahrawy Eiman, Nath Badri. Context-Aware Sensors. *Wireless Sensor Networks (Springer*) 2004;2920:77–93.

Gardarin O, Gardarin G. Le Client-Serveur.Paris: *Editions Eyrolles*; 1997.

Golab Lukasz, Özsu Tamer M. Issues in data stream management. *ACM Sigmod Record (ACM)* 2003;32(2):5–14.

Gonçalves Joel, Ferreira Luis Lino, Chen Jianxin, Pacheco Filipe. Real-Time Data Dissemination for Wireless Sensor Networks using XMPP. 2009.

Gupta S, Dave M. Real time approach for data placement in wireless sensor networks. *International Journal of Electronic Circuits System* 2008;2(3): 132–9.

Gutierrez C, Servigne S. Métadonnées spatio-temporelles temps réel. *Journal de INGENIERIE DES SYSTEMES D INFORMATION (HERMES SCIENCE)* 2007;12(2): 97–119.

Gutierrez Rodriguez C, Servigne S, Laurini R. Towards real-time metadata for network-based geographic databases. *Proceedings of ISSDQ2007, 5th International Symposium of Spatial Data Quality* 2007. p. 13–15.

Idoudi Nizar, Duvallet Claude, Sadeg Bruno, Bouaziz Rafik, Gargouri Faiez. How to model a real-time database?. *Proceedings of 12th IEEE International Symposium on Object-oriented Real-time distributed Computing (IEEE ISORC'2009)*. Tokyo, Japan: IEEE, 2009a. p. 321–5.

Idoudi Nizar, Louati Nada, Duvallet Claude, Bouaziz Rafik, Sadeg Bruno, Faiez Gargouri A. Framework to Model Real-Time Databases. *International Journal of Computing & Information Sciences* 2009b;7:1–11.

ISO19115. An International Metadata Standard for Geographic Information. */http:// grdc.bafg.de/servlet/is/2376/S.* 2006; accessed March 2011.

Kang Kyoung-Don, Son Sang H, Stankovic John A. Managing deadline miss ratio and sensor data freshness in real-time databases. *Journal of IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING. IEEE* 2004:1200–16.

Lam YK, Kuo WT. Real-Time Database Systems: Architecture and Techniques Netherlands: *Springer;* 2001.

Leite CRM. Linguagem de Consulta para Aplicações em Tempo-real. *Dissertaço, Programa de Pos-Graduação em Engenharia Elétrica*.Campina Grande: Universidade Federal de Campina Grande; 2005.

Madden Samuel, Franklin Michael J, Hellerstein Joseph M, Wei Hong. TAG: A tiny aggregation service for ad-hoc sensor networks. *In OSDI. Citeseer*, 2002.

Mainwaring A, Culler D, Polastre J, Szewczyk R, Anderson J. Wireless sensor networks for habitat monitoring. *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. New York: *ACM Press*; 2002. p. 88–97.

Mathiason G, Andler FS, Son HS. Virtual full replication for scalable and adaptive real-time communication in wireless sensor networks. *The Second International Conference on Sensor Technologies and Applications. IEEE*, 2008. p. 55-64.

Mendes LDP, Rodrigues JJPC. A survey on cross-layer solutions for wireless sensor networks. *Journal of Network and Computer Applications; Elsevier* 2010.

Neto Ribeiro PF, Fernandes YYMP, Mendes Neto FM, Leite CRM, Costa CM. Rede de Sensores: O Estado da Arte. EPOCA-Escola Potiguar de Computação e suas Aplicaçãoes. 2008. p. 1–16.

Neto Pedro Fernandes R, Perkusich Maria LB, Perkusich Angelo. Real-Time Database for Sensor Networks. *In: Proceedings of the 6th International Conference on Entreprise Information Systems*. 2004. p. 599–603.

Noel G, Servigne S. Indexation multidimentionnelle de bases de donnees capteur temps-réel et spatiotemporelles. *Journal of INGENIERIE DES SYSTEMS D'INFORMATION (HERMES SCIENCE)* 2005;10(4):59–88.

Noel G, Servigne S, Laurini R. Spatial and Temporal Information Structuring for Natural risk monitoring. *Proceedings of the GIS Planet*. Lisbonne, 2005b.

Noel G, Servigne S, Laurini R. The Po-tree: a Real-time Spatiotemporal Data Indexing Structure. *Journal of Developments in Spatial Data Handling (Springer)* 2005a. 259–27.

Nöel Guillaume, Servigne Sylvie. Structuration de données spatio-temporelles temps-réelles: vers la gestion de la saturation de base de données. *In: Proceedings of INFORSID 2006.* Tunisie, 2006.

Nöel Guillaume, Servigne Sylvie, Laurini Robert. Real-time spatiotemporal data indexing structure. *Proceeding of 2004 AGILE 7th Conference on Geographic Information Science. Heraklion: Citeseer*, 2004a. p. 261–8.

Nöel Guillaume, Servigne Sylvie, Laurini Robert. The Po-tree: a soft real-time spatiotemporal data indexing structure. *Developments in Spatial Data Handling. SDH04, 11th International Symposium on Spatial Data Handling. Springer Verlag*. 2004b. p. 259–70.

Plattner C, Alonso G. Ganymed: Scalable replication for transactional web applications. *Journal of Middleware 2004 (Springer)* 2004:155–74.

Ramamritham K. Real-time databases. *Journal of Distributed and Parallel Databases (Springer)* 1993;1(2):199–226.

Sacks L, et al. The development of a robust, autonomous sensor network platform for environmental monitoring. *Sensors and their Applications XXII (Citeseer)* 2003.

Schulzrinne H, Casner S, Frederick R, Jacobson V. RFC3550: RTP: A Transport Protocol for Real-Time Applications 2003.

Servigne S, Noel G. Real time and spatiotemporal data indexing for sensor based databases. *Journal of Geospatial Information Technology for Emergency Response (Taylor & Francis Group)* 2008;6:123–42.

Servigne S, Lesage N, Libourel T. Spatial data quality components, standards and metadata. Spatial data quality: an introduction. *International scientific and technical encyclopedia* 2006:179–208.

Silva FA, Braga TRM, Ruiz LB, Nogueira JMS. Tecnologia de Nos Sensores Sem Fio Contrôle & Instrumentação; agosto 2004;92:76–87.

Srivastava V, Motani M. Cross-layer design: a survey and the road ahead. *Journal of Communications Magazine, IEEE (IEEE)* 2005;43(12).

Ulmer C, Yalamanchili S, Alkalai L. Wireless distributed sensor networks for in-situ exploration of mars. *Work in Progress for NASA Technical Report (Citeseer)*, 2003.

Wei Y, Son SH, Stankovic JA. RTStream: Real-Time Query Processing for Data Streams. *IEEE Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'06).* Gyeongju, Korea: IEEE Computer Society; 2006.

Zhu Yunyue, Shasha Dennis. Statstream: Statistical monitoring of thousands of data streams in real time. *Proceedings of the 28th international conference on Very Large Data Bases. VLDB Endowment*, 2002. p. 358–369.

Chapter 3

Distributed Database Management Techniques for Wireless Sensor Networks

This chapter consists of the following article:

Distributed Database Management Techniques for Wireless Sensor Networks

O. Diallo, J. J. P. C. Rodrigues, M. Sene, J. Lloret

IEEE Transactions on Parallel and Distributed Systems, ISSN: 1045-9219, 19 Aug. 2013, IEEE Computer Society, DOI: 10.1109/TPDS.2013.207 (in press).

According to the 2012 Journal Citation Reports published by Thomson Reuters in 2013, this journal scored ISI journal performance metrics as follows:

ISI Impact Factor (2012): 1.796

ISI 5-Years Impact Factor (2012): 2.031

Transactions on Parallel and Distributed Systems

Distributed Database Management Techniques for Wireless Sensor Networks

Ousmane Diallo, Joel J. P. C. Rodrigues, *Senior Member, IEEE*,
Mbaye Sene, and Jaime Lloret, *Senior Member, IEEE*

Abstract— In sensor networks, the large amount of data generated by sensors greatly influences the lifetime of the network. In order to manage this amount of sensed data in an energy-efficient way, new methods of storage and data query are needed. In this way, the distributed database approach for sensor networks is proved as one of the most energy-efficient data storage and query techniques. This paper surveys the state of the art of the techniques used to manage data and queries in wireless sensor networks based on the distributed paradigm. A classification of these techniques is also proposed. The goal of this work is not only to present how data and query management techniques have advanced nowadays, but also show their benefits and drawbacks and to identify open issues providing guidelines for further contributions in this type of distributed architectures.

***Index Terms*—Distributed database management; wireless sensor networks; distributed storage; query techniques, data reduction techniques; query optimization.**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are composed of a large number of devices, called sensornodes, which are able to sense, process, and transmit information about the environment on which they are deployed. These devices are usually distributed in a geographical area in order to collect information for users interested in monitoring and controlling a given phenomenon. This information is transferred to a sink node in order to be accessible by remote users through generally application-level gateway, e.g. global sensor network (*GSN*) [1],[2], [3]. To obtain the data, these applications should also provide supports of efficient queries, which allow communication with the network [4],[5],[6] (see Fig. 1 for an illustration of a WSN).

Manuscript received ……

Ousmane Diallo is with the Instituto de Telecomunicações, University of Beira Interior Covilhã, Portugal and with the Department of Mathematics and Informatics, UCAD, Senegal. E-mail: Ousmane.diallo@it.ubi.pt.
Joel J. P. C. Rodrigues is with the Instituto de Telecomunicações, University of Beira Interior, Covilhã, Portugal. E-mail: joeljr@ieee.org.
Mabye Sene is with the Department of Mathematics and Informatics, UCAD, Senegal. E-mail: mbaye.sene@ucad.sn.
Jaime Lloret is with Integrated Management Coastal Research Institute, Universidad Politécnica de Valencia, Spain. E-mail:jlloret@dcom.upv.es.

In wireless sensor networks, the sensor nodes are battery powered and are considered intelligent with acquisitional, processing, storage, and communication capacities [7],[8]. However, these resources are generally very limited, especially in terms of storage and energy, and the sensor nodes activities are sometimes not negligible in energy consumption [9], [10]. One of the most used techniques to save power is to activate only necessary nodes and to put other nodes to sleep [11]. Some authors have studied how a 3 dimensional sensor field can be efficiently partitioned into cells in order to save energy [12].

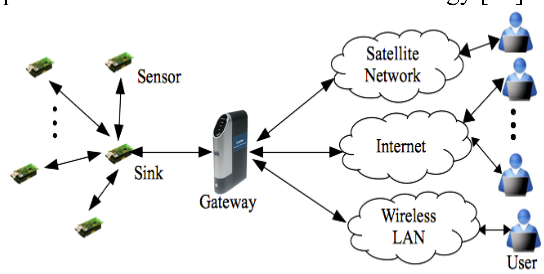


Fig. 1. Illustration of a wireless sensor network architecture.

Sensors can be placed anywhere there is data that should be collected, what makes information omnipresent. Consequently, systems based on sensor networks are increasingly common in many areas of the knowledge, giving rise to several flavors of WSNs [13], [14]. These numerous WSNs have allowed the development of many applications [15], [16], [17]. In addition to data gathering [18] and data replication issues [19], in such applications, a database-oriented approach of WSNs has proven to be useful in order to manage the large amount of data generated by the sensors. According to this approach, a WSN is viewed as a distributed database where sensor nodes are considered as data sources with sensed data stored in the form of rows of a relation distributed across a set of nodes in the network [20], [21]. This database-oriented approach has motivated the design of WSN data acquisition with two fundamental objectives [22]: similarly to traditional database systems, a WSN database should provide SQL-like abstractions so that nodes can be easily programmed for simple data sensing and collection. In addition, the data collection process should minimize the energy consumption in the network.

The main goal of distributed database management on WSNs is to support the management of the huge amount of sensed data in an energy-efficient manner [23]. In fact, research into sensor hardware has shown that the energy depletion in the network is mainly due to the data communication tasks among the nodes [24]. To deal with this problem, various data reduction techniques exist [25], [26], including *data aggregation* [27], [28], *packet merging*, *data compression techniques* [29], [30], *data fusion*, and *approximation based techniques* [31], [32]. The data aggregation techniques consist to perform data aggregations (e.g., MAX, AVG etc.) at intermediate nodes between the source nodes and the sink node. The packet merging combines multiple small packets into a big one, without considering the semantics and the correlation between the packets. The data compression techniques are also used to reduce the amount of data transmitted between the nodes, but they involve data encoding at the source nodes, data decoding at the sink node. The data fusion techniques refer to more complex operations on a data set and are usually used in multimedia data processing [33]. The approximation based techniques use statistical techniques to approximate the queries results. These techniques provide, among other advantages, the reduction of the size of the transmitted data, the communication tasks, the network load, and the data transmission time.

The aim of this paper is to show how distributed database techniques are adapted to wireless sensor networks in order to improve the management of the great amount of sensed data in an energy-efficient way by presenting and classifying the most recent and relevant proposals of distributed database management on WSNs. Moreover, a discussion and open issues on distributed database management techniques for wireless sensor networks are identified in order to facilitate further contributions.

The remainder of this paper is organized as follows. Section II presents the essential conceptual features of distributed data storage and querying in WSNs, while protocols and techniques used on the studied proposals on distributed data management in WSNs are exposed in Section III. Section IV discusses the techniques used on the studied approaches and proposes some open research issues. Finally, Section V concludes the paper and pinpoints further research works.

II. BACKGROUND

As in traditional database systems, the sensor databases try to create an abstraction between the end-users and the sensor nodes. This abstraction aims to permit the users to only concentrate on the needed data to be collected rather than bothering with the complexities of mechanisms deciding how to extract data from a network [27], [34]. As such, the sensor databases have been subject to two main approaches to data storage and query in WSNs [35]: the *warehousing* approach and the *distributed* approach.

1. In the *warehousing* approach, the sensors act as collectors. The data gathered by sensors are periodically sent to a central database where user queries are processed. This model is the most used one in data storage and query processing. However, it has some drawbacks, such as eventually wasting resources and creating a bottleneck with an immense amount of transmitted data. This approach is unsuitable for real-time processing.
2. The *distributed* approach is the alternative, where each sensor node is considered as a data source, and then the WSN forms a distributed database where the sensed data are in the form of rows with columns representing sensor attributes [20], [21]. In this second approach, the sensed data are not periodically sent to the database server. They remain in the sensor nodes and some queries are injected in the network through the base station. These queries are disseminated into the network according to the routing techniques as per [36], [37], and the sensors, thanks to their processing and storage capabilities, process them. The sensors send their data to their parent nodes whenever they correspond to the query requirements. The parent nodes combine this coming data with their own data and transmit to their parent nodes and so on until the data reaches the gateway. This approach that consists to process the data inside the sensor nodes themselves is called in-network processing and it reduces the amount and size of transmitted data and the latency [38]. According to the scope of this work, an illustration of a distributed database on WSN may be seen in Fig. 2.

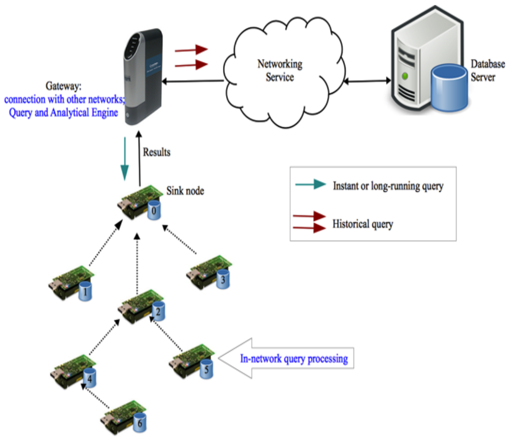


Fig. 2. Illustration of a distributed database architecture for wireless sensor networks.

As largely detailed in [39], there are four essential methods to design a distributed data management system: *in-network processing*, *acquisitional query processing*, *cross-layer optimization*, and *data-centric data/query dissemination*.

A. In-network Processing

The in-network processing technique [38], [40], [41] generally includes the different types of operation that are traditionally done on the server, for instance, aggregations to inside the sensor nodes themselves. It is generally used, as its name indicated, to process sensing values inside the network nodes so as to filter and reduce the huge and needless data.

B. Acquisitional Query Processing

The acquisitional query processing [20] permits to minimize energy consumption in the network by reducing the number of sensor nodes participating in the query processing. This reduction is done by expressing in the query when or what sensors to sample.

C. Cross-layer Optimization

Unlike the traditional computer networks in which layers in the conventional OSI model are separated and isolated, the cross-layer optimization [42], [43] permits to combine information available on these different layers and profit from this information sharing. For instance, in wireless sensor networks the routing takes care of, among others, the quality of service (QoS) parameters of the network, network connectivity, the power available on the node, and the network lifetime [44]. In traditional computer networks the routing is done by the network layer only considering the destination address of a packet.

D. Data-centric data/query dissemination

In contrast with traditional networks, nodes in WSNs usually do not have a single identifier because of data-centric [45], [46], [47] nature of sensor applications as well as the large number of sensors deployed. Generally, applications are not interested in specific sensors, but rather in data, which they generate. For example, a query as *"which is the temperature measurement of the sensor with the ID XXXX"* does not have much interest for a sensor application, but a query like *"in which region, sensors measure fewer than 7ºC"* is more significant. Routing protocols must take these characteristics into account.

In the next section, the most relevant proposals of distributed data storage and query management techniques designed for WSNs are going to be discussed. This discussion will be done by classifying each proposal into the four above-mentioned approaches.

III. DISTRIBUTED DATA STORAGE AND QUERY MANAGEMENT TECHNIQUES FOR WSNs

There is a lot of research addressing various aspects of data management in sensor networks. However, generally, the main goals are the transparent access of the end-users to the sensor nodes for retrieving required relevant sensor data as well as the improvement of the network's lifetime.

A. In-Network Processing

After receiving one query, the query processor analyses it and generates an optimized query execution plan that consists of the best possible execution of the query inside the sensor network. While the different solutions to perform in-network processing may differ, the objective is to save energy by reducing communication. The in-network processing technique can be divided into two sub-categories, *aggregation based techniques* and *approximation based techniques* [48].

1. Aggregation based Techniques

Since wireless communication involves more energy consumption than the sensing and processing operations, aggregating the data through intermediate nodes results in less message transmissions. Therefore, the network lifetime is improved with less energy consumption.

As part of the pioneering researches on distributed database on WSNs, TinyDB [20] and COUGAR [21], [34], [49] are the first to adopt a database query optimization technique for WSNs based on the data acquisition declarative approach.

The TinyDB project [50] was developed for networks based on the TinyOS operating system [51]. It is a distributed query processor for sensor networks that incorporates acquisitional techniques. Through an interface, the user chooses what data he wishes to acquire. The query is decomposed by a query processor and distributed across the network. The sensor nodes collect, filter and aggregate the data and respond to the user query.

The interrogation of the sensors is based on the relational model and queries are specified using an SQL-like query language [20]. The sensor data are form of tuples that conform to a predetermined schema. For example, tuples produced by a temperature sensor may be of the form <*sensorId, location, temperature, timestamp*>.

Physically, the sensor tuples belong to a sensors table, which is partitioned across all of the devices in the network, with each device producing and storing its own readings. Queries are formulated on a virtual table that is logically formed by the sensor tuples horizontally partitioned in the network. Unlike traditional queries that focus on the current state of a database, these queries are often continuous with as function to continually run in order to inform the applications of changes recorded by the sensors. Results of queries stream to the root (base station) of the network by multi-hop topology.

TinyDB includes support for grouped aggregation queries (e.g. *MIN*, *MAX*, *SUM*, *COUNT*, *AVERAGE*, etc.), as sensor readings flow up the communication tree called a semantic routing tree (*SRT*), they are aggregated by intermediate nodes that contain relevant information for the query (See Fig. 3). This in-network aggregation reduces the huge quantity of data that must be transmitted through the network, preventing the bottleneck to the root node and increasing the lifetime of the network.

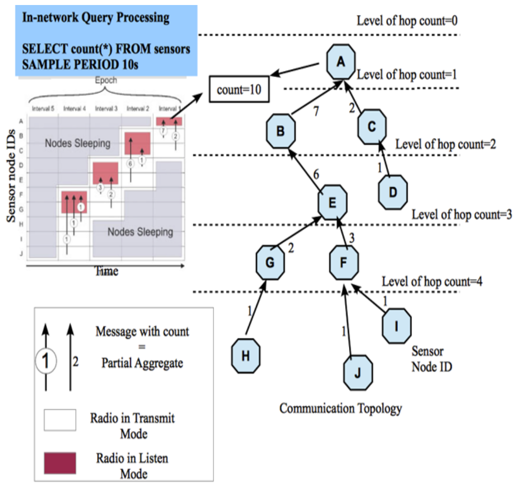


Fig. 3. Aggregation steps of sensor readings during an epoch using interval-based communication [38].

The Cougar project [52] is a platform for distributed query processing. To deal with in-network processing in this platform, they use a clustered approach. A network is composed of several clusters, each of them managed by a cluster head. The child nodes that belong to clusters send periodically their readings to the corresponding cluster head, which then aggregates the received readings and forwards the computed result toward the Front End of the network. This Front End is a query optimizer, located at the gateway node, which generates optimized distributed query processing plans after receiving user queries. Furthermore, in this architecture, each node embeds a query layer. The query layer [21], [49] is a query proxy between the network layer and the application layer, which process queries (See Fig. 4). Additionally, Cougar carries out packet merging by aggregating several packets into one. This increases the lifetime of the network, since sending multiple small packets is more expensive than sending one larger packet.

Like TinyDB, Cougar adopts a declarative queries approach [49] to in-networking processing. This approach allows users and application queries a transparent access to sensor nodes. Thus, Cougar uses an efficient catalog management, query optimization, and query processing techniques to abstract the user from the physical details of contacting the relevant sensor nodes, which process the sensor data and send the results to the user. The queries are specified using an SQL-like query language and the sensor data are form of records with several fields included information about the sensor node (e.g., id, location, etc.), a timestamp, the sensor type (e.g., temperature, light, etc.), and the value of the reading. The sensor network is considered as a widely distributed database system consisting of multiple tables of different types of sensors.

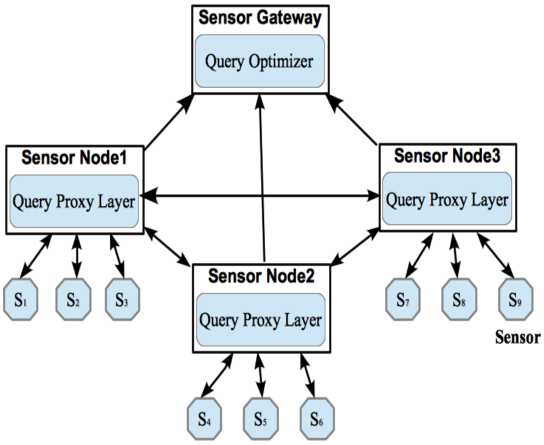


Fig. 4. Illustration of Cougar Architecture.

Cougar claims to be designed for WSNs, while it was deployed on PDA-class devices that had significantly larger processing power than WSNs and could even run Windows CE and Linux [20]. So, it does not take into account the power and computational constraints of sensor nodes.

TiNA (Temporal Coherency-Aware In-Network Aggregation) [53], [54], [55] is an improvement over TinyDB. It uses in-network processing to increase the lifetime of the network. It is freshness-aware as it adds a new clause, *VALUES WITHIN tct,* in the specification of the query aggregation syntax of TinyDB, which indicates the temporal relaxation degree allowed by the user or the network. In TinyDB, the readings are transmitted at fixed interval, but TINA transmits the sensor node reading only if that reading differs from the last recorded reading by more than the accepted tolerance *tct*(See Fig. 5).

In [54], TiNA is improved by designing a semantic routing tree for sensor networks with a main objective the reduction of the size of transmitted data. For that, by performing in-network aggregation, the reduction of the number of groups is adopted when a node performs Group-By query by clustering the sensor nodes belong to the same group along the same path. This approach is called group-aware network configuration method. This technique can certainly reduce the size and number of messages circulating in the network but at the expense of some consistency of data transmitted.

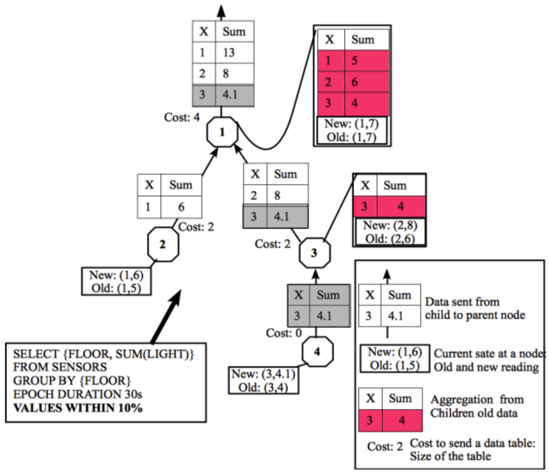


Fig. 5. In-network aggregation query using TINA [53].

The authors in [56] proposed a set of algorithms to minimize the overall energy consumption of the sensor nodes by considering a real time scenario where the raw data gathered from the source nodes must be aggregated and transmitted to the sink within a specified latency constraint. This work is particularly important for applications requiring a prompt delivery of the information to the sink. However, it does not address the problem of constructing the underlying aggregation tree.

The main purpose of data management in WSN is to allow transparent access to sensed data, as well as, to increase the network lifetime. To these ends, the authors of [57], [58] propose an adaptive algorithm, called *ADAGA* (Adaptive AGgregation Algorithm for sensor networks), for processing in-network aggregation in WSNs. In fact, in order to reduce the amount of data transmitted between sensor nodes, *ADAGA* performs in-network aggregation. The key idea is to aggregate the sensed data progressively in each node it passed through. This will further reduce the data traffic, the energy consumption, and the memory usage in the network. The algorithm also supports the packets replication in order to reduce the packets losses and allows the approximation of sensed values from collected data. Furthermore, a data model for data streams and a declarative SQL-like query language named *SNQL* (Sensor Network Query Language) for WSNs are provided.

The context aware system paradigm means the capability of a system to adapt according to a rapidly changing context in which it is. That is useful because the system should adapt to context changes in order to react rapidly. In this context, the authors of [59] propose a framework in the context aware architecture (See Fig. 6). This framework exploits a distributed query processor approach for integrating wireless sensor networks. This proposed architecture is based on the *MaD-WiSe* system [60]. This latter includes a set of modules running on the WSN nodes; the *MaD-WiSe* network side and a set modules running on the base station; the MaD-WiSe context information provider. The *MaD-WiSe* network side implements an in-network distributed data stream management system that acts as server; while the *MaD-WiSe* context information provider offers access to the sensor services. The queries are specified using an SQL-like query language named *MW-SQL* that allows users to express queries to manipulate, temporal aggregate, filter, and organize sequences of tuples generated by the sensors. Through the *MaD-WiSe* system interface the user chooses what data he wishes to acquire. An optimized distributed query execution plan is generated and disseminated in the network by the query manager. The latter receives the results of the query stream obtained from in-network query execution in an on-line fashion.

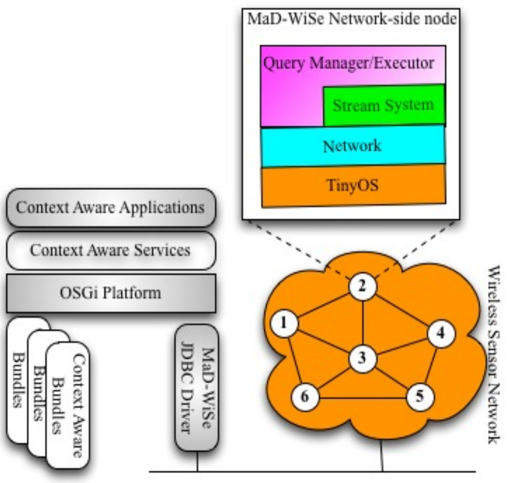


Fig. 6. Context aware architecture using in-network query processing.

Similar to most of the proposed solutions about distributed query processing in WSNs, in [61] the authors propose an in-network aggregate query processing. But rather than using aggregate operators in sensor nodes to reduce the number of transmitted messages, they propose a mechanism that allows queries to share intermediate results together in order to reduce the number of messages transmitted. The main idea assumes that when the sink receives multiple aggregate queries it propagates them via a routing tree. Thus, with this set of query tree, it will decide how to determine a set of backbones and non-backbones and where each non-backbone is allowed to access intermediate results from the backbones in order to reduce the total number of transmitted messages. For that, according to a cost function, they derive a reduction graph and propose two algorithms. First, they propose a heuristic algorithm *BM* (standing for Backbone Mapping), which is based on the reduction graph; it determines the set of backbones and mapping relationships between backbones and non-backbones. Second, they propose an algorithm named *OOB* (standing for Obtaining Optimal Backbones) to obtain the optimal backbone set. Furthermore, they propose a maintenance mechanism for dealing with dynamic scenarios as queries could submit or leave.

Generally, because of wireless sensor network application types (e.g., military battle field, volcano, etc.), sensor nodes are scattered in hostile environments and may relay sensitive data. Moreover like in traditional computer networks, sensor networks are particularly vulnerable to several key types of attacks that can be performed in a variety of ways [62], [63], [64], [65], as Sybil attack [66], traffic analysis attacks [67], node replication attacks [68], and so on. Therefore, security concerns should be addressed efficiently in order to provide reliable communication between nodes and to collect reliable data from the network.

The work in [69] is one of the pioneering wireless sensor network security paradigms. It analyzed the resilience of aggregation techniques for cluster based WSN and proposed a mathematical framework for formally evaluating the security for aggregation, allowing them to quantify the robustness of an aggregation operator against malicious data. However, one can argue that the one-level homogeneous aggregation model is simple to represent real sensor network deployments.

In [70], the authors present a secure information aggregation technique (*SIA*) that helps to defend against a type of attack called the *stealthy attack*. In a *stealthy attack*, the attacker tries to provide incorrect aggregation results to the user without he/she knows that the results are incorrect. Therefore in order to prevent from that type of attack, the mechanism in [70] aims to ensure that if a user accepts an aggregate value as correct, then there is a high probability that the value is close to the true aggregation value. Otherwise, if the aggregate value has been altered, the incorrect results should be rejected with high probability.

The proposal in [71], [72] proposed an energy-efficient and secure pattern-based data aggregation (*ESPDA*) protocol for wireless sensor networks. *ESPDA* is intended for hierarchy- based sensor networks, then a cluster-head first requests sensor nodes to send the corresponding pattern code for the sensed data. If multiple sensor nodes send the same pattern code to the cluster-head, only one of them is permitted to send the data to the cluster-head. ESPDA is secure because it does not require encrypted data to be decrypted by cluster-heads to perform data aggregation.

2. *Approximation based Techniques*

Here statistical techniques like approximation, linear regression, probabilities, etc. are used to approximate data or query answers.

BBQ [73] improves upon TinyDB as it includes certain statistical modelling techniques to answer queries about the current state of the sensor network. The BBQ query system creates models that provide approximated results with probabilistic confidence interval, improving thus the lifetime of the network. Through an SQL like query that includes error tolerances and target confidence bounds, the user chooses what data he/she wants to acquire. The query is parsed by the query-processor, an energy-efficient observation plan is generated, and a time-varying multivariate Gaussians model, which includes correlations and statistical relationships between sensor readings on different nodes, is used to estimate its answer. Unlike TinyDB and COUGAR that interrogate all sensors every time a query is injected into the network, in this query processing, sensors are only solicited to update the data to refine the model if the model itself can't satisfy the query with acceptable confidence.

In [74] the authors adopt a replication solution. However, their proposal is based on the hypothesis of the random nature of failures in wireless sensor networks. In fact, in wireless sensor network areas, due to generally harsh environments of deployment and the resource limitation, sensor nodes are always subject to random failures [38]. For these reasons, in [74] the authors propose a distributed energy-efficient replica placement for increasing data availability and prolonging the network lifetime. The basic idea is, taking into account probabilistic node failures, they compute (based on probabilistic equations) sufficient data replica of nodes with minimal communication cost between nodes such that the whole network data could be reached after a failure. The replication can protect from failures and provide data availability but it should be completed by an efficient data update policy in order to ensure the data freshness.

To satisfy as much as possible the end user requirements as well as to improve the network lifetime, the distributed top-k query processing [75] computes, in a quick and efficient manner, the subset of most relevant answers rather than the all answers. This prevents the transmission of irrelevant answers and minimizes the power cost of retrieving the huge amount of values.

Like the previous work, the distributed processing of probabilistic top-k queries in wireless sensor networks [76], [77] proposes, with bounded rounds of communications in cluster-based wireless sensor networks, three suite of energy-efficient algorithms: sufficient set-based (*SSB*), necessary set-based (*NSB*) and boundary-based (*BB*). Instead of transferring the huge amount of sensor data from the network to the end users, these algorithms return the subset of most relevant data answers efficiently with a constant round of data communications according to a probabilistic weight. This permits to minimize the cost of retrieving all huge values and just transfer relevant answers. Moreover, for better minimizing the communication and energy overhead, this solution proposes also an adaptive algorithm that dynamically switches among the three algorithms based on their estimated costs.

The work in [78] improves the top-k query in which the algorithm tries to find the *k* nodes with highest readings among the sensor nodes, by implementing top-k query in duty-cycled WSNs

(*DC-WSNs*). For that, this work solves the underlying data accessibility and network connectivity problems in *DC-WSNs* by proposing a mechanism named *DCDC-WSNs,* where data replication (*DR*) is applied into *DC-WSNs* and the whole combined with a sleep scheduling algorithm named connected k-neighborhood (*CKN*). Thus, the implementation of top-k query in *DCDC-WSNs* can achieve very high query data accessibility at the cost of low total energy consumption and top-k query response time.

As much of the WSN research centers around increasing network lifetime [79], *ENERGY** [80] based on *ENERGY* (Energy Efficient Rate Governed Yardstick) provides an approximate but effective solution to minimize the energy consumption in WSNs. Thus, based on the information about the complete network topology and the Euclidean distance as an estimate to the hop count between two nodes, *ENERGY** provides the optimal placement of the data transformation function which impacts on the energy consumption on data transmission. However, considering this solution for virtual nodes, the proposed algorithm uses the sink node that requires to know the locations and bit rates of all the sources to map the virtual nodes to the real nodes in the network.

In [81] an energy-efficient and accurate estimate is proposed. Hence, the proposal based on a distributed algorithm for in-network data processing provides a new cost function, which is robust to node failure and impulsive noise. The main strategy is to pass around the network a parameter estimate, and along the way small adjustments to the estimate are made by each node based on its local measured data.

Like their predecessors, in [82] the authors propose an efficient approximation algorithm, which improves the network lifetime. The main idea is to optimize the communication cost by performing in-network data aggregations of approximate coefficient. Thus, each node transmits the approximate coefficient upward after it compresses and aggregates the child node's coefficient. Finally, each root node of concerned splay tree obtains the approximate coefficient set about its complete covered area. Therefore, the sink node can query any position of interested area through the root nodes. The approximate is performed by using multiple linear regression models.

B. *Acquisitional Query Processing*

Like in traditional database systems, in distributed database systems, where the requested data might be stored in small fragments around the whole network, the complexities of mechanisms used by the query processor to efficiently extract the relevant data from the network are completely transparent to the end-users.

In WSNs, the query processor is charged, among other tasks, to generate an optimized query execution plan that defines how a query should be executed in an energy-efficient way. This optimization is performed by minimizing the activities of sensor nodes, principally by reducing the data transmission and sampling the sensors that participate in a query processing.

TinyDB [20] is a distributed query processor for sensor networks that has first introduced the management of sensor sampling, called as *acquisitional query processing*. It incorporates a metadata management system that supports optimizations of query processing. In fact, to manage sampling of the sensors for a particular query processing, metadata such as information about the costs of processing and delivering data, the necessary time and energy for that sampling, etc. are periodically copied from the nodes to the root and used by the query optimizer. TinyDB also provides extensions to SQL to formulate queries evaluated when the event specified in the request is made. The main objective is thus avoiding the sending of measures that are not relevant. TinyDB also includes the possibility of using time windows applied to a particular sensor. A time window contains the measures recently performed by the sensor. Moreover, a multi-query optimization on event-based queries is taken into account in order to reduce costs due to transmission and sensor sampling.

Replication [83] allows managing multiple copies that differ at a given time, but eventually converging to the same values [84]. The motivations to make a replication are essentially improved performance, increased data availability and eventually prevent from failures [32], [85], [86].

According to the harsh energy constraints of sensor networks, the authors of [41], [87], [88] propose a hierarchical architecture for in-network data acquisition and replication in mobile sensor networks called *SenseSwarm*. It is composed of two levels: the *perimeter nodes* that perform the data acquisition in energy efficient manner, and the *core nodes*, which are physically and logically strong in order to manage the storage and replication of sensed data (See Fig. 7). In order to increase the fault-tolerance and the availability of the system, a data replication algorithm (*DRA*) is proposed. This algorithm is based on a vote that consists of deciding which set of neighbors will participate to the most energy-efficient replication strategy. Additionally, the *DRA* is extended with a spatial-temporal in-network aggregation strategy based on minimum bounding rectangles. This work leads to a hierarchical data replication algorithm (*HDRA*), which allows an approximate answer.

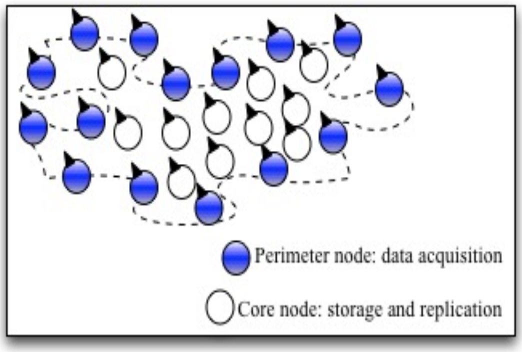


Fig. 7. Framework for the acquisition and storage of spatio-temporal data in mobile sensor networks.

Like most of the above works, in [22] minimizing the energy consumption is a main objective. For that, these works use selectivity awareness technique to optimize the monitoring queries in sensor network. An algorithm called *PDT* (pocket driven trajectories) based on the main features of monitoring queries is used for optimizing the network lifetime. The idea is the use of the data acquisition technique to optimize the communication overhead by sampling sensor nodes (notion of pocketed node participation) for processing a given query, after the setting up of efficient communication paths between the clustered sensor nodes and the base station. This discards the non-selected nodes in the data collection and optimizes the energy consumption. Furthermore, the *PDT* is extended for context aware system. Therefore, it continuously adapts the data collection paths according to changing of node participation and environmental conditions.

In [89] the authors perform a distributed data processing by adapting artificial neural-networks algorithms for wireless sensor networks. For that, three types of cluster based architectures are presented in order to carry out distributed computation and storage, auto-classification of sensor data. The auto-classification of sensor data is performed by using different sensitivity threshold. Thus, depending on the level of details needed at the moment, the corresponding device unit can be queried depending on the level of the sensitivity threshold used to classify the data. The aims of this work are to provide data robustness and improving the network lifetime by optimizing the communication costs and energy savings.

Similarly to TinyDB and Cougar, in [90] the authors have designed a distributed query processing system called *SSDQP*. Through a high-level user interface, the user formulates what data he/she wishes to acquire. The query is decomposed, optimized and distributed by the optimizer across the network. The sensor nodes collect, process the data, and respond to the user query. In this work, each node runs a time-triggered query engine and the queries are scheduled in a distributed manner among the sensor nodes. For improving the network lifetime, the optimizer is parameterized. Thus, according to the user's need, it makes a trade-off between accuracy of time synchronization and consumed energy by choosing an optimized network tree topology. Furthermore, the system performs a synchronized merge operation to minimize the transmitted packet and thus reduce the communication overhead.

Corona [91] system improves upon the *SSDQP* [90] system and the proposal in [92] as it is an in-network distributed query processor that allows sharing sensor readings between several user declarative queries. Through a declarative query interface, several users can choose what sensor data they wish to acquire. With dynamic multi-query execution capability, the query processor can execute concurrently these various queries with different start-times, epochs, and lifetimes. Furthermore, Corona is freshness-aware since it permits to reuse the previous sensor readings from the cache on the sensor node. Hence, if the previous sensor reading differs from the newly required reading by an amount that is less than the query-specified threshold, the query engine will not activate the sensor again. This minimizes the sensor activities. For further improving the network lifetime, instead of sending several small packet messages, corona does in-network clustering of sensor reading results. Furthermore, this clustering operator is resource-aware. Thus, it can dynamically adapt its processing according to the available resources.

Another technique for improving the network lifetime and providing QoS is provided in [93]. In this work, an algorithm that optimizes the execution of continuous queries is provided. Their basic idea, using SQL-like query language, wants to profit from a predefined power and delay weight specified in the query in order to provide an optimal query plan. For that, by taking into account both power and time cost simultaneously, a query plan with minimum cost of computing and data transferring is obtained. Furthermore, to additionally increase the network lifetime, they use reduction techniques like packet merging or data compression to decrease the size of the transferred data into the network.

C. Cross-layer Optimization

The traditional layer approach leads to independent design of different layers and results in strict boundaries between layers. Cross-layer optimization exploits interactions between different layers and can significantly improve energy efficiency as well as adaptability to service, traffic, and environment dynamics. For example, having knowledge of the current physical state will help a channel allocation scheme at the MAC layer in optimizing tradeoffs and achieving throughput maximization.

In WSNs, the sensor network lifetime depends intrinsically on the available energy in the nodes composing the network. This available energy is consumed by the sensing activity, the communication (sending and receiving packets) activity, which is essential to form a WSN, and the data processing [7]. However, the communication activity is more costly in energy than the sensing and processing activities. Hence, current cross-layer optimization techniques use a variety of methods to schedule tasks in an energy-efficient way.

Among these techniques, one can notice the synchronization mechanism used in TinyDB [20] for data transmission between nodes forming the network. In fact, queries in TinyDB are flooded throughout the network. An interval-based communication scheduling protocol is used to collect the query answers via a semantic routing tree, with the root node being the endpoint of the query. Every other node maintains a parent node one step closer to the root from where it is, along with other routing information. The synchronization of the data transmission between nodes is performed by making a parent node in wait for a certain interval of time before reporting its own reading. Specifically, in TinyDB every epoch is subdivided into shorter fixed intervals, with the number of intervals equal to the maximum depth of the routing tree (see figure 3). During its own interval, a parent node will be active and collecting results from its child nodes. In the next interval, the children nodes will be idle, while the parent is still active transmitting the partial aggregate result. The parent node will become idle when it finished receiving and transmitting the partial aggregates in its sub-tree.

Besides performing in-network processing, to increase the network lifetime and the accuracy of data and queries, the *ADAGA* [57], [58] system adapts the collecting and sending activities of the devices according respectively to the remaining memory and energy of sensors. The main idea is to dynamically adapt values for *Sense interval* clause that specifies the interval between consecutive data collections and values for *Send interval* clause that specifies the interval between consecutive sending of packets. The values of these two parameters are updated according respectively to the available memory and power in the sensor nodes. This makes the sensor nodes self-configurable and leads to reduce the power consumption and improve the memory availability.

In wireless sensor networks communicating, the huge amount of sensed raw data between the nodes within the network leads to a lot of problems (including energy wasting, useless data transferring, etc.) because of the limited sensor resources. To deal with these problems, in [94] the authors proposed a distributed and self-organizing scheduling algorithm (*DOSA*) that allows an in-network data aggregation, which is based on spatial and temporal correlations between sensor readings of neighboring nodes. This will permit to avoid the transmission of redundant data, thus improving the network lifetime. The first function of the *DOSA* algorithm is to decide when a particular node should perform this correlating function. Moreover, According to the eventual changes in the network topology, *DOSA* uses cross-layer information from the underlying MAC layer to detect these changes and autonomously reassigns schedules of nodes.

In [92], the authors present a resource-awareness framework for in-network data processing in wireless sensor networks. This proposal is an enhancement over the distributed query-processing engine, *SSDQP* [90]. This approach is a two-phase approach. Hence, in addition to in-network data processing features from the *SSDQP* system, this proposal adapts to changing resource levels such as battery power or available memory. The main idea is to use a publish/subscribe pattern to distinguish the monitoring of the resource from adaptive algorithms that subscribe to receive resource availability updates. Hence, the processing techniques can subscribe to act according to the remaining resources (battery, memory, and CPU utilization). The published phase is performed whenever the total change in resource level is greater than a given threshold. The second phase consists of integrating the resource-awareness algorithm into the *SSDQP* system. Moreover, to reduce the communication cost, the proposal benefits from the advantage of the multi-tasking of the *SSDQP* to de-couple the in-network data processing and communication.

Like TinyDB [20] and Cougar [21], [49], in [95] the authors propose a complete distributed query processing framework for wireless sensor networks. This framework includes principally two components: a compiler/optimizer, named *SNEE* (Streaming NEtwork Engine) [95], [96], [97], and a continuous declarative query language over sensed data streams, named *SNEEql* [98], [99]. This latter is an expressive and SQL-like query language, convenient to query data stream from wireless sensor networks. After receiving the *SNEEql*queries, the compiler/optimizer engine (*SNEE*) optimizes them by taking into account metadata such as information about the network topology, the required energy and time, the cost of nodes sampling among other relevant parameters. After the optimization phase, the compiler/optimizer engine creates, compiles and deploys an executable code, the energy-efficient query evaluation plan, which will run into the participating nodes. This work fully describes also the *SNEE* query compilation/optimization architecture (See Fig. 8) [95] and the difference steps that it takes to optimize the *SNEEql* queries. Moreover, for more flexibility, the query can be coupled with user QoS requirements, such as the energy consumption of nodes, the tolerated response time, etc.

The work in [100] is an enhancement over *SNEE*. The basic idea is based on declarative query approach to specify, optimize, and deploy automatic data analysis techniques in a sensor network. Thus, at first, the *SNEEql* query language is refactored with in-network data analysis capabilities, so that it can be well optimized by the query optimization engine *SNEE*. Second, the new in-network declarative query is deployed by

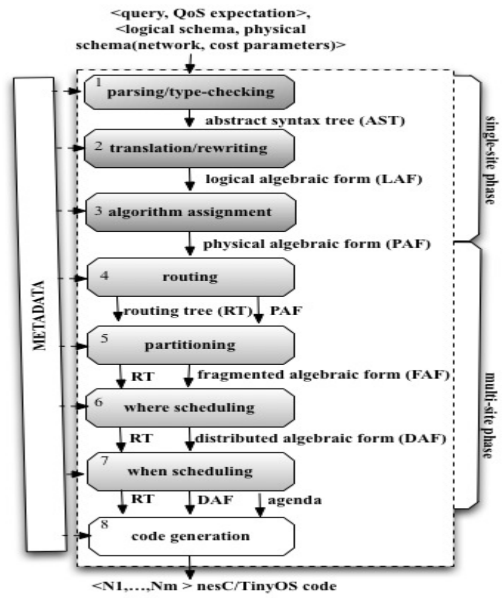


Fig. 8. SNEEql Compiler/Optimizer architecture.

using the basic functionalities of SNEE, which are the automatic code generation, the operator placement, and the deployment of an efficient query execution plan. Therefore, this platform can help, in mobile environments, to adjust changes in the network topology efficiently.

D. Data-centric data/query Dissemination

In data-centric addressing scheme, the collected data is stored by attributes or types (e.g., geographic location and event type) at nodes within the network. Queries for data with a particular attribute will be sent directly to the relevant nodes instead of performing flooding throughout the network; therefore, data-centric approach enables efficient data or query dissemination.

Directed diffusion [101] is one of the pioneering data-centric data dissemination protocols designed specifically for WSNs. The data generated by the sensors are called *events* and queries *interests*. Interests, events and responses of interest are represented by lists of *"attribute = value"*. The interests are injected into the network through one sink node arbitrarily chosen. This sink node broadcasts the interest to its neighbors, which in turn broadcast it to their neighbors, and so on. A node which detects an event compares it with the interests it received, and if there is a match, it sends the description of the event (response to the query) towards the nodes from which it received the interest. Therefore, the event is broadcasted up to the sink node.

In WSNs, the huge amount of information transferring wastes resources. The processing of instructions is much less expensive than the wireless data transmission. Thus, to minimize the communication overhead, in [102] the authors keep, as much as possible, the sensed data in the network and do transmission on demand. To reach their objectives, an API, called *miniDB*, is used to store and extract required data in sensor nodes. In fact, names are associated as identifiers to the data items in order to simplify access. This meta information increases the necessary memory and helps to locate and to retrieve data from arbitrary nodes, even from dynamically placed new nodes, in a simplified manner. Moreover, a tiny query management system, named *miniSQL*, is designed for data identification and interrogation anywhere in the network. An SQL-like query is used to interact with the network. This solution allows to access historic data stored in the sensor network but in order to prevent from the memory saturation it should include an efficient policy of memory saturation management.

Generally, according to natural conditions of environment, the sensor nodes are subjected to failures. Thus, in addition to take resource constraints of WSNs into account, proposed solutions should be aware of risk of sensor failures. For that, the *DISC* (Distributed Information Storage and Collection for WSNs) is proposed in [103]. *DISC* is a protocol for distributed information storage and collection. In *DISC* protocol the network is logically divided into clusters, where each of them is managed by a cluster head. There are two specifics cluster heads; *the primary cluster head (PCH*) and *the backup cluster heads* (*BCH*). The sensor data are periodically reported to the *PCH*, which in turn after eventual aggregation transmits the data to the *BCHs* chosen in a probabilistic way. The *DISC* protocol, compared with the other protocols in this area that make deterministic choice of the backup nodes, performs a random selection strategy. This can prevent from destruction of the exact nodes keeping important information. Moreover, the *DISC* protocol uses the *Bloom filters* method for the data description and the routing mechanism. The data descriptor is unique and it includes the time epoch of the data aggregation, the region identifier of the aggregator and the type of data. Although this protocol is interesting for the robustness of the network, it should be revised in order to take the severe resource constraints of WSNs into account. In fact, the complexity of the algorithms with various backup nodes can be negative factors for the network lifetime and the low memory resource.

Similarly to previous works that investigate the improvement of the network lifetime, the *DDCRS* [104] (dynamic data-centric routing and storage) mechanism minimizes the communication overhead by dynamically determining the locations of the data-centric nodes according to the multiple sink nodes' location and data collecting rate. Moreover, this proposed scheme automatically constructs shared routes from data-centric nodes to multiple sinks, reducing thus duplicate packets transmission. The *DDCRS* scheme consists of two phases: a static phase which consists of defining a data-centric node, called home data-centric responsible for storing sensed data and future replying to sink nodes, by a hash table. In fact, sensor nodes transmit sensed data to neighboring data-centric nodes through *GPRS* routing algorithms. The dynamic phase applies to handle data storage and delivery when the locations of the data-centric node change according to the locations and the reply frequencies of multiple sink nodes.

In [105] a distributed spatial-temporal Similarity Data Storage (*SDS*) scheme is provided. For that, the monitoring area is partitioned into zones based on the geographical layout. Nodes are deployed according to zones and each clustered nodes of a particular zone has a head, which acts as a server for all the remaining nodes. According to a hash function the sensed data are mapped and stored in certain nodes where similarity of data in the zones is preserved. Thus, a data search in a particular zone could find similar data in the neighboring zones. In addition to a search based on data similarity, *SDS* also allows spatial-temporal data searching as data stored in the neighborhood is also based on time and location. This further optimizes the data querying. Moreover, *SDS* uses geographical information of nodes for routing, instead of based on *GPS*. It also provides a carpooling routing algorithm, which can combine the messages belonging to the same destination in routing for minimizing the communication overhead.

Data-centric storage provides energy-efficient data dissemination and organization in wireless sensor networks. Moreover, using metadata is useful for efficient query routing and data access. To support all these requirements, in [106], [107] the authors propose a framework to accelerate query evaluation in content caching networks using XML metadata. This framework includes a concept of content caching networks in which, the collected data are stored by their contents in a distributed way and the data are cached in the network for a certain period of time before they are sent to a database server. Moreover, this framework includes a metadata-guided query evaluation mechanism to improve the query execution. Thus, each cache node maintains a metadata of its data content. Therefore, queries will be based on these metadata to efficiently access the relevant cached data. For the representation of the metadata, they use an XML representation. Additionally to deal with the memory overhead, this work proposes two data clustering and compression algorithms for metadata construction: a clustering, balancing, and compression (*CBC*) algorithm for numerical data, and a clustering, expanding, and compression (*CEC*) algorithm for categorical data.

IV. DISCUSSION AND OPEN ISSUES

The distributed database approach for sensor networks is adopted when sensor nodes do not need to send periodically the collected data to the base station. The sensed data remains on the sensor nodes and some queries are distributed and processed in the sensor nodes. Thus, in this case, the whole sensor network is viewed as a distributed database. This approach is commonly called *in-network processing*.

The in-network processing is a generic term, which could simply refer to any sort of processing that takes place inside a node. Therefore, the data reduction techniques (data aggregation, packet merging, data compression techniques, data fusion, and approximation based techniques) can be placed in the class of in-network processing techniques because each of these techniques is done within nodes composing the network.

Moreover, one can notice that the three approaches (acquisitional query processing, cross-layer optimization and data-centric data/query dissemination) can be all classed inside the category of in-network processing, since all these three sub-categories require a sensor node to process data and make decision instead of transmitting all the data to a central server for off-line processing, as it is done in in-network processing [39].

The tables considered above (Table I, Table II, and Table III) provide a summary and offer a comparison between the above-described distributed database management solutions for WSNs. Moreover, these tables highlight the classification of each solution into specific categories.

After the detailed analysis of the most recent distributed data management techniques for sensor networks, the following open issues can be identified and suggested:

- Keeping and processing sensed data among the sensor nodes are very useful, according to the fact that in-network processing is more energy-efficient than transmitting sensed data for off-line processing. However, this approach may be revised and improved to take into account of the memory overload of the sensor nodes according to the huge amount of generated data compared with the hard resource constraints. Moreover, for efficient query execution in terms of saving energy and good latency, an efficient load balancing policy according to the remaining power and the load of the nodes can be adopted.
- This above study shows that most of the sensor data representations are based on the relational model and the queries are specified using an SQL-like query language. According to the frequent changing characteristics of collected data, the sensor data schema should well fit into

Transactions on Parallel and Distributed Systems

Table I– Summary to various kinds of proposals and their features (part 1)

	Characteristics				
Name of Project/ Authors	**Proposal approach**	**Basic conceptual characteristics**	**Queries expression**	**Optimization / Metrics**	**Platform**
Cougar, 2002, 2003	Distributed in-network query processor	• In-networking processing: aggregation based techniques, packet merging; • Clustered approach	Declarative SQL-like query	•Power-aware optimization: Query optimization plan; catalog management; aggregation in cluster heads; Minimizing Communication overhead	Simulation PDA class
TinyDB, 2002, 2005	Distributed Acquisitional Query Processing System	•In-network processing: aggregation based techniques; •Acquisitional query processing: sensor sampling, query optimization based on metadata, multiquery optimization; •Cross-layer optimization: data stream communication scheduling, self-organization using time interval; •Data-centric query/data dissemination: semantic routing tree	Declarative SQL-like query	•Power-aware optimization: Query optimization plan: Metadata management, ordering of sampling, aggregation; Minimizing the Communication overhead	Simulation
TINA, 2003, 2004	Temporal Coherency-Aware In-network Aggregation	•In-network processing: temporal freshness-aware readings; transmit readings if only threshold defined is exceeded: trade-off between the quality of transmitted data and energy consumption	Declarative SQL-like query	•Power-aware optimization: coherence-aware in-network aggregation, optimize the communication cost	Simulation
Yu et al., 2004	Energy-Latency Tradeoffs for data gathering in WSNs	• In-networking processing: aggregation based techniques • Minimize energy dissipation: energy-latency tradeoffs	---	•Power-aware optimization: energy-latency tradeoffs	Simulation
ADAGA, 2007, 2008	Resource-aware in-network aggregation operator	•In-network aggregation, packets replication •Approximation based techniques •Cross-layer optimization: collecting and sending data according respectively to the remaining memory and energy of sensors	SQL-like query: SNQL	•Power-aware optimization: In-Network Aggregation, resource-aware processing; •Optimize data access: packets replication	Simulation
Amato et al., 2007	Context-aware architecture for distributed query processing	•Framework in the context-aware architecture •In-network query processing	SQL-like query	•Power-aware optimization: In-Network Aggregation	---
Hung and Peng, 2011	Mechanism of optimizing in-network aggregate queries	•In-network processing: in-network aggregation, in-network sharing of intermediate results •Acquisitional query processing: sampling by access intermediate results from others sensors.	---	•Power-aware optimization: Minimizing communication overhead; •Multi-query optimization	Simulation
Wagner, 2004	Resilient aggregation techniques for cluster based WSN	• Aggregation based techniques • Security: framework to evaluate the security of data aggregation schemes	---	• Maximize the resilient of data aggregation techniques	Mathematica l theory
Sia, 2003	Secure aggregation technique against stealthy attack	• Aggregation based techniques • Security: framework for secure data aggregation against stealthy attack; user accepts aggregated result according to a given bound	---	• Ensure the reliability of data aggregation results	---
ESPDA, 2003, 2005	Energy-efficient and secure pattern-based data aggregation for WSNs	• Aggregation based techniques • Cluster-based approach • Security: encrypted data not be decrypted by cluster heads when performing data aggregation	---	•Power-aware optimization: prevent from the transmission of redundant data •Sensor data transmitted to base station in encrypted form	Simulation
BBQ, 2004	Framework to acquire data using statistical modeling techniques	•Approximation based techniques: approximate sensor readings using statistical models; temporal and correlation correlations	SQL-like query with error tolerance and target confidence	•Power-aware optimization: Minimizing the communication cost and the sensor activities, sampling	Simulation

Transactions on Parallel and Distributed Systems

Table II– Summary to various kinds of proposals and their features (part 2).

Name of Project/ Authors	Characteristics				
	Proposal approach	**Basic conceptual characteristics**	**Queries expression**	**Optimization / Metrics**	**Platform**
Yulong et al., 2011	Distributed replication schemes	•In-network data storage and replication(fault-tolerance); •Probabilistic based techniques	---	•Power-aware optimization: minimize communication overhead; •Maximize fault-tolerance	---
Zeinalipour-Yazti et al., 2008	Distributed top-k query processing	•In-network processing: approximation based technique	Top-k query	•Power-aware optimization: Data Retrieving cost minimizing	---
Ye et al., 2010, 2011	Framework for distributed processing of probabilistic top-k queries	•In-network processing: approximation based technique, aggregation based techniques, algorithms adapted according to the communication cost	Probabilistic Top-k query	•Power-aware optimization: Data Retrieving cost minimizing, minimizing of the communication cost.	Experiment
ENERGY*, 2006	Optimal placement problem of proxy node	•In-network data processing: approximation based technique	---	•Power-aware optimization: minimize the energy consumption on data transmission	Analytic Simulation
Panda *et al.*, 2010	Distributed data processing algorithms that prevent from link failure and noise	•In-network processing •Approximation based techniques	---	•Power-aware optimization: Optimize the communication cost; Cost function optimization	Simulation
ShuKui et al., 2009	Approximation Algorithm for Data Aggregation	•In-network processing: aggregation based techniques; •Approximation based techniques	SQL-like query	•Power-aware optimization: Optimize the communication cost	Simulation
SenseSwarm, 2007, 2009, 2011	Framework for in-network data acquisition and replication in mobile WSN	•Hierarchical voting-based fault-tolerance architecture; •In-network data acquisition and replication(fault-tolerance); •spatio-temporal in-network aggregation; •Approximation based techniques	---	•Power-aware optimization: in-network aggregation; •Maximize fault-tolerance	Simulation
Umer et al., 2009	Selectivity-awareness query optimization	•In-network data processing: in-network aggregation •Acquisitional techniques: queries sample spatially correlated nodes •Data-centric data/query dissemination •Context aware system	SQL-like query	•Power-aware optimization: Optimize energy of the data collection process	Simulation
Kulakov and Davcev, 2005	Distributed data processing based on artificial neural-networks algorithms	•In-network data processing; •Artificial neural-networks algorithms adaptation for WSN	---	•Power-aware optimization: Optimize the communication cost	Simulation
SSDQP, 2007	Time Triggered Query Processing system	•In-network processing: synchronized merge operation, data compression method, aggregation based techniques; •Acquisitional query processing: time-triggered query engine, query optimization based on user's need •Cross-layer optimization: task scheduler, sensing and process before transmitting are synchronized	SQL queries	•Power-aware optimization: Minimizing Communication overhead	Simulation
Corona, 2010	Distributed multi-query processor	•In-network aggregation based techniques, sensor reading clustering; •Acquisitional query processing: freshness-aware data acquisition •Cross-layer optimization: task scheduler	Acquisitional SQL queries	•Power-aware optimization: Optimize the communication cost and the sensor activities	Simulation
Sun and Zhou, 2008	Power-aware query execution plan	•Distributed multi-query engine •In-network processing: packet merging or compression based techniques; •Acquisitional query processing: power-aware data sampling;	SQL-like query	•Power-aware optimization: Optimize the communication and execution cost	Analytic experiment

Transactions on Parallel and Distributed Systems

Table III– Summary to various kinds of proposals and their features (part 3).

	Characteristics				
Name of Project/ Authors	**Proposal approach**	**Basic conceptual characteristics**	**Queries expression**	**Optimization / Metrics**	**Platform**
DOSA, 2008	A distributed and self-organizing scheduling algorithm for in-network data aggregation aware of correlated data	•In-network data processing: in-network aggregation, approximation based technique •Cross-layer optimization: self-organization scheduling based on the MAC layer	---	•Power-aware optimization: avoid sending redundant data by in-network aggregation aware of correlated data	Theoretical Simulation results
Rohm, 2008	Resource-awareness framework for in-network data processing	•Resource-awareness framework; •In-network data processing: on-line data clustering, merge operation; •Acquisitional query processing: time-triggered query engine, query optimization based on user's need •Cross-layer optimization: task scheduler, sensing and communication are synchronized	SQL queries	•Power-aware optimization: Optimize the communication cost	Simulation
SNEE, 2008, 2009, 2011	Distributed query processing framework for WSNs	•In-network aggregation based techniques; •Acquisitional query processing: sensor sampling, query optimization based on metadata; •Cross-layer optimization: tasks scheduling based on available resources, communication scheduling for data flow •Data-centric query/data dissemination: choice of routing tree nodes based on semantic constraints	SNEEql queries	•Power-aware optimization: Query optimization plan: Metadata management, sampling, aggregation; energy-efficient communication strategy, tasks scheduling based on remaining energy on nodes	Empirical evaluation, Simulation
Valkanas et al., 2011	In-network query processing based on SNEE	•In-network processing: adjust of network topology changes; Data analysis techniques	Data analysis queries based on SNEEql	•Optimization based on SNEE	---
Directed Diffusion, 2002	Reactive data-centric protocol	•In-network processing: filters, suppression of duplicate messages Data analysis techniques •Data-centric query/data dissemination: interest dissemination based on named data, gradient setup used to route data back to the sink node	---	•Optimization based on filter, suppression of duplicate messages	---
Awad et al., 2008	Distributed data management system	•In-network data processing: data transmission on demand; •Utilization of an API: miniDB/miniSQL; •Data-centric query/data dissemination	SQL-like query	•Power-aware optimization: Minimizing the communication overhead	Simulation
DISC, 2007	Distributed data storage and collecting protocol	•backup cluster heads, in-network aggregation; •Randomly selection strategy of backup nodes: lead network robustness; •Data descriptor: Bloom filters technique; •Data-centric data/query dissemination	SQL-like query	---	Formal analysis Simulation
DDCRS, 2010	Frequency-aware data-centric routing and storage	•In-network processing •Data-centric Storage	---	•Power-aware optimization: Minimizing Communication overhead	Simulation
SDS, 2011	Distributed spatial-temporal similarity data storage scheme	•Distributed data-centric storage: semantic routing, spatial-temporal and similarity data searching	---	•Power-aware optimization: Minimizing communication overhead;	Simulation
Liu et al., 2009, 2011	Framework to accelerate query evaluation in content caching networks using XML metadata.	•In-network aggregation; •Acquisitional query processing: sampling nodes by metadata guided; •Distributed data-centric storage	XPath queries	•Power-aware optimization: in-network aggregation; •Improving query execution	Simulation

XML representation. Then, XPATH can be used for flexible and easy queries.

- The distributed technique treats the information within sensor nodes. According to the unstable and generally harsh environment, there may be sudden failures of sensors. This can lead to information loss that greatly influences the result analysis or even the system blocking. To deal with this situation, one can opt for a hybrid approach where, in addition to in-network data processing, some data can be kept in the base station database and updating from time to time. After a certain time, these data may gradually lose their freshness. Thus, one can use a generalized database management system to handle various types of applications and user needs.
- Efficient meta-data management is very useful when one can manage a huge amount of distributed data. Although TinyDB and the work in [22], [107] include meta-data management for query optimization, it is not enough. Thus, proposals may be improved by including energy-efficient meta-data management in order to have fast and accurate routing.
- In multi-users environment, simultaneous multi-query processing is very useful. Systems connected to a database are often subject to numerous queries especially for consultation. Hence, a good policy of multi-query optimization would be a good deal.
- Another important research issue is the database security. Although research efforts have been made on security issues in wireless sensor network in general and particularly in security on existing data gathering protocols, some challenges can still be addressed. First, WSNs are application-specific and the choice of the appropriate cryptographic methods depends on the processing capability of sensor nodes, then there is no unified solution for all sensor networks. Therefore, the design of a security mechanism adaptable to various WSN applications could be interesting. Second, WSN are resource constraints, so the design of security services in WSNs must be aware of these constraints. Third, most of security protocols are designed for fixed topologies. Whereas with applications based on mobile sensors, the mobility of the devices influences the sensor network topology, therefore leads to many problems in secure routing protocols.

V. CONCLUSION

Given the great limited sensor device resources, storing and exploiting the large amount of data generated by sensors is a very big problem. The data management techniques used in traditional databases to manage a huge amount of data are not generally suitable for sensor networks taking their specificities into account. Then, the research community has provided a new data storage and querying method, named the distributed database approach for sensor networks. This latter is viewed as the most energy-efficient method to manage the large amount of data generated by the sensor nodes.

Many different techniques have been proposed to manage data and queries in a distributed architecture, but all these techniques can be categorized by their way of processing data into the network (in-network aggregation processing, in-network approximation processing, acquisitional query processing, cross-layer optimization, data-centric data/query dissemination). Thus, this work presented and discussed from the oldest to the most recent proposed techniques that have been performed and it is particularly interesting in various stages of distributed data storage, distributed query processing and optimization for sensor networks. The processing techniques aim, generally, to optimize the energy consumption in the network and to retrieve more accurate information. Moreover, the queries used in these techniques are generally SQL-like.

Systems based on sensor networks are increasingly common in many areas of the knowledge, giving rise to several flavors of WSN applications. These applications are generally specific. For instance, there are critical applications that have temporal constraint and need more accurate information in order to take efficient decisions. New query processing optimization technique, while ensuring data availability in case of failure is very important in this context. So, one processing technique may not be efficient for the different applications. To deal with these challenges one can opt for a hybrid approach. Furthermore, regarding the Tables I, II, and III, few proposals have been based on metadata management, which is very useful when one manages a huge amount of distributed data. The design of a framework that takes into account the various particular characteristics of WSN applications can well meet the requirements to a generalized database management system to handle various types of applications and user needs.

ACKNOWLEDGMENTS

This work was partially supported by the *Instituto de Telcomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal, by the *Ministerio de Ciencia e Innovación*, through the *Plan Nacional de I+D+i 2008–2011* in the *Subprograma de Proyectos de Investigación Fundamental*, project TEC2011-27516, by the Polytechnic University of Valencia, though the PAID-05-12 multidisciplinary projects, and by National Funding from the FCT – *Fundação para a Ciência e a Tecnologia* through the Pest-OE/EEI/LA0008/2013 Project.

REFERENCES

[1] L. Shu, M. Hauswirth, L. Cheng, J. Ma, V. Reynolds, and L. Zhang, "Sharing Worldwide Sensor Network," International Symposium on Applications and the Internet, IEEE, pp. 189-192, 2008, DOI 10.1109/SAINT.2008.22.

[2] K. Aberer, M. Hauswirth, and A. Salehi, " Global Sensor Networks," TechnicalreportLSIR-REPORT-2006-001, http://lsir.epfl.ch/research/current/gsn/, 2006.

[3] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for Data Processing in Large-Scale Interconnected Sensor Networks," in Proceedings of the 8th International Conference on Mobile Data Management, Mannheim, Germany, May 7-11, 2007.

[4] E. H. Callaway, "Wireless Sensor Networks: Architectures and Protocols," CRC Press, 2004.

[5] P. Baronti, P. Pillai, C: V. Chook, S. Chessa, A. Gotta, and F. Y. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards," Journal of Computer communications, 30 (7), 1655–1695, 2007.

[6] L. M. L. Oliveira and J. J. P. C. Rodrigues, "Wireless Sensor Networks: a Survey on Environmental Monitoring," Journal of Communications (JCM), Vol 6, No 2, pp. 143-151, April 2011, DOI: 10.4304/jcm.6.2.143-151

[7] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," Computer networks, 38, 393-422, 2002.

[8] A. Nayak, I. Stojmenovic, "Wireless sensor and actuator networks: algorithms and protocols for scalable coordination and data communication,"Wiley-Interscience,2010.

[9] K. Lin, M. Chen, S. Zeadally, and J. J. P. C. Rodrigues, "Balancing Energy Consumption with Mobile Agents in Wireless Sensor Networks," Future Generation Computer Systems, Vol. 28, Issue 2, pp. 446-456, Feb. 2012, DOI: 10.1016/j.future.2011.03.001,

[10] K. Lin, J. J. P. C. Rodrigues, H. Ge, N. Xiong, and X. Liang, "Energy Efficiency QoS Assurance Routing in Wireless Multimedia Sensor Networks," IEEE Systems Journal, Vol. 5, Issue 4, pp. 495-505, December 2011, DOI: 10.1109/JSYST.2011.2165599

[11] S. Sendra, Jaime Lloret, Miguel Garcia and Jose F. Toledo, "Power saving and energy optimization techniques for Wireless Sensor Networks," Journal of Communications, Vol. 6 Issue 6, Pp. 439-459. August 2011. doi:10.4304/jcm.6.6.439-459

[12] S. Matsumae, "Energy-Efficient Cell Partition of 3D Space for Sensor Networks with Location Information," Network Protocols and Algorithms, Vol 1, No 2 (2009), Pp. 85-98

[13] P. A. Neves, J. J. P. C. Rodrigues, M. Chen, and A. V. Vasilakos, "A Multi- channel Architecture for IPv6-enabled Wireless Sensor and Actuator Networks Featuring PnP Support," Journal of Network and Computer Applications, 2011, (in press), DOI: 10.1016/j.jnca.2011.03.033

[14] L. M. L. Oliveira, J. J. P. C. Rodrigues, A. F. de Sousa, and J. Lloret, "Network Access Control Framework for 6LoWPAN Networks," Sensors, Vol. 13, Issue 1, pp. 1210-1230, 2013. DOI: 10.3390/s130101210.

[15] J. M. L. P. Caldeira, J. J. P. C. Rodrigues, and P. Lorenz, "Towards Ubiquitous Mobility Solutions for Body Sensor Networks on HealthCare," IEEE Communications Magazine, Vol. 50, No. 5, pp. 108-115, May 2012.

[16] A. Campos, E. Souza, F. Nakamura, J. J. P. C. Rodrigues, and E. Nakamura, "On the Impact of Localization and Density Control Algorithms in Target Tracking Applications for Wireless Sensor Networks,",Sensors, Vol. 12, No. 6, pp. 6930-6952, May 2012, DOI: 10.3390/s120606930

[17] L. Sacks, M. Britton, I. Wokoma, A. Marbini, T. Adebutu, I. Marshall, C. Roadknight, J. Tateson, D. Robinson, A. G. Velazquez, "The development of a robust, autonomous sensor network platform for environmental monitoring," Sensors and their Applications XXII, Limerick, Ireland, 2003.

[18] L. Shu, Y. Zhang, G. Min, Y. Wang, M. Hauswirth, "Cross-Layer Optimization on Data Gathering in Wireless Multimedia Sensor Networks within Expected Network Lifetime", accepted in Springer Journal of Universal Computer Science (JUCS), 2009.

[19] C. Zhu, L. Shu, T. Hara, L. Wang, S. Nishio, L.T. Yang, "A survey on communication and data management issues in mobile sensor networks," Wireless Communications and Mobile Computing, 10 Nov. 2011. DOI: 10.1002/wcm.1219

[20] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TinyDB: An Aquisitional Query Processing System for Sensor Networks," ACM Transactions on Database Systems (TODS), 30 (1), 122-173, 2005.

[21] Y. Yao and J. Gehrke, "Query Processing in Sensor Networks," In First Biennial Conference on Innovative Data Systems Research (CIDR 2003),Asilomar, California, January 5-8, 2003.

[22] M. Umer, L. Kulik, and E. Tanin, "Optimizing query processing using selectivity-awareness in Wireless Sensor Networks," Journal of Computers, Environment and Urban Systems, 33 (2), 79-89, 2009.

[23] L. Cheng, Y. Chen, C. Chen, J. Ma, L. Shu, A.V. Vasilakos, N. Xiong, "Efficient Query-Based Data Collection for Mobile Wireless Monitoring Applications," The Computer Journal, 53 (10), 1643-1657. 2010.

[24] J. G. Pottie and J. W. Kaiser, "Wireless integrated network sensors," Communication of the ACM, 43(5), pp. 51-58, 2000.

[25] R. Kacimi,"Techniques de conservation d'énergie pour les réseaux de capteurs sans fil," Ph.D. dissertation, Dept. Math., Info. and Telecom., INPT Univ., Toulouse, France, September 2009.

[26] C. Dini,"Les RéseauxCapteurs Sans Fil Avec Access Sporadique Au Noeud-puits," Ph.D. dissertation, Info. Eng., Haute Alsace Univ., France, December 2010.

[27] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: a tiny aggregation service for ad-hoc sensor networks,"ACM Symposium on Operating System Design and Implementation, OSDI 2002, Boston, MA, USA, pp.131-146, 2002.

[28] E. Fasolo, M. Rossi, J. Widmer, and Michele Zorzi, "In-network aggregation techniques for wireless sensor networks: a survey," IEEE Wireless Communications, 14(2), pp. 70-87, April 2007.

[29] C. Tang and C. S. Raghavendra, "Compression techniques for wireless sensor networks," Wireless Sensor Networks, pages 207-231, 2004.

[30] C. Dini and P. Lorenz, "Primitive Operations for Prioritized Data Reduction in Wireless Sensor Network Nodes," 4th International Conference on Systems and Networks

Communications, ICSNC 2009, Porto, Portugal, pp. 274-280, 2009.

[31] J. Xu, X. Tang, and W. C. Lee, "A New Storage Scheme for Approximate Location Queries in Object Tracking Sensor Networks," IEEE Transactions on Parallel and Distributed Systems (TPDS), 19(2): 262-275, Feb. 2008.

[32] C. J. Debono, and N. P. Borg, "The Implementation of an Adaptive Data Reduction Technique for Wireless Sensor Networks," IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Sarajevo, 16-19 Dec. 2008.

[33] L. Shu, J. Lloret, J. J. P. C. Rodrigues, M. Chen, "Distributed Intelligence and Data Fusion for Sensor Systems," IET Communications, Vol. 5, Issue 12. Pp. 1633-1636. October 2011.

[34] P. Bonnet, J. Gehrke, and P. Seshadri, "Towards sensor database systems," 2nd International Conference on Mobile Data Management, Hong Kong, China, January 8-10, 2001.

[35] O. Diallo, J. J. Rodrigues, and M. Sene, "Real-time data management on wireless sensor networks: A survey," Journal of Network and Computer Applications, 35 (3), pp. 1013-1021, May 2012.

[36] C. Li, H. Zhang, B. Hao, and J. Li, "A Survey on Routing Protocols for Large-Scale Wireless Sensor Networks," Sensors, 11(4), 3498-3526, 2011.

[37] J. N. Al-karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," IEEE Wireless Communications , 11 (6), 6-28, 2004.

[38] M. Demirbas, X. Lu, and P. Singla, "An in-network querying framework for wireless sensor networks," IEEE Trans. on Parallel and Distributed Systems (TPDS), vol. 20, no. 8, pp., 1202-1215, 2009.

[39] S. Chatterjea and P. Havinga, "A Taxonomy of Distributed Query Management Techniques for Wireless Sensor Networks," International Journal of Communication Systems, 20 (7), 889-908, 2007.

[40] A. Girighar and P. R. Kumar, "Toward a theory of in-network computation in wireless sensor networks," IEEE Communications Magazine, 44 (4), 98-107, 2006.

[41] P. Andreou, D. Zeinalipour-Yazti, P. K. Chrysanthis, and G. Samaras, "In-network data acquisition and replication in mobile sensor networks," Distributed and Parallel Databases, 29, 87–112, 2011.

[42] B. Radunovic, "A Cross-Layer Design of Wireless Ad-Hoc Networks," Thesis, 2005.

[43] L. D. Mendes and J. J. Rodrigues, "A survey on cross-layer solutions for wireless sensor networks," Journal of Network and Computer Applications. Volume 34, Issue 2, March 2011, Pp. 523–534.

[44] 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, Vol. 24, Issue 11, pp. 1445-1466, November 2011. DOI: 10.1002/dac.1228.

[45] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking,"IEEE/ACM Transactions Networking, 11 (1), 2-16, 2003.

[46] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin, "Data-Centric Storage in Sensornets," CM SIGCOMM Computer Communication Review, 33 (1), 137-142, 2003.

[47] Y. Diao, D. Ganesan, G. Mathur, and P. Shenoy, "Rethinking data management for storage-centric sensor networks," Third Biennial Conference on Innovative Data Systems Research (CIDR'07),Asilomar, CA, USA, Jan. 2007, pp. 22–31.

[48] M. Bin Ahmad, M. Hasan Islam, M. Asif, and Dr. Sadia Aziz, "A Short Survey on Distributed In-Network Query Processing in Wireless Sensor Networks," Networked Digital Technologies (NDT), IEEE, pp. 541-543, 2009.

[49] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor network," ACM SIGMOD Record, 31 (3), 9-18, 2002.

[50] S. Madden, W. Hong, J. Hellerstein, and K. Stanek, "TinyDB: A Declarative Database for Sensor Networks," accessed 25/01/2013 Available: http://telegraph.cs.berkeley.edu/tinydb/overview.html

[51] Berkeley. TinyOS. Available: http://www.tinyos.net/, 1999.

[52] P. Bonnet, J. Gehrke, and P. Seshadri, "Cougar: The Sensor Network is the Database," accessed 25/01/2013 Available: www.cs.cornelle.edu/database/cougar/, 2001.

[53] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis, "TiNA: A Scheme for Temporal Coherence-Aware in-Network Aggregation," 3rd ACM international workshop on Data engineering for wireless and mobile access MobiDe '03, pp. 69–76, San Diego, California, USA, 2003.

[54] A. Sharaf, J. Beaver, A. Labrinidis, and K. Chrysanthis, "Balancing energy efficient and quality of aggregate data in sensor networks," The International Journal on Very Large DataBases (VLDB), 13 (4), 384 – 403, 2004.

[55] J. Beaver, M. A. Sharaf, A. Labrinidis, and P. K. Chrysanthis, "Power-aware in-network query processing for sensor data," Second hellenic data Management Symposium (HDMS), pp. 1-17. Athens, Greece, 2003.

[56] Y. Yu, B. Krishnamachari, and V. Prasanna, "Energy-Latency Tradeoffs for Data Gathering in Wireless Sensor Networks," in IEEE Infocom 2004, Hong Kong, Mar. 2004.

[57] A. Brayner, A. Lopes, D. Meira, R. Vasconcelos, and R. Menezes, "An adaptive in-network aggregation operator for query processing in wireless sensor networks," The journal of Systems and Software , 81 (3), 328–342, 2008.

[58] A. Brayner, A. Lopes, D. Meira, R. Vasconcelos, and R. Menezes, "Toward adaptive query processing in wireless sensor networks," Signal Processing, 87 (12), 2911–2933, 2007.

[59] G. Amato, F. Furfari, S. Lenzi, and S. Chessa, "Enabling Context Awareness through Distributed Query Processing in Wireless Sensor Networks," 2nd International Workshop on Requirements and Solutions for Pervasive Software infrastructures, Innsbruck, Austria, September 16, 2007.

[60] G. Amato, S. Chessa, A. Caruso, and P. Baronti, "MaD-WiSe: Management of Data in Wireless Sensor networks", 2005. Available: http://mad-wise.isti.cnr.it, accessed 25/01/2013

[61] C. Hung, and W. Peng, "Optimizing in-network aggregation queries in wireless sensor networks for energy saving," Data and Knowledge Engineering, 70 (7), 617-641, 2011.

[62] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, "Wireless Sensor Network Security: A Survey," Security in Distributed, Grid, and Pervasive Computing, pp. 1-50, Auerbach Publications, CRC Press, Chapter 17, 2006.

[63] P. Mohanty, S. Panigrahi, N. Sarma, and S. S. Satapathy, "Security Issues In Wireless Sensor Network Data Gathering Protocols: A Survey," Journal of Theoretical and Applied Information Technology, 13(1), pp. 14-27, 2010.

[64] J. Sen, "A Survey on Wireless Sensor Network Security," International Journal of Communication Networks and Information Security (IJCNIS),1(2), August 2009.

[65] S. K. Singh, 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 (IJCTT), pp. 1-9, May to June Issue 2011

[66] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," In Proceedings of the third international symposium on Information processing in sensor networks, pp. 259–268, ACM Press, 2004.

[67] J. Deng, R. Han, and S. Mishra, "Counter measuers against traffic analysis in wireless sensor networks," Technical Report CU-CS-987-04, University of Colorado at Boulder, 2004.

[68] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," In Proceedings of IEEE Symposium on Security and Privacy, May 2005.

[69] D. Wagner, "Resilient aggregation in sensor networks," In Proceedings of the 2nd ACM workshop on Security of Ad hoc and Sensor Networks (SASN '04), pp. 78–87, New York, NY, USA, ACM Press, 2004.
[70] B. Przydatek, D. Song, and A Perrig, "Sia: Secure information aggregation in sensor network," 2003.
[71] H. Cam, D. Muthuavinashiappan, and P. Nair, "ESPDA: Energy-efficient and secure pattern-based data aggregation for wireless sensor networks," In Proceedings of IEEE Sensors, pp. 732-736, Toronto, Canada, October 2003.
[72] H. Cam, D. Muthuavinashiappan, and P. Nair, "Energy-efficient security protocol for wireless sensor networks," In Proceedings of IEEE VTC Conference, pp. 2981- 2984, Orlando, Florida, October 2005.
[73] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, W. Hong, "Model-driven acquisition in sensor networks," Proceedings of VLDB, Toronto, Canada, 2004.
[74] S. Yulong, X. Ning, P. Qingqi, M. Jianfeng, X. Qijian, and W.Zuoshun, "Distributed Storage Schemes for Controlling Data Availability in wireless Sensor Networks," Seventh International Conference on Computational Intelligence and Security (CIS), pp. 545-549, 3-4 Dec. 2011.Sanya, Hainan, China.
[75] D. Zeinalipour-Yazti and Z. Vagena, "Distributed Top-K Query Processing in Wireless Sensor Networks," 9th International Conference on Mobile Data Management (MDM 2008), Beijing, China, April 27-30, 2008.
[76] M. Ye, X. Liu, W. Lee, and D. L. Lee, "Probabilistic Top-k Query Processing in Distributed Sensor Networks," IEEE 26th International Conference of Data Engineering (ICDE), March 1-6, 2010, Long Beach, California, USA. pp. 585-588.
[77] M. Ye, W. Lee, D. L. Lee, and X. Liu, "Distributed Processing of Probabilistic Top-k Queries in Wireless Sensor Networks," IEEE Transactions on Knowledge and Data Engineering, Vol. 25, Issue 1, June 2013. Pp. 76 - 91. DOI: 10.1109/TKDE.2011.145
[78] C. Zhu, L. T. Yang, L. Shut, T. Hara, and S. Nishio, "Implementing Top-k Query in Duty-cycled Wireless Sensor Networks," 7th Int. Wireless Communications & Mobile Computing Conference (IWCMC 2011), pp. 553 – 558, Istanbul, Turkey, July 4-8, 2011.
[79] Y. Li, C. Ai, C. Vu, Y. Pan, and R. Beyah, "Delay Bounded and Energy Efficient Composite Event Monitoring in Heterogeneous Wireless Sensor Networks," IEEE Transactions on Parallel and Distributed Systems (TPDS), 21(9):1373-1385, 2010.
[80] Y. Chen, H. V. Leong, M. Xu, J. Cao, K. C. Chan, and A. T. Chan, "In-network Data Processing for Wireless Sensor Networks," 7th International Conference on Mobile Data Management (MDM'06), Nara, Japan, May 10-12, 2006.
[81] M. Panda, T. Panigrahi, P. M. Khilar, and G. Panda, "Learning with Distributed Data in Wireless Sensor Networks," First International Conference on Parallel, Distributed and Grid Computing, (PDGC-2010),Waknaghat, Solan (H.P), India, October 2010. Pp. 241-245.
[82] Z. Shukui, C. Zhiming, G. Shengrong, and F. Jianxi, "An Efficient Approximation Algorithm for Data Aggregation in Wireless Sensor Networks," Int. Symposium on Information Processing (ISIP'09), pp. 017-020, Huangshan, P. R. China, August 2009.
[83] C. Plattner and G. Alonso, "Ganymed: Scalable Replication for Transactional Web Application," Journal of Middleware, 3231, 155-174, 2004.
[84] O. Gardarin and G. Gardarin, "Le Client-Server," Edition Eyrolles, Paris, 1997.
[85] I. Sarr, H. Naacke, and S. Gançarski, "Failure-Tolerant Transaction Routing at Large Scale," Second International Conference on Advances in Databases, Knowledge, and Data Application (DBKDA), Menuires, France, 11-16 April 2010.
[86] O. Diallo, M. Sene, and I. Sarr, "Freshness-Aware Metadata Management: Performance Evaluation with SWN models," IEEE/ACS Int. Conference of the Computer Systems and Applications (AICCSA 2010),Hammamet, Tunisia, May 16-19, 2010.
[87] P. Andreou, D. Zeinalipour-Yazti, M. Andreou, P. K. Chrysanthis, and G. Samaras, "Perimeter-Based Data Replication in Mobile Sensor Networks," 10th International Conference on Mobile Data Management: Systems, Services and Middleware (MDM), pp. 244-251, Taipei, Taiwan. 18-20 May 2009.
[88] D. Zeinalipour-Yazti, P. Andreou, P. K. Chrysanthis, and G. Samaras, "SenseSwarm: a perimeter-based data acquisition framework for mobile sensor networks," 4th workshop on Data Management for Sensor Networks: In Conjunction with 33rd International Conference on Very Large Data Bases (DMSN), pp. 13-18. Vienna, Austria, April 2008.
[89] A. Kulakov and D. Davcev, "Distributed Data Processing in Wireless Sensor Networks Based on Artificial Neural-Networks Algorithms," 10th IEEE Symposium on Computers and Communications (ISCC 2005), June 27-30, 2005, Cartagena, Spain.
[90] B. Scholz, M. M. Gaber, T. Dawborn, R. Khoury, and E. Tse, "Efficient time triggered query processing in wireless sensor networks," Int. Conference on Embedded Software and Systems (ICESS 2007),Daegu, Korea, May 14-16, 2007. Pp. 391-402.
[91] R. Khoury, T. Dawborn, B. Gafurov, G. Pink, E. Tse, Q. Tse, et al. "Corona: Energy-Efficient Multi-query Processing in Wireless Sensor Networks," Database Systems for Advanced Application (DASFAA), Lecture Notes on Computer Science 5982/2010, Pp. 416-419, 2010.
[92] U. Roehm, M. M. Gaber, Q. Tse, "Enabling resource-awareness for in-network data processing in wireless sensor networks," Nineteenth Australasian Database Conference (ADC 2008), Wollongong, NSW, Australia, Pp. 107-114, January 22-25, 2008.
[93] J. Sun and J. Zhou, "Power-Aware Data Reduction for Continuous Query In Wireless Sensor Networks," IEEE International Conference on Industrial Technology (ICIT 2008), Pp. 1-6, Chengdu, China, April 21-24, 2008.
[94] S. Chatterjea, T. Nieberg, M. Meratnia, and P. Havinga, "A Distributed and Self-Organizing Scheduling Algorithm for Energy-Efficient Data Aggregation in Wireless Sensor Networks," ACM Transactions on Sensor Networks (TOSN), 4 (4), 2008.
[95] I. Galpin, C. Y. Brenninkmeijer, A. J. Gray, F. Jabeen, A. A. Fernandes, N. W. Paton, "SNEE: a query proc essor for wireless sensor networks," Distributed and Parallel Databases, 29, 31–85, 2011.
[96] I. Galpin, C. Y. Brenninkmeijer, A. J. Gray, F. Jabeen, A. A. Fernandes, N. W. Paton, "An architecture for query optimization in sensor networks," IEEE 24th International Conference on Data Engineering (ICDE 2008), Cancun, Mexico, 7-12 April 2008. Pp. 1439 - 1441. DOI: 10.1109/ICDE.2008.4497582
[97] I. Galpin, C. Y. Brenninkmeijer, F. Jabeen, A. A. Fernandes, and N. W. Paton, "Comprehensive optimization of declarative sensor network queries," 21st International Conference on Scientific and Statistical Database Management (SSDBM 2009), Lecture Notes in Computer Science, Vol. 5566. Pp. 339-360.
[98] C. Y. Brenninkmeijer, I. Galpin, A. A. Fernandes, and N. W. Paton, "A semantics for a query language over sensors, streams and relations," 25th British National Conference on Databases, BNCOD 25, Cardiff, UK, July 7-10, 2008. Lecture Notes in Computer Science Vol. 5071, Pp. 87–99.
[99] C. Y. Brenninkmeijer, I. Galpin, A. A. Fernandes, and N. W. Paton, "Validated cost models for sensor network queries," Sixth International Workshop on Data Management for Sensor Networks (DMSN '09), Lyon, FRANCE, August 24, 2009.
[100] G. Valkanas, D. Gunopulos, I. Galpin, A. J. G. Gray, and A. A. A. Fernandes, "Extending query languages for in-network query processing," 10th ACM International Workshop on Data Engineering for Wireless and Mobile

Access (MobiDE '11), Athens, Greece, June 2011. pp. 34-41.
[101] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, "Directed diffusion for wireless sensor networking," ACM/IEEE Transactions on Networking, 11(1), 2–16, 2002.
[102] A. Awad, W. Xie, E. Rose, R. German, and F. Dressler, "Distributed Data Management in Sensor Networks using miniDB / miniSQL," 4th IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS), pp. 27-28. Santorini Island, Greece, June 2008.
[103] C. Jardak, E. Osipov, and P. Mahonen, "Distributed Information Storage and Collection for WSNs," 4th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), Italy-Pista, October 2007.
[104] C. Chang, J. Sheu, S. Chang, and Y. Chen, "A frequency-aware data-centric mechanism for wireless sensor networks," Wireless Communications and Mobile Computing 2010, 10 (8), 1078–1101, 2010.
[105] H. Shen, L. Zhao, and Z. Li, "A Distributed Spatial-Temporal Similarity Data Storage Scheme in Wireless Sensors Networks," IEEE Transactions on Mobile Computing, 10 (7), 982-996, 2011.
[107] H. Wang, R. Liu, X. Zheng, Y. Chen, and H. Liu, "To do or not to do: metadata-guided query evaluation in content caching networks," 28th IEEE Conference on Global Telecommunications, Honolulu, Hawaii, USA, pp. 4524-4529, 30 November - 4 December 2009.
[107] R. Liu, X. Zheng, H. Liu, H. Wang, and Y. Chen, "Metadata-guided evaluation of resource-constrained queries in content caching based wireless networks," Wireless Networks, 17 (8), 1833–1850, 2011.

AUTHORS' BIOGRAPHIES

Ousmane Diallo is a PhD student of Computer Science at the University of Beira Interior under supervised by Professor Joel Rodrigues and co-supervised by Dr. MbayeSene. He received his 5-years BS degree in Computer Science from the Universityof Cheikh Anta Diop of Dakar, Senegal, in 2008; and his MSc degree in Computer Science from the University of Cheikh Anta Diop of Dakar, Senegal, in 2009. He is also teacher in the Department of Informatics at the University of Ziguinchor, Senegal. He is a PhD student member of the Institute of Telecommunications, Portugal. His current research interests are distributed databases for wireless sensor networks, real-time databases, wireless sensor networks and performance evaluation. He is author of one journal paper and one international conference paper. He also participated on several reviewed papers for international conferences.

Joel José P. C. Rodriguesis 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 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 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 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.

Mbaye Sene is a Lecturer in the Department of Mathematics and Informatics of the University of Cheikh Anta Diop of Dakar, Senegal, and researcher at the Faculty of Science and Technology, Senegal. He received a PhD degree in ComputerScience from the University of Paris-Dauphine, a MSc degree and a 5-years BS degree in Computer Science from the University of Cheikh Anta Diop of Dakar, Senegal. His main research interests include distributed systems, distributed database systems, design of inter-operable open systems, and performance evaluation of stochastic complex systems. He has authored or co-authored more than 15 international conference papers and journals. He is the coordinator of USCP/ Manager of the project VAE, Ministry of Youth, of the Vocational Training and Employment.

Jaime Lloret (M'07 – SM'10), received his M.Sc. in Physics in 1997, his M.Sc. in electronic Engineering in 2003 and his Ph.D. in telecommunication engineering (Dr. Ing.) in 2006. He is a CCNP Instructor. He is Associate Professor in the Polytechnic University of Valencia and he is the research line coordinator of the "communications and remote sensing" of the Integrated Management Coastal Research Institute. He is currently the Cognitive Networks Technical Committee (IEEE Communications Society) Vice-chair for the Europe/Africa Region. He more than 210 research and educational papers published in national and international conferences, international journals (most of them with Impact Factor in Journal Citation Report), and books. He has been the co-editor of 15 conference proceedings and guest editor of several international books and journals. He is editor-in-chief of the international journal "Networks Protocols and Algorithms", of "Advances in Network and Communications", IARIA Journals Board Chair, and he is associate editor of several international journals. He has been involved in more than 150 Program committees of international conferences and in several organization and steering committees. He has been the general chair of SENSORCOMM 2007, UBICOMM 2008, ICNS 2009 and ICWMC 2010 and co-chairman of ICAS 2009 and INTERNET 2010. He is the co-chairman of IEEE MASS 2011 and SCPA 2011. He is IEEE Senior Member and IARIA Fellow Member.

Chapter 4

Simulation Framework for Real-Time Database on WSNs

This chapter consists of the following article:

Simulation Framework for Real-Time Database on WSNs

O. Diallo, J. J. Rodrigues, M. Sene, J. Lloret

Journal of Network and Computer Applications, Elsevier, ISSN: 1084-8045, Vol. 39, March 2014, pp. 191-201, DOI: 10.1016/j.jnca.2013.07.001.

According to 2012 Journal Citation Reports published by Thomson Reuters in 2013, this journal scored ISI journal performance metrics as follows:

ISI Impact Factor (2012): 1.467

ISI 5-Years Impact Factor (2012): 1.251

Journal of Network and Computer Applications -- (----) ---- – ----

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

ARTICLE IN PRESS

Simulation Framework for Real-Time Database on WSNs

Ousmane Diallo[a,b], Joel J.P.C. Rodrigues[a,*], Mbaye Sene[b], Jaime Lloret[c]
[a]*Instituto de Telecomunicações, University of Beira Interior, Portugal*
[b]*Department of Mathematics and Informatics, UCAD, Senegal*
[c]*Integrated Management Coastal Research Institute, Universidad Politécnica de Valencia, Spain*

ARTICLE INFO

Article history :
Received 21 December 2012
Received in revised form 7 July 2013
Accepted 7 July 2013

Keywords :
Wireless sensor networks
Real-time database management
Model
Simulator
Performance Evaluation

ABSTRACT

Wireless sensor networks (WSNs) have been the focus of many research works. Nowadays, because of the time-critical tasks of several WSN applications, one of the new challenges faced by WSNs is handling real-time storage and querying the data they process. This is the real-time database management on WSN and it deals with time-constrained data, time-constrained transactions, and limited resources of wireless sensors. Developing, testing, and debugging this kind of complex system are time-consuming and hard work. The deployment is also generally very costly in both time and money. Therefore in this context, the use of a simulator for a validation phase before implementation and deploying is proved to be very useful. The aim of this paper is to describe the different specificities of real-time databases on WSN and to present a model for a simulation framework of the real-time databases management on WSN that uses a distributed approach. Then, the model of the simulator is described and developed in Java and a case study with some results demonstrates the validity of the structural model.

1. Introduction

Wireless sensor networks (WSNs) may be defined as a set of smart devices, called sensors, which are able to sense, process and transmit information about the environment on which they are deployed. These devices, distributed in a geographical area, collect information for users interested in monitoring and controlling a given phenomenon and transfer them to a sink node. The latter makes the information available to a gateway where remote users can access. So as to obtain those information, users use applications that communicate with the network through queries (Callaway, 2004; Sacks et al., 2003; Baronti et al., 2007).

Systems based on sensor networks are more and more used in many areas providing various types of WSNs (Mendes and Rodrigues, 2010).

These different WSNs involved the development of many applications, which are generally connected to databases treating the amount of data collected from sensors. However, the processing time becomes increasingly critical for certain applications. These applications must query and analyze the data in a bounded time in order to make decisions and to react as soon as possible (Diallo et al., 2011). Some examples of the most popular applications are the following: the control of network traffic (Cranor et al., 2002), transactional analysis (web, banking or telecommunication transactions) (Cortes et al., 2000), human motion tracking application (Chen and Ferreira, 2009), the tracking of actions on dynamic Web pages (Zhu and Shasha, 2002; Chen et al., 2000), monitoring of urban or environmental phenomena (Mainwaring et al., 2002; Ulmer et al., 2003), and the sensors data management (Arasu et al., 2003).

Once the sensors perform their measurement, the problem of data storing and querying arises. Indeed, the sensors have restricted storage capacity (Silva et al., 2004) and the ongoing interaction between network devices and environment results in huge amounts of data.

There are two main approaches for data storage and querying in WSN: distributed and warehousing (Bonnet et al., 2000b). The first approach aims at exploiting the capacities of calculation of sensors. Some queries are distributed and evaluated among the nodes into the network. The objective is to

*Corresponding author. Tel.: +351 27 524 2081; fax: +351 27 531 9899.
E-mail addresses: ousmane.diallo@it.ubi.pt (O. Diallo), joeljr@ieee.org (J.J.P.C. Rodrigues), mbaye.sene@ucad.sn (M. Sene), jlloret@dcom.upv.es (J. Lloret).

1084-8045/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2013.07.001

locally calculate in order to limit sending of messages, reducing thus the energy consumption. In the second approach, warehousing, one has a centralized system. Collected data from sensors (in stream) are sent to a central database server, in which user queries are processed. Note that this last technique generates large data flows.

The data collected by the WSN must closely reflect the current state of the targeted environment. However, the environment changes constantly and the data are collected in discreet times. So, the collected data have temporal validity, as time advances they become less and less accurate, until the time where they do not reflect the state of the environment (Idoudi et al., 2009a, 2009b). It is fundamental that responses to application queries ensure that returned data comply with logic and temporal constraints. In this context, real-time data management on WSNs is necessary to deal with those constraints. The main goal of real-time data management is to ensure temporal consistence data and process transactions within real-time constraints.

Developing, testing, and debugging this kind of complex system are time-consuming and hard work. The deployment is also generally very costly in both time and money. Therefore in this context, the use of a simulator for a validation phase before implementation and deploying is proved to be very useful (Sundani et al., 2010; Nasreddine, 2012; Egea-López et al., 2005).

The main contributions of this paper are the following:

- Description of the different characteristics of real-time data bases management on WSN.
- Proposition of a model for a simulation framework of the real-time databases management on WSN that uses a distributed approach.
- Implementation of the model and performance evaluation.

The remainder of this paper is organized as follows: Section 2 presents the wireless sensor networks. Section 3 discusses the related work, while the characteristics of real-time database management systems are exposed in Section 4. Section 5 briefly presents the data management on WSNs. In Section 6, the description of the real-time database simulation model is provided. In Section 7, some screenshots of the case study is presented to validate the model and Section 8 concludes the paper.

2. Wireless sensor networks

Generally a WSN has a large number of nodes distributed on an interest area and communicating between them so as to measure a physical quantity (e.g., pollution level in a given area) or to do an event monitoring (e.g., vehicles tracking). WSNs are used with different applications in many areas and are very important for applications that should be deployed in places hostile to human interventions (e.g., volcano monitoring). Each network node is considered smart and embeds these units: a sensor unit which provides a measure of environmental data (such as temperature, humidity, pressure, acceleration, sound, etc.), a processing unit, a storage unit, a communication unit, and an energy unit. The communication unit usually performs data transmission by means of radio (Akyildiz et al., 2002, 2007; Oliveira et al., 2011a; Oliveira and Rodrigues Joel, 2011b).

However, these resources are generally very limited, particularly those of storage and energy. The sensor network lifetime depends on the available energy in the nodes composing the network (Lin et al., 2011, 2012). This available energy is consumed by three activities (Akyildiz et al., 2002): sensing activity (data acquisition from the environment), communication (sending and receiving packets), which is essential to form a WSN, and data processing, which consists in some operations applied over data by smart sensors (Cho et al., 2001; Madden et al., 2005). However, the sensing and processing activities are much less expensive in energy consumption than the wireless communication activities (Madden et al., 2002). Therefore, saving energy by optimizing the communication activities is the main point attention in many algorithms designed for sensor networks.

Generally, proposed architectures for WSNs are based on the distribution of sensor nodes in a geographical area in such a way that sensors send collected data to a base station by using the routing techniques like in (Li et al., 2011); Al-karaki and Kamal, 2004).

3. Related work

A lot of simulation tools for WSNs have been already proposed. These simulators are generally designed to meet the development constraints (e.g. communication constraints, constraints on the nodes) related to WSNs and can be divided into classes according to the nature of the specified constraints.

The first class of simulators is the oriented network. They emphasize on the behavior of the wireless network and the protocol stack of the operation. These simulators are based on the computer network simulators designed for computers, such as NS-2 (Zhang et al., 2009), OMNeT (Xian et al., 2008; Varga, 2001), J-Sim (Sobeih et al., 2006), etc. Among these simulators it may be noted: SensorSim (Park et al., 2000), Castalia (Boulis, 2007, 2009), J-Sim sensor simulator (Sobeih et al., 2006), and Prowler (Simon et al., 2003). SensorSim is built on top of a NS-2 802.11 network model and it models a software model of the sensor node and a hardware model. The power models of the hardware components have been implemented and the state of the hardware model changes according to the function performed by the software model. Therefore, the power consumption of the network could be simulated. However, the CPU has not been implemented (Park et al., 2000). Castalia is an application-level simulator for wireless sensor network based on OMNeT++. It is parametric and can be used to evaluate various characteristics for specific applications. In Castalia, multiple simple modules can be linked together and form a compound module, e.g. a sensor node. The simple modules implement the atomic behavior of a model, for instance, the network stack layers of a sensor. Node modules are connected to wireless channel and physical process modules (Boulis, 2007, 2009). J-Sim has been developed at the University of Washington by the National Simulation Resource. It is a general-purpose network simulator modeled after NS-2, but unlike NS-2, J-Sim uses the concept of components, replacing the representation of each node as an object. J-Sim provides support for sensors and physical phenomena. Moreover, the energy modeling, except the radio energy consumption, is provided for sensor networks (Sobeih et al., 2006). Prowler is a probabilistic wireless sensor network simulator running on Matlab environment. It is an event-driven simulator that can be set to operate, either in deterministic mode, or in probabilistic mode in order to simulate the nondeterministic nature of certain component of the network such as the communication channel. Prowler is adapted to the development of algorithms and optimized protocols. However, it does not model the energy consumption of the sensor node (Simon et al., 2003).These simulators are oriented network and they do not deal with the databases management techniques, and particularly the real-time databases management techniques.

The second class of simulators is oriented node. They empha-size on the function of a single node with simple communication models. They are specific to targeted nodes and their operating systems. In addition, these simulators are used to verify the compatibility of a node with a given application. Among these simulators include: TOSSIM (Levis et al., 2003), ATEMU (Karir et al., 2004), and SEN (Sundresh et al., 2004). TOSSIM is a discrete event simulator for TinyOS (TinyOS, 2012) wireless sensor networks. TOSSIM can capture the behavior of the network of several TinyOS nodes at bit granularity. It is designed specifically for TinyOS/NesC applications to be run on MICA Mote platforms. With TOSSIM, designers can easily translate between running an application in the simulation environment and on motes (Levis et al., 2003). ATEMU is an instruction-level cycle-accurate emulator for WSN written in C language. It simulates programs of nodes with an accuracy clock cycle. ATEMU provides support for the AVR processor, and other function units on the MICA2 sensor node platform, such as the transceiver. Moreover, it provides users a complete system for debugging and monitoring the execution of their code via a GUI, called Xatdb (Karir et al., 2004). Sensor Environment and Network Simulator (SENS) is a wireless sensor network simulator with a layered architecture, modular. The components of this architecture are customizable and they model the application layer, network communication, and physical environment. SENS allows realistic simulations by using values from real sensors to represent the behavior of component implementations (Sundresh et al., 2004).

All these simulators do not fit within the real-time databases on WSNs. To the best of our knowledge, there is not a tool that takes into account the specific characteristics of the models of real-time databases for WSNs. However, true that our simulator is not a full end-to-end simulator like for instance OMNeT or any other full end-to-end simulator tool with all protocols of WSNs, it is based on a model that focuses on the real-time databases management on WSNs, which are by their characteristics and constraints very difficult to study and model.

4. Characteristics of data and transactions in real-time database systems

Like a traditional database management system (DBMS), a real-time DBMS (RT-DBMS) must process transactions and ensures that the logical consistency of the data is not violated. However, unlike a traditional DBMS, a RT-DBMS emphasizes on the temporal validity of the data and the time constraints or deadlines for transactions (Idoudi et al., 2009a, 2009b; DiPippo and Wolfe, 1997).

The main purpose of a RT-DBMS is to process transactions on time, while maintaining logical and temporal consistency of data. The temporal consistency expresses the need to maintain consistency between the current state of the targeted environment and the state as reflected by the database contents. The temporal consistency can be measured in two ways (Ramamritham, 1993):

- Absolute consistency, which deals with the need to maintain the view representing the state of the targeted environment consistent with the real state of the environment.
- Relative consistency, which concerns data derived from other ones.

To satisfy these temporal constraints, the structure of the data must include these attributes: (i) timestamp, which indicates the instant when the observation relating the data was made; and (ii) absolute validity interval (*avi*) that denotes the time interval following the timestamp during which the data are considered valid. Another attribute can be considered; the imprecision or data error (*DE*), which refers to how the current state of the targeted environment may differ from the measured data (Ramamritham, 1993). The data error on a data version d is defined by:

$$DE(d) = 100 * \left|\frac{CurrentValue(d) - UpdateValue(d)}{CurrentValue(d)}\right| (\%) \qquad (1)$$

There are two kinds of transactions in real-time databases: update transactions and user transactions. Update transactions are executed periodically to update real-time sensor data, or sporadically to update the derived data in order to reflect the state of the real world. Derived data are the data computed using sensor data. User transactions, representing user queries, arrive aperiodically. They may read or write non real-time data, but only read real-time data (Idoudi et al., 2009a, 2009b). These update transactions also to comply with temporal constraints must have these following attributes: (i) liberation time that represents the moment on which all the resources for the transaction processing is available; (ii) computing time that indicates the execution time needed for the transaction; and (iii) maximum time, which indicates the maximum time limit for the transaction execution and the periodicity that refers to the frequency with which the transaction happens (Chagas et al., 2010).

To satisfy the logical consistency of the data, transactions must be processed with Atomicity, Consistency, Isolation and Durability (ACID) properties. But unlike the conventional databases, in real-time databases these properties are relaxed. Firstly, the atomicity may be relaxed. It is only applied to the sub-transaction that wants to deal with completely data consistency. Secondly, since timeliness is more important than correctness, in many situations, correctness can be traded for timeliness. Thirdly, the isolation allows transactions to communicate with others to better perform control functions. Similarly, in real-time databases, not all data must be permanent and some of them are temporal (DiPippo and Wolfe, 1997; Chagas et al., 2010; Ramamritham, 1993).

5. Data management on wireless sensor networks

Among the activities of wireless sensor nodes, the communication activities are more expensive in terms of energy consumption (Madden et al., 2002, 2005). Therefore, saving energy by optimizing the communication activities is the main point attention in many algorithms and architectures designed for sensor networks (Madden et al., 2005). There are two main approaches to data storage and query in WSNs (Bonnet and Seshadri, 2000; Bonnet et al., 2000): the warehousing approach and the distributed approach.

The warehousing approach is a centralized system (Bonnet et al., 2001). The data gathered by the sensors are sent to a central database server where user queries are processed. In this case, the sensors act as simply collectors. The warehousing approach is the most used one in data storage and query processing. However, it has some drawbacks, such as, the huge number of generated data can easily create a bottleneck on the central server; and the huge amount of information transferred wastes the resources. The wireless data transmission decreases consequently the WSN lifetime. Indeed, it is shown in (Madden et al., 2002) that the wireless data transmission is more expensive in terms of energy consumption than the data processing and the sensing activities. Moreover, it is clear that this approach is not adequate for non-historical queries because it involves time delay for the results.

In the other hand, the distributed approach exploits the capabilities of sensors calculation and the sensors act as local databases (Bonnet and Seshadri, 2000). It aims to locally calculate in order to limit sending of messages, reducing thus the energy consumption in the network. In this approach, the data are stored in a central database server and in the devices themselves, enabling one to query both. Here, the devices act as part of the database. This approach can offer several advantages: on sensors, the processing of queries is done on quasi real-time. In fact, query processing inside the devices themselves means that the most current data will be acquired. The data will certainly arrive to the users with their temporal validity. It supports long-running queries and instant queries. Furthermore, the distributed approach increases the lifetime of the network by reducing the communications activities among the devices.

There are three sorts of queries in sensor networks: (i) historical data queries, which are run against the server; (ii) instant queries, which are run against a device in an instant of time; and (iii) long-running queries, which refer to queries run against a device during a time interval (Bonnet and Seshadri, 2000; Neto et al., 2004).

6. Real-time database simulation model for WSNs

The design of the simulator involves the development of a library of models from which it will be possible to build our architecture of real-time databases management for WSN. These models are descriptions of behaviors and functions supported by each component of the system and can replace, in the simulation, the actual components of the architecture to be modelled.

Many researchers argue that the object-oriented model is more suitable than the relational model (Kim, 1995) to model real-time data, because of the nature of several real-time applications, which handle complex real-world objects with time constraints. Thus, many projects on real-time databases have chosen the object-oriented model for their systems (Terrier et al., 1997; Wolfe et al., 1997). However, the relational model (Madden et al., 2005) is the most used on real-time or distributed databases modelling.

The global architecture of our simulator model of real-time database management techniques on WSNs (see Fig. 1) uses the distributed data storage and the query approach and is inspired by the works made in (Neto and Perkusich Maria, 2004). Thus, our model considers the gateway more general by integrating into it a database warehousing (DBW) that stores real-time data and a real-time database management system (RT-DBMS) that handles real-time transactions. The model of the wireless sensor network (WSN) nodes is built by using an event-based model, particularly an event-advance design (Granat, 2006). Thus, the nodes are programmed as follow: each time an event (user transaction) occurs in the network; all the nodes are triggered to compute the previous acquisitions, store the sensor data, and send all the underlying periodic update transactions to the database server. Moreover, each node targeted by the user transaction at the same time, after completion of its previous works, sends the results on required time. All these actions are scheduled on time based on the instant time that the network was initialized, the periodicity of acquisition and database update transactions, and the instant time that the user transaction arrives in the network. Therefore, the devices act as local databases, including a software component that manages the data storage and user queries with time constraints. The database server incorporates also an application that deals with time-constrained data and time and logic constrained transactions. We distinguish two kinds of transactions: update transactions and user transactions. The update transactions can be local updates for periodically updating the local sensor databases or server updates that periodically update the real-time data in the database server. The user transactions are real-time user queries that only read real-time data or historical queries that may read or write historical data.

Below, the description of the behavioral models of various elements that compose the simulator model of real-time database management techniques for WSNs is made, by explaining clearly the principle of object-oriented modelling used in the simulator developed in Java.

6.1. Wireless sensor network

The WSN is composed by smart sensors (Ruiz et al., 2004), which are able to sense, process and transmit information about the environment on

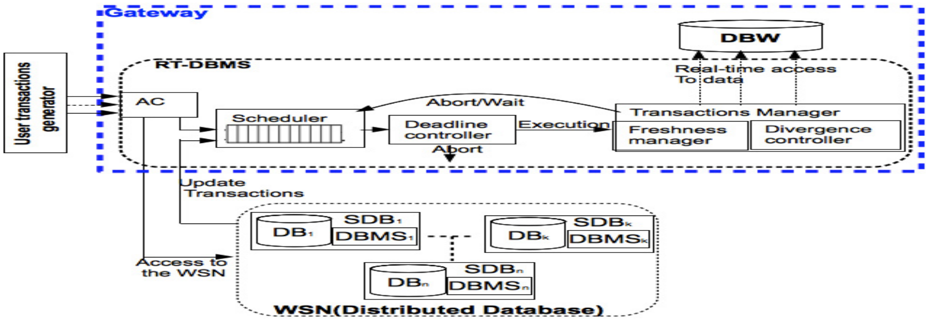


Fig. 1. Components of the simulator model for real-time database management techniques on WSNs.

which they are deployed. These sensors act as local databases, named *sensor database* (*SDB*). Thus, the WSN forms a distributed database. An illustration of a generic sensor model with its basic components is given in Fig. 2.

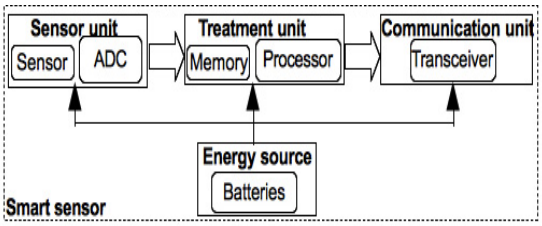


Fig. 2. Illustration of a sensor generic model.

The *sensor unit* is formed by the *sensor* and the Analog-to-Digital Converter (*ADC*). The sensor senses a change in the physical condition and transforms it to electrical signal. This signal is converted by the *ADC* to digital data, which can be stored in the *memory* and processed by the *processor*. The processor and the memory compose the *treatment unit*. The data communication into the network is performed by the *transceiver* located in the *communication unit*. This communication can be performed with or without wire. In the case of wireless communication, the most used technique is the radio frequency transmission (Adelstein et al., 2005). The *batteries* are used as energy sources.

For modelling our generic sensor as a local database, we use an object-oriented modelling as follow: a *SensorNetworkNode* class encapsulates all the properties and functionalities of the device. Thus, the measured physical phenomenon, because our case study uses temperature measures, is modelled by the *next Gaussian( )* function of the class *Random* that will generate pseudo-random available temperature measures (Deshpande et al., 2004) with best precision. This function is used in the acquisition phase (*acquisition ()* function in our model). However, depending in the case study, others methods can be used, for example, a memory that contains the measures of the physical phenomenon (Nasreddine, 2012). Depending of the sampling frequency of the local updates, the *acquisition()* function is called in each period and the temperature value returned by the *nextGaussian( )*function is stored in coherent way in the memory (list of sensor data) of the sensor that stores sensor data. Moreover, functions that periodically send the acquired data, according to the periodicity of the database updates, to the database server are implemented. Concerning the other components of the sensor model, functions that receive queries, process them in real-time according to their time constraints, and send the data to the users are added. Thus, these functions model the *treatment unit* and the *communication unit*. For the energy consumption, functions are also used to retrieve an amount of energy from the global energy remained on the sensor according to the data sensing, processing, transmission, and the sensor's duty cycle operation. Formula (2) below gives the energy consumption of the sensor node. For more details about the calculation of each term of the formula, readers can refer to (Briand et al., 2010).

$$E_{node} = E_{sensor} + E_{ADC} + E_{\mu c} + E_{transceiver} \qquad (2)$$

To capture the actual energy consumption of each sensor's activity in the simulation, each activity needs to be calibrated by real measures. For that, the current consumptions of the main sensor's activities are configured according to the TelosB datasheet (CrossBow Inc, 2013; Nguyen et al., 2011), as presented in Table 1. In addition, the simulation used AA-size batteries with each nominal ampere-hour capacity of 2.4 Åh. However, the values of the parameters (the initial battery capacity according to its type, the cost of the operations with respect to the sensor type, etc.) can be configured in the main configuration file.

In real-time databases, the data should comply with logic and temporal constraints. Thus to comply with temporal constraints, the structure of the data should have these properties: (i) timestamp, (ii) absolute validity interval (avi), and the imprecision

(Ramamritham, 1993). All these attributes including the value of the data and the location-identification of the sensor that acquires the data are encapsulated in our model of sensor data with a class named *SensorData*, which implements the data acquired by a sensor.

Because of the temporal constraints of sensor data, they become quickly invalid and should be periodically updated. Thus to deal with theses constraints, transactions have logical and temporal constraints. In the sensors we distinguish two kinds of transactions: the *local update transactions* and the *read-only transactions*. The local update transactions, represented in our implementation by the *LocalUpdate* class, represent a new acquisition of a data that will be stored in the sensor database. These transactions are periodic and firm (Pailler, 2006; Neto and Perkusich Maria, 2004) and they are executed by call of the *acquisition()* function. The read-only transactions are the transactions sent by the user applications to read the local databases.

Table 1
Nominal power consumption of TelosB nodes (Nguyen et al., 2011).

Components	Mode	Current draw
Module	Active	1.8 mA
	Sleep	5.1 µA
	Receive	19.7 mA
RF-Transceiver	Transmit(at 0dBm)	17.4 mA
	Sleep	0.01 mA

In our case study, the transactions are the instant query transactions and the long-running transactions. They can be executed by calling the *instantQueryExecution()* and *longRunningQueryExecution()* functions respectively. These kinds of transactions are aperiodic.

As already mentioned above, the processing model in the sensor nodes is built by using an *event-based model*, particularly an *event-advance design* (Granat, 2006). Thus, the nodes are programmed as follow: each time an event (a read-only transaction) arrives in the network, all the nodes are triggered to compute the previous acquisitions, store the sensor data, and send all the underlying periodic update transactions to the database server. Moreover, each node targeted by the user transaction at the same time, after completion of its previous works, sends the results on required time. All these actions are scheduled on time based on the instant time that the network was initialized, the periodicity of acquisition and database update transactions, and the instant time that the user transaction arrives in the network.

The implementation of this event-advanced design technique is done as follow: we know the instant time that the network was initialized, let call it *initialize-time*. We know the arrival time of the user query, let call it *now*, and finally we know the *periodicity* (let call it *T*) of the local update transactions. Thus, the difference between *now* and *initialize-time*, divided by the *periodicity* (*T*) gives us the number of local updates (let call it *Q*) that the sensor should perform before processing the query. Finally, having the release time (the date on which the first instance of the query should be activated) included in the local update transaction, let call it *r*, we can compute the activation dates for each instance of the local update transaction with this formula (Pailler, 2006):

Let $Q \in N^{+}$, by using Q the size of a loop with k as index started at *one*, then the waking date of the k^{th} instance of the local update transaction is :

$$r^{k} = r + (k-1) * T \quad (3)$$

The same technique is used to calculate the activation date for all the instances of the server update transactions (explained hereafter) that should be done before the arrival of user transactions. Thus, by scheduling all the instances of the local and server update transactions, and the user transaction with respect to their activation dates, we can manage the data storage and transaction processing with respect to the logic and time constraints of data and transactions.

6.2. User transactions generator

The transactions generator is a component, which is responsible for generating the user transactions aperiodically. Thus, it is composed of an interface where the user transactions will be parametered to be sent into the system and a pseudo-random function that provides the time intervals representing the sending moments of transactions. As mentioned above, the properties of a specific transaction are also encapsulated into a specific transaction class. For example using the transaction class, a long-running query is parametered as: *Transaction <idTrans = 12 startTrans = "Dec 03,2012 11:09:34" perTrans = 1 time unit (t.u.), idLocSensor = 0..5, endTrans = "Dec 03,2012 11:09:40">*. In this example, the query numbered 12 is interested of the values of sensor readings for sensors in the locations 0 through 5. The query should be executed in each targeted sensor from from Dec 03, 2012 11:09:34 to Dec 03, 2012 11:09:40 and the values is sampled each 1 t.u.

6.3. Gateway

The gateway usually is used to make the connection between the wireless sensor network and the external networks (e.g. IP). It has the same power as a PC. In our model the gateway is more general by including a database warehousing (DBW) that stores real-time data and a real-time database management system (RT-DBMS) that handles in real-time the transactions.

6.3.1. RT-DBMS

The *admission controller* (*AC*) has the task of controlling user transactions that are accepted in the system. Thus, the user transactions that represent instant or long-running queries will be sent to the WSN, whereas the real-time user transactions and the historical queries will be directed towards the server database.

In that phase, we consider now the real-time user transactions that arrive aperiodically to read the real-time data storing in the database server, modeled by the *RTreadTransaction* class and the real-time update transactions that arrive periodically from the WSN for updating the real-time data in the server database, modeled by the *ServerUpdate* class (Idoudi et al., 2009b). Thus, the real-time user transactions include among other properties their *computing time* (Pailler, 2006) and the real-time update transactions have among other attributes (i) the *liberation time*, (ii) the *computing time*, (iii) the *deadline*, and (iv) the *periodicity* (Chagas et al., 2010).

The real-time user transactions admitted in the system and the server update transactions from the sensors are placed in a *waiting queue* before being sent to the *transaction manager*. All these transactions in wait are scheduled by the *scheduler* according to their priorities, particularly their deadlines. In fact, we use the *Earliest Deadline First (EDF) protocol* (Liu and Leyland, 1973), which orders the transactions according to the principle that the transaction that has the nearest deadline must be executed in priority.

The *transaction manager* consults regularly the waiting queue, updates it, and checks the validity of transactions (i.e. transactions that have not yet missed their deadline in relation to the current time). The transactions that have lost their deadline are aborted, the others are still handled by the transaction manager for a possible execution: the *freshness manager* checks the freshness of data that will be accessed by a transaction. If the data is outdated then the transaction is waiting in a queue. Since several transactions can access the data stored on the database server, a *concurrence control protocol* is needed to deal with the cases in which concurrent transactions may lead to inconsistent data in the database. In our model we use the *Epsilon Serializability* (*ESR*) criterion (Pu, 1991) to deal with concurrent transactions. Indeed, the *Epsilon Serializability* (*ESR*) criterion is well adapted in the processing of firm (the deadline of the transaction must be met, otherwise the transaction is rejected) real-time transactions (Bouazizi et al., 2004).

The *ESR* relaxes the severity of the classic serializability (*SR*) in the transaction processing by allowing a limited inconsistency in the database. This limited inconsistency is automatically maintained by the divergence control algorithms (*DC*), in the same way that the SR is managed by the concurrency control techniques (*CC*). In our model we use the *DC* algorithm with two phases (Bouazizi et al., 2004) named *2PLDC*, which is an extension of the algorithm of *2PL* concurrency control of the classic serializability. In this algorithm, the update transactions called write Epsilon-transaction (*ETw*) export an inconsistency in the database, while user transactions called read Epsilon transaction (*ETr*) import an incoherence from the base. These inconsistencies are gathered in two accumulators and the transactions can continue their executions as long as these accumulators do not exceed a certain given limit value.

Thus, in addition to the attributes mentioned early in this paragraph, the real-time update transaction becomes an *ETw* and incorporates two other attributes in our model: an attribute that records the total amount of inconsistency exported in the database and an attribute that records the maximum imprecision that the real-time update transactions (*ETw*) can export in the database. These attributes are modeled respectively by *accumuExport* and *limExport* in the model. Similarly, the real-time user transaction becomes an *ETr* and has two more attributes: an attribute that records the total amount of inconsistency imported by the *ETr* modeled by *accumuImport* and an attribute that records the maximum imprecision that the *ETr* can import modeled by *limImport* in the model. All such characteristics will permit to deal with both the time and logic restrictions.

This divergence control algorithm is implemented to allow conflicting transactions (*Read/Write* in the serialisability classic) to execute concurrently in way that their scheduling does not lead an imprecision that is higher than the one accepted in the data. Thus, when an *ETr* wants to access a data being running by an *ETw*, we increment the value of the variables *accumuImport* and *accumuEx-port*, respectively of the *ETr* and the *ETw* and verify if :

$$accumuImport \leq limImport \quad (4)$$

$$accumuExport \leq limExport \quad (5)$$

O. Diallo et al. / Journal of Network and Computer Applications ■ (■■■■) ■■■–■■■

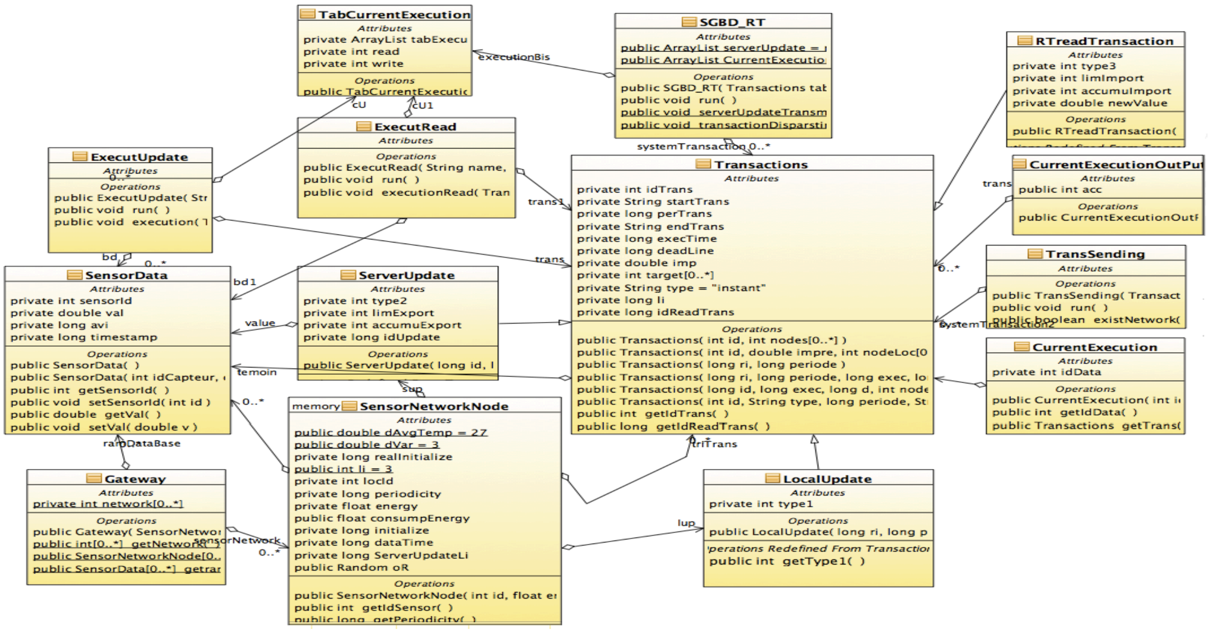


Fig. 3. Class diagram of the model.

In the same fashion, when an *ETw* demands to access a data being executed by several *ETr*, all the *ETr* increments by one unit their *accumuImport* and verify if formula (4) is true. However, the *ETw* should verify its *accumuExport* regarding the incrementation by the number of *ETr* in conflict with it. If condition (5) is false or if any *accumuImport* of one *ETr* exceeds its *limImport*, then we make wait or abort the transaction which demands to access the data in processing. Thus, this mechanism allows several real-time transactions to meet their deadlines.

6.3.2. Database warehousing

The sensor data are modelled by the *SensorData* class, which encapsulates these attributes: the location identification of the sensor that acquires the data, the timestamp, absolute validity interval (*avi*), the imprecision, and the value of the data. Thus, the database warehousing (*DBW*) is a list of a sensor data and is modelled by the *Memory class*. The size of this memory can be fixed on the beginning of the simulation experience. The coherency of the *DBW* is maintained by the real-time update transactions and the Epsilon Serializability (*ESR*) criterion. This allows to simulate a real-time database as in the real databases.

Fig. 3 illustrates the class diagram of the implementation of the whole simulation framework.

7. Performance evaluation of a simulator model for real-time databases management techniques in WSN based on a distributed approach

The model considers a WSN where the sensor nodes, by using an event-based design, receives user queries, computes the previous acquisitions of temperature, send all the underlying periodic update transactions to the database server, and process user queries with respect to time and logic constraints as explained early. The *RT-DBMS* receives the real-time update transactions and processes them according to logic and time constraints of data and transactions. Thus, the *RT-DBMS* schedules the transactions with respect to the *Earliest Dead-line First* (*EDF*) protocol (Liu and Leyland, 1973) and uses the *Epsilon Serializability criterion* to deal with concurrent transactions.

The execution of the simulator is made by setting up some parameters according to a particular experiment. Indeed, the simulator is implemented in Java programming language and in the file containing the *MainSimulation* class one can fix almost all the parameters of the simulation, e.g., the number of sensor nodes composing the network. Each sensor node has to be configurated also in this file by setting up the attributes (identification of the sensor location, the maximum power of the batteries, the parameters of the local and sever updates) of the *SensorNetworkNode* class. In addition, the practitioner has to configure here the maximum number of user transactions (instant query transactions and long-running query transactions) and their characteristics to be sent to the network nodes. The real-time user transactions (*RTreadTransaction* class) that want to access the real-time data storing in the database server has to be configured also in that file. Thus for analysis the validity of the model, five (5) sensors forming the sensor network were considered in the configuration file. Table 2 summarizes the configurations of the local updates (periodic acquisitions) and the server updates (periodic updates of the database server) of each sensor node.

The parameters in the above table are specific to those transactions. The transactions are like the Query Language for Real-Time Database (*QL-RTDB*) (Chagas et al., 2010) with properties encapsulated in a class as already explained early in the paper. This class, depending of the transaction type, is composed of attributes that indicate the identification of the transaction (*idTrans*), the liberation time (*Li*), the computational time (*Ct*), the periodicity (*Pr*), the deadline (*Dl*), the total amount of inconsistency exported in the database or imported from the database

Table 2
Summary of the configuration parameters of the sensor.

Server updates idLocSensor	Li in time unit (t.u.)	Ct (t.u.)	*Dl* (t.u.)	Pr (t.u.)	Local updates Li (t.u.)	Local updates Pr (t.u.)
0	3	2	12	12	1	6
1	9	4	18	18	9	9
2	4	2	14	14	1	7
3	5	3	15	15	3	7
4	2	2	12	12	2	6

(*accumu-Export/Import*), the imprecision limit (*Lim-Imp/Exp*), and the location identifications (*idLocSensor*) of the temperature sensors targeted by the transaction. The syntax is like*: Transaction <idTrans, Li, Ct, Pr, Dl, accumu-Export/Import, Lim-Imp/Exp, idLoc-Sensor0..idLocSensorN>.*

This experiment considers also two real-time user transactions (*rt-uT*): a real-time user transaction with an identification *500*, a computational time *6 t.u*., a deadline *12 t.u*., and a real-time user transaction with an identification *600*, a computational time *5 t.u*., a deadline *12 t.u.*. The user transaction with an identification *500* demands to read a data sent by the sensor in location *0* from the real-time database server and the user transaction with an identification *600* wants to read the data item sent by the sensor *1*. These transactions are aperiodically sent towards the database server.

According to the *Epsilon Serializability criterion*, the server update transactions and the real-time user transactions include respectively an attribute that stores a value representing the amount of inconsistency exported in the database and an attribute that stores a value representing the amount of inconsistency imported from the database. The structure of these transactions has also a value that represents the limit of imprecision exported in the database and a value that represents the limit of imprecision imported from the database. For more efficient analysis, these values are generated by a random function in the model.

Depending on the parameters and transaction used, the results are given by numerical values.

7.1. Case where an instant query is sent to the network to query sensor data

We first analyze the case where an instant query is sent in the network. The properties of these queries are encapsulated in class that is composed of attributes that indicate the identification of the query and the locations of the temperature sensors that the user wishes to query. The syntax is like: *Transaction <idTrans, idLocSensor0..idLocSensorN>*. As case study, the instant query *<idTrans=0, idLocSensor0..idLocSensor1>* gives the results in the Table 3.

Table 3
Summary of the results of the instant query that queries the temperature sensors in locations 0 and 1.

Instant query with an idTrans=*0*, idLocSensor=*0..1*						
IdLocSensor	Access date	Returned data(°C)	AVI (t.u.)	Timestamp	Number of local updates (lup)	Number of server updates (sup)
0	Oct 31, 2012 23:17:22	28.032	14	Oct 31, 2012 23:17:18	3	2
1	Oct 31, 2012 23:17:22	28.033	14	Oct 31, 2012 23:17:14	1	1
2	X	X	X	X	3	1
3	X	X	X	X	2	1
4	X	X	X	X	3	2

An instant query is a query that is run against a device in an instant of time. Thus, according to Table 3, the query arrives in the network at *Oct 31, 2012 23:17:22* and is immediately answered by the targeted sensors. By using an event-advance desing, the sensor nodes are based on formula (3) and compute all the previous local updates and send all the underlying server updates towards the database server before returning the required data with their temporal parameters. Thus, the simulation model shows also for each sensor the number of local updates and server updates to compute before processing the query. Here, only the sensors in locations *0* and *1* reply because they were targeted by the query. However, the remaining sensors perform their local and server updates.

Once the server updates (*sup*) arrive in the Gateway, they are handled with the real-time user transactions (*rt-uT*) by the *RT-DBMS* with respect to time and logic constraints (see Table 4).

Table 4
Summary of the sequence of executions of real-time transactions in the database server.

idTrans	idLocSensor	Li	Ct (t.u.)	Pr (t.u.)	Dl (t.u.)	LimExp	LimImp	Value (°C)
2000 (sup)	4	Oct 31, 2012 23:17:07	2	12	12	2	X	21.152
3000 (sup)	0	Oct 31, 2012 23:17:08	2	12	12	1	X	27.964
4000 (sup)	2	Oct 31, 2012 23:17:09	2	14	14	2	X	21.914
5000 (sup)	3	Oct 31, 2012 23:17:10	3	15	15	4	X	26.452
9000 (sup)	1	Oct 31, 2012 23:17:14	4	18	18	8	X	28.033
2000 (sup)	4	Oct 31, 2012 23:17:19	2	12	12	2	X	21.151
3000 (sup)	0	Oct 31, 2012 23:17:20	2	12	12	1	X	28.032
500 (rt-uT)	0	Oct 31, 2012 23:18:45	6	X	12	X	6	28.032±0.241%
600 (rt-uT)	1	Oct 31, 2012 23:18:45	5	X	12	X	0	28.033±0.0%

Table 4 presents the sequence of both real-time user transactions and real-time update transactions in the server database. It is also possible to observe the time properties for each transaction, the sensor performing the updates or the sensor data targeted by the user transaction, the parameters to deal with the Epsilon Serialisability, and the data value added or read including its eventual error. The liberation times (*Li*) indicate the executions of the transactions according to their periods. Considering the quality of service management, the model uses the Earliest Deadline First (*EDF*) protocol (Liu and Leyland, 1973) to schedule the transactions and performs the Epsilon Serializability techniques to deal with concurrent transactions. Formulas (4) and (5) are used here to allow or forbid the simultaneous execution of transactions. Thus as illustrated in the table, the user transaction *500* allows to read the data with an imprecision *0.241%*, while the user transaction *600* does not allow any imprecision in the read value.

The model allows also to compute the energy consumed in the network, see Table 5.

Table 5 above presents the percentage of energy consumed and remaining energy for each sensor node according to its activities performed. These activities are mainly data sensing, processing, communication, and the sensor's duty cycle operation. In the table, one can observe that the sensors in locations *0* and *4* have spent most of the energy. This is due to the fact that they perform more server updates (see Table 3), leading then to more data transmission. In fact generally in WSNs and with respect to the TelosB datasheet (see Table 1), the wireless communication consumes more energy than the other activities performed by the sensors. The energy consumed by the sensor *0* is the biggest because, regarding the sensor *4* of which it has the same number of local updates and server updates, it executes the instant query and sends the result to the base station. This leads to more energy consumption. The energy consumption is computed based on formula (2) in (Briand et al., 2010).

7.2. Case where a long-running query is sent to the network to query sensor data

The configurations of the network and the two real-time user transactions addressed to the database server are not changed. However, we sent a long-running query to trigger the process of the network. A long-running query refers to a query runs against a device during a time interval (Bonnet and Seshadri, 2000; Neto et al., 2004). The characteristics of the query are encapsulated in class that is composed of attributes that indicate the identification of the query (*idTrans*), the initiate date of the query execution (*startTrans*), the period of the data sampling (*perTrans*), the terminate date (*endTrans*) of the query execution, and the locations of the temperature sensors (*idLocSensori*) that the query targets. The syntax is: *Transaction <idTrans, startTrans, perTrans, idLocSensor0..idLocSensorN, endTrans>*. For the case study, the long-running query parameters are:

<idTrans=2, startTrans="Sep 03, 2012 11:09:34", perTrans=1 t.u., idLocSensor=0..1, endTrans="Sep 03,2012 11:09:37">. The results are summarized in Table 6.

Table 5
Summary of the energy consumed in the network.

idLocSensor	Consumed energy (%)	Remaining energy (%)
0	1.64817083	98.3518292
1	1.2104375	98.7895625
2	0.88565	99.11435
3	0.84804375	99.1519563
4	1.24817083	98.7518292

Table 6 illustrates the execution of the long-running query, which returns the temperature values stored in the sensors 0 and 1. This query samples the temperature values each second from *"Sep 03, 2012 11:09:34"* to *"Sep 03, 2012 11:09:37"*. Moreover, all the nodes perfoms their local updates and sends the update transactions towards the database server.

Table 6
Summary of the results of the long-running query that queries the temperature sensors in locations 0 and 1.

Long-running query with: idTrans=*2*, initialize date=*"Sep 03,2012 11:09:34"*, period=*1t.u.*, idLocSensor=*0..1*, termination date=*"Sep 03, 2012 11:09:37"*					
idLocSensor	**Returned data (°C)**	**AVI (t.u.)**	**Timestamp**	**Number of local updates (lup)**	**Number of server updates (sup)**
0	25.647	14	Sep 03, 2012 11:09:34	4	1
	25.825	14	Sep 03, 2012 11:09:35		
	25.596	14	Sep 03, 2012 11:09:36		
	25.630	14	Sep 03, 2012 11:09:37		
1	26.088	14	Sep 03, 2012 11:09:34	4	1
	26.024	14	Sep 03, 2012 11:09:35		
	26.109	14	Sep 03, 2012 11:09:36		
	26.094	14	Sep 03, 2012 11:09:37		
2	X	X	X	4	1
3	X	X	X	4	1
4	X	X	X	4	1

Table 7 shows the summary of the energy consumed by each device according to their activities completed.

Table 7
Summary of the energy consumed in the network.

idLocSensor	Consumed energy (%)	Remaining energy (%)
0	1.32325625	98.6767438
1	1.32325625	98.6767438
2	0.92325625	99.0767438
3	0.92325625	99.0767438
4	0.92325625	99.0767438

8. Conclusion and future work

This paper presented a simulation framework for real-time database management on wireless sensor networks. Thus, the description of the different characteristics to design real-time databases on WSN has been done and we proposed a model for a simulation framework of real-time databases on WSN that uses a distributed approach.

The distributed approach was chosen considering its advantages, unlike the warehousing approach. Thus, with the distributed approach the queries could be processed inside the network that diminishes the data flow in the network and increases the lifetime of the network. This is due to the fact that the communication activities use more power than the local process activities. Moreover, by using the distributed approach it is possible to process instant-queries and long-running queries, which are quasi-real-time queries.

In real-time database management on wireless sensor network, the data and transactions have temporal restrictions. For that, the proposed model uses the *Earliest Dead-line First* (*EDF*) protocol to schedule transactions and the *Epsilon Serialisability techniques* to allow conflicting transactions to execute simultaneously in way that their scheduling does not lead an imprecision that is higher than the one accepted in the data. Unlike to usual sensor network simulators oriented network that study the network in the point of view communication or simulators oriented node, which study the functions of a single node, this simulator model emphasizes on the real-time database management techniques for WSNs that use a distributed approach. Thus, this model integrates all the components of a sensor network and the characteristics of real-time database techniques. The full model has been implemented in Java. The object-oriented model was chosen because it is more suitable to model the complex real-world objects with time constraints than the relational model (Kim, 1995). Moreover, the simulator is configurable and the protocols may be changed as required.

A case study shows the execution of real-time transactions and the energy consumed in the network that demonstrates, thus, the validity of the model.

We are currently working on a probabilistic study of real-time transactions with an architecture of real-time database for WSN that uses a distributed approach. This study should allow us to optimize the performance of real-time transaction executions and network resources.

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 a Tecnologia through the PEst-OE/EEI/ LA0008/2011 Project, and by the Ministerio de Ciencia e Innovación, Spain, through the Plan Nacional de I+D+i2008–2011 in the Subprograma de Proyectos de Investigación Fundamental, Project TEC2011-27516.

References

Adelstein F., Gupta S. K. S., Richard G. G. III, Schwiebert L. Fundamentals of mobile and pervasive computing. McGraw-Hill Companies, Inc., New York 2005 (ISBN 0-07-141237-9).

Akyildiz FI, Melodia T, Chowdhury RK. A survey on wireless multimedia sensor networks. Journal of Computer Networks (Elsevier) 2007;51(4):921–60.

Akyildiz FI, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: a survey. Computer Networks (Elsevier) 2002;38:393–422.

Al-Karaki J.N., Kamal A.E. Routing techniques in wireless sensor networks: a survey. IEEE Wireless Communications, Dec. 2004; 11(6), pp. 6-28.

Arasu A., Babcock B., Babu S., Datar M., Ito K., Motwani R., Nishizawa I., Srivastava U., Thomas D., Varma R., Widom

J. STREAM: The Stanford stream data manager. IEEE Data Engineering Bulletin (Citeseer) 2003: 26(1), 19–26.

Baronti P, Pillai P, Chook CVW, Chessa S, Gotta A, Hu FY. Wireless sensor networks: a survey on the state of the art and the 802.15. 4 and ZigBee standards. Journal of Computer Communications (Elsevier) 2007;30(7):1655–95.

Bonnet Ph, Seshadri P. Device database systems. In: Proceedings of the international conference on data engineering ICDE'99. San Diego, CA: Published by the IEEE Computer Society; 2000.

Bonnet Ph, Gehrke J, Seshadri P. Querying the physical world. IEEE personal communications. Special issue ìnetworking the physical world; October 2000.

Bonnet Philippe, Gehrke J, Seshadri P. Towards sensor database systems. In: Proceedings of the second international conference on mobile data management MDM '01. London, UK: Springer-Verlag; 2001. p. 3–14.

Bouazizi E, Sadez B, Duvallet C. Applicabilité du critère d'epsilon-sérialisabilité dans les SGBD temps réel. Special issue ISDM MCSEA; 2004.

Boulis Athanassios. Demo abstract: Castalia: revealing pitfalls in designing distributed algorithms in WSN. In: Proceedings of SenSys '07; November 2007..

Boulis Athanassios. Castalia, a simulator for wireless sensor networks and body area networks. Version 2.0, user's manual; May 2009. [retrieved 27.06.09].

Briand R, Arrijuria O, Dupé V, Richon C, Terrasson G. Evaluation de la consommation énergétique d'un nœud: du hard au soft. RESSACS 2010; 2010.

Callaway E. H. Wireless sensor networks: architectures and protocols. Auerbach Publications (CRC Press), New York 2004 (ISBN: 0849318238).

Chagas Lunas Dantas, Lima Eduardo Pereira, Neto Pedro Fernandes Ribeiro. Real-time databases techniques in wireless sensor networks. In: Proceedings of the sixth international conference on networking and services, IEEE; 2010. p. 182–7.

Chen J, DeWit JD, Tian F, Wang Y. NiagaraCQ: A scalable continuous query system for internet databases. In: Proceedings of the 2000 ACM SIGMOD international conference on management of data. New York: ACM Press; 2000. p. 379–90.

Chen J, Ferreira L. Human motion tracking using wireless sensor networks. Technical report HURRAY-TR-090407. Available from: www.cister.isep.ipp.pt; 2009..

Cho S, Shih E, Ickes I, Min R. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. ACM/IEEE MOBICOM.ACM; 2001. p. 272–287.

Cortes C, Fisher K, Pregibon D, Rogers A. A language for extracting signatures from data streams. In: Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining. New York: ACM Press; 2000. p9–17.

Cranor C, Gao Y, Johnson T, Shkapenyuk V, Spatscheck O. Gigascope: high performance network monitoring with an SQL interface.Proceedings of the 2002 ACM SIGMOD international conference on Management of data. New York: ACM Press; 2002. p. 623.

CrossBow Inc., TelosB (TPR2400) product datasheet. Available from: http://www. willow.co.uk/TelosB_Datasheet.pdf. [retrieved 03.07.13].

Deshpande Amol, Guestrin Carlos, Madden R Samuel, Hellerstein M Joseph, Hong Wei. Model-driven data acquisition in sensor networks. In: Proceedings of the thirtieth international conference on very large data bases. Toronto, Canada; August 31–September 03 2004. p. 588–99.

Diallo O, Rodrigues JJPC, Sene M. Real-time data management on wireless sensor networks: a survey. Journal of Network and Computer Applications (Elsevier) 2011;35:1013–21.

DiPippo L. C., Wolfe V. F. Real-time databases. Chapter in Database Systems Handbook, Multiscience Press (Citeseer), 1997.

Egea-López E, Vale-Alonso J, Martínez-Sala AS, Pavón-Mariño P, García-Haro J. simulation tools for wireless sensor networks. In: Proceedings of summer simulation multiconference—SPECTS 2005; 2005.

Granat J. A framework for event based modeling and analysis. Journal of Telecommunications and Information Technology, 2006; 4: 88–90.

Idoudi N, Duvallet C, Sadeg B, Bouaziz R, Gargouri F. How to model a real-time database? In: Proceedings of the 12th IEEE international symposium on object-oriented real-time distributed computing (IEEE ISORC'2009). Tokyo, Japan: IEEE; 2009a. p. 321–5.

Idoudi N, Louati N, Duvallet C, Bouaziz R, Sadeg B, Gargouri F. A framework to model real-time databases. International Journal of Computing and Information Sciences 2009b;7:1–11.

Karir M, Polley J, Blazakis D, McGee J, Rusk D, Baras JS. ATEMU: a fine-grained sensor network simulator. In Proceedings of the 1st IEEE international conference on sensor ad hoc communication networks; 2004.

Kim W. Object-oriented database systems: promises, reality, and future. Moderne database systems. Addison Wesly; 1995. p. 255–280.

Levis P, Lee N, Welsh M, Culler D. TOSSIM: accurate and scalable simulation of entire TinyOS applications. In: Proceedings of the 1st ACM Conference on embedded networked sensor systems; 2003. p. 126–137.

Li, C., Zhang H., Hao B., Li J. A survey on routing protocols for large-scale wireless sensor networks. Sensors 2011;11(4), pp. 3498–3526.

Lin K., Chen M., Zeadally S., Rodrigues Joel J. P. C. Balancing energy consumption with mobile agents in wireless sensor networks. in Future Generation Computer Systems, (Elsevier), ISSN: 0167-739X, 28(2), February 2012, pp. 446-456, DOI: 10.1016/j.future.2011a.03.001.

Lin K., Rodrigues Joel J. P. C., Ge H., Xiong N., Liang X. Energy efficiency qos assurance routing in wireless multimedia sensor networks. in IEEE Systems Journal, IEEE, ISSN: 1932-8184, Vol. 5, Issue 4, (December 2011b), pp. 495-505, DOI: 10.1109/JSYST.2011.2165599.

Liu C, Leyland J. Scheduling algorithms for multiprogramming in hard real-time environment. Journal of the ACM 1973;20(1):46–61.

Madden S, Franklin MJ, Hellerstein JM, Hong W. TAG: a tiny aggregation service for ad-hoc sensor networks. In: Proceedings of OSDI. Citeseer; 2002.

Madden SR, Franklin MJ, Hellerstein JM, Hong W. TinyDB: an aquisitional query processing system for sensor networks. ACM Transactions on Database Systems (TODS) (ACM) 2005;30(1):122–173.

Mainwaring A, Culler D, Polastre J, Szewczyk R, Anderson J. Wireless sensor networks for habitat monitoring. In: Proceedings of the 1st ACM international workshop on wireless sensor networks and applications. New York: ACM Press; 2002. p. 88–97.

Mendes L. D. P., Rodrigues J. J. P. C. A survey on cross-layer solutions for wireless sensor networks. Journal of Network and Computer Applications (Elsevier), 34(2):523-534, 2010.

Nasreddine N. Conception et modélisation d'un émulateur d'un réseau de capteurs communicant sans fil; 2012.

Neto Pedro Fernandes R, Perkusich Maria LB, Perkusich Angelo. Real-time database for sensor networks. In: Proceedings of the 6th international conference on enterprise information systems; 2004. p. 599–603.

Nguyen AH, Foerster A, Puccinelli D, Giordano S. Sensor node lifetime: an experimental study. In: Proceedings of the 7th IEEE international workshop on sensor networks and systems for pervasive computing; 2011. p. 146–1c51.

Oliveira LMLuís, de Sousa Amaro F, Rodrigues Joel JPC. Routing and Mobility approaches in IPv6 over LoWPAN mesh networks. International Journal of Communication Systems, Wiley1074-5351 2011a;24(11):1445–66, http://dx. doi.org/10.1002/dac.1228.

Oliveira LMLuís, Rodrigues Joel JPC. Wireless sensor networks: a survey on environmental monitoring. Journal of Communications (JCM), Academy Publisher1796-2021 2011b;6(2):143–51, http://dx.doi.org/10.4304/jcm.6.2.143-151.

Pailler S. Analyse hors ligne d'ordonnançabilité d'applications temps réel comportant des tâches conditionnelles et sporadiques 2006:21–30.

Park S, Savvides A, Srivastava MB. Sensorsim: a simulation framework for sensor networks. In: Proceedings of the 3rd ACM international workshop on modeling, analysis and simulation of wireless and mobile systems, ser. MSWIM '00. New York, NY, USA: ACM; 2000. p. 104–111.

Pu C. Generalized transactions processing with epsilon serializability. In: Proceedings of 4th international workshop on high performance transactions systems. Asimolar, California; 1991.

Ramamritham K. Real-time databases. Journal of Distributed and Parallel Databases (Springer) 1993;1(2):199–226.

Ruiz LB, et al. Architectures for wireless sensor networks. In: Proceedings of the 22nd brazilian symposium on computer networks SBRC'04. Gramado, RS, Brazil: [s.n.]; 2004. p. 167–218. Tutorial. [ISBN: 85-88442-82-5] [in Portuguese].

Sacks L., et al. The development of a robust, autonomous sensor network platform for environmental monitoring. Sensors and their applications XXII (Citeseer); 2003.

Silva FA, Braga TRM, Ruiz LB. Nogueira JMS. Tecnologia de Nós Sensores Sem Fio.

Contrôle & Instrumentação 92; Agosto 2004:76–87.

Simon G, lgyesi P Vo, Maroti M, Ledeczi A. Simulation based optimization of communication protocols for large scale wireless sensor networks. Proceedings of IEEE 2003 aerospace conference, vol. 3; 2003. p. 1339–1346.

Sobeih A, Hou JC, Kung L-C, Li N, Zhang H, Chen W-P, Tyan H-Y, Lim H. J-Sim: a simulation and emulation environment for wireless sensor networks. IEEE Wireless Commun ications Magazin 2006:13(4):104–119.

Sundani H., Li H., Devabhaktuni v. K., Alam M., Bhattacharya P. Wireless sensor network simulators a survey and comparisons. International Journal Of Computer Networks (IJCN) 2010; 2(5), pp. 249-265.

Sundresh S, Wooyoung K, Gul A. SENS: a sensor, environment and network simulator. In: Proceedings of the 37th annual simulation symposium; 2004. p. 221–228.

Terrier F, Lanusse A, Bras D, Roux P, Vanuxeem P. Concurrent object for multitasking. L'objet, v3. Paris; 1997. p. 179–196.

TinyOS, an open-source operating system for wireless embedded sensor networks [online]. Available from: <http://www.tinyos.net/>. [retrieved: 01.11.12].

Ulmer C, Yalamanchili S, Alkalai L. Wireless distributed sensor networks for in-situ exploration of mars. Work in progress for NASA technical report (Citeseer); 2003.

Varga A. The omnet++ discrete event simulation system. In: Proceedings of the european simulation multiconference (ESM); June 2001.

Wolfe V. F., Prichard J. J., Dipippo L. C., Black J. Real-time database system: issues and applications. The RTSORAC real-Time-object-oriented dataBase prototype. Kluwer Academic Publlishers; pp 279-301, 1997.

Xian Xiaodong., Shi W, Huang H. Comparison of OMNET++ and other simulator for WSN simulation. In: Proceedings of the 3rd IEEE conference on industrial electronics and applications, ICIEA 2008; 2008. p. 1439–1443.

Zhang Junguo, Li W, Cui D, Zhao X, Yin Z. The NS2-based simulation and research on wireless sensor network route protocol. In: Proceedings of the 5th international conference on wireless communication, networking and mobile computing, WiCom'09; 2009. p. 1–4.

Zhu Y, Shasha D. Statstream: statistical monitoring of thousands of data streams in real time. In: Proceedings of the 28th international conference on very large data bases, VLDB Endowment; 2002. p. 358–369.

Chapter 5

Real-Time Query Processing Optimization for Wireless Sensor Networks

This chapter consists of the following article:

Real-Time Query Processing Optimization for Wireless Sensor Networks

O. Diallo, J. J. Rodrigues, M. Sene, F. Xia

International Journal of Sensor Networks (IJSNet), InderScience Publishers, ISSN (Online): 1748-1287 - ISSN (Print): 1748-1279 (in press).

ISI Impact Factor (2012): 1.2760

Int. J., Vol. x, No. x, 200x

Real-Time Query Processing Optimization for Wireless Sensor Networks

Ousmane Diallo
Instituto de Telecomunicações, University of Beira Interior, Portugal
Department of Mathematics and Informatics, UCAD, Senegal
E-mail: ousmane.diallo@it.ubi.pt

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

Mbaye Sene
Department of Mathematics and Informatics, UCAD, Senegal
Email: mbaye.sene@ucad.sn

Feng Xia
School of Software, Dalian University of Technology, China
Email: f.xia@ieee.org

Abstract

Wireless sensor networks (WSNs) have been the focus of many and high-relevant works. Nowadays, because of time-critical tasks of several applications, one of the new challenges faced in WSNs is to handle real-time data storage and query. Since in such networks energy is the crucial resource, it is challenging to design a query processing mechanism that meets both time constraints and energy. This paper addresses this challenge and proposes a new architecture that combines statistical modelling techniques with the distributed approach and a query processing algorithm to optimize the real-time user query processing. Such statistical model provides good approximate answers to queries with a given probabilistic confidence. This combination allows performing a query processing algorithm based on admission control that uses the error tolerance and the probabilistic confidence interval as admission parameters. The experiments based on real world as well as synthetic data sets demonstrate that the proposed solution optimizes the real-time query processing to save more energy while meeting time constraints.

Keywords: *wireless sensor network; real-time database management techniques; query estimation; query optimization; distribution.*

1. Introduction

Wireless sensor networks (WSNs) may be defined as a set of smart wireless devices, called sensors that are able to sense, process and transmit information about the environment on which they are deployed. These devices are usually distributed in a geographical area, forming thus a network, in order to collect information for users interested in monitoring (e.g., vehicles tracking) and controlling a given phenomenon (e.g., pollution level in a given area). This information is relayed among themselves and to a sink node in order to be accessible by remote users through a gateway. To retrieve the data, users use applications that should provide supports of efficient query techniques, which allow communication with the network [1], [2], [3].

In WSNs, sensor nodes are battery powered and are considered intelligent with acquisitional, processing, storage, and communication capacities. The communication is usually performed by means of radio [4], [5], [6], [7]. However, these resources are generally very limited, especially in terms of storage and energy. The sensor network lifetime depends on the available energy in the nodes composing the network [8], [9]. This available energy is consumed by three activities [4]: sensing activity (data acquisition from the environment), communication (sending and receiving packets), which is essential to form a WSN, and data processing, which consists in some operations applied over data by smart sensors [10], [11]. However, the sensing and processing activities are much less expensive in energy consumption than the wireless communication activities [12]. One of the most used techniques to save power is to activate only necessary nodes and to put other nodes to sleep [13]. Some authors have studied how

Int. J., Vol. x, No. x, 200x

a 3 dimensional sensor field can be efficiently partitioned into cells in order to save energy [14].

Therefore, optimizing sensor activities, mainly the communication activities, as a function of energy is one of the main research topics on this area, and the main driver of this work.

Systems based on sensor networks are getting increasingly used in many areas of knowledge, giving rise to several flavours of WSNs [15], [16]. These different WSNs involved the development of many applications, which are generally connected to databases treating the amount of data collected from sensors. However, the processing time becomes increasingly critical for certain applications. These applications must query and analyze the data in limited time periods in order to make decisions and to react as soon as possible [17]. Some examples of the most popular applications are the following: the control of network traffic [18], transactional analysis (Web, banking or telecommunication transactions) [19], human motion tracking application [20], the tracking of actions on dynamic Web pages [21], [22], monitoring of urban or environmental phenomena [23] [24], and the sensors data management [25].

Once the sensors perform their measurement, the problem of data storing and querying arises. Indeed, the sensors have restricted storage capacity [26] and the on-going interaction between network devices and environment results in huge amounts of data. There are two main approaches to data storage and query in WSNs [27], [28]: the *warehousing approach* and the *distributed approach*. The *warehousing approach* is a centralized system where the sensors act as simply data collectors [29]. The data gathered by sensors are periodically sent to a central database where user queries are processed. This model is the most used one; however, it has some drawbacks, such as eventual wasting resources and bottleneck with an immense amount of transmitted data. Moreover, this approach is unsuitable to real-time processing because it involves time delay for the results. The alternative is the *distributed approach* where each sensor node is considered as a data source, and then the WSN forms a distributed database where the sensed data are form of rows with columns representing sensor attributes [11], [30], [31]. In this second approach, the sensed data are not periodically sent to the database server. They remain in the sensor nodes and some queries are injected in the network through the base station. This approach can offer several advantages such as quasi real-time query processing, support of long-running and instant queries. Furthermore, the distributed approach increases the lifetime of the network by reducing the communications activities among the devices [32].

The data collected by WSNs must closely reflect the current state of the targeted environment. However, the environment changes constantly and the data is collected in discreet moments of time. As such, the collected data has a temporal validity, and as time advances, it becomes less accurate, until it does not reflect the state of the environment any longer [33], [34]. It is thus fundamental that responses to application queries assure that the returned data complies with logical and temporal constraints, and it is in this context that real-time data management on WSNs becomes important [35]. The main goal of real-time data management is to assure temporal consistent data while processing transactions within real- time constraints.

In this context, the design of a query processing mechanism that takes into account both the time-constraints of data and transactions and the energy consumption is fundamentally important due the real-time requirements of data and tasks and the resource limitations of WSNs. Therefore, this paper focuses on optimizing the real-time query processing in WSN architectures for both of these constraints. To reach this goal, the distributed approach is combined with a statistical modelling technique to perform query processing algorithm based on admission control that uses the error tolerance and the probabilistic confidence interval as admission parameters into the system. In real-time systems, for some applications, the accuracy of the results may be sacrificed to reduce the latency [17], [36]. Thus, some applications can tolerate reading stale data under some limits. The main contributions of this paper are the following:

- Proposal of a new architecture and a query processing algorithm to optimize the real-time user query processing for providing both real-time data processing and energy saving. The query processing algorithm, based on the new architecture, uses a statistical model to compute the error and the confidence interval of the sensor data to be accessed in a new concept of virtual network located in the gateway. These error and confidence interval are used for performing query processing based on admission control. Thus, as long as these parameters are accepted by the query, the model can answer the query at the virtual network. Otherwise, the query must be satisfied in the network and the model updated. The error tolerance and confidence interval are controlled by the applications, which define their constraints. Therefore, the query processing is optimized: sensors should be solicited only when the error and confidence interval are not accepted by the query. This method, in addition to a gain in terms of latency and energy consumption, may also improve fault tolerance, because the model can estimate some readings of unreachable sensors for some time.

- Evaluation of the performances of the proposed architecture and query processing algorithm: The experiment results based on real world as well as synthetic data sets demonstrate that this combination and the proposed algorithm optimize the real-time query processing to save more energy while meeting good individual query latency.

The remainder of this paper is organized as follows: Section 2 discusses the related work, while the characteristics of real-time database management systems are exposed in Section 3. In Section 4, the description of the proposed architecture with its underlying query processing algorithm is provided. In Section 5, the performance analysis and the results of the new model for optimizing the real-time user query processing are presented for validation and finally Section 6 concludes the paper.

2. Related work

In the literature, many works were dedicated to effective solutions of transaction processing in WSNs that use the distributed approach. Among these works it may be noted: TinyDB [11], Cougar [31], [37], [38], SNEE [39], [40], [41], [42], and the work in [43]. The TinyDB project [44] was developed for networks where nodes are based on the TinyOS operating system [45]. It is a distributed query processor for sensor networks that incorporate acquisitional techniques. Through an interface, the user chooses what data he wishes to acquire. The query is decomposed by a query processor and distributed across the network. The sensor nodes collect, filter and aggregate the data and respond to the user query. The Cougar project [46] is a platform for distributed query processing. To deal with in-network processing in this platform, they use a clustered approach. A network is composed of several clusters, each of them managed by a cluster head. The child nodes that belong to clusters send their readings according to query needs to concerned cluster head, which then aggregates the received readings and forwards the computed result toward the Front End of the network. This Front End is a query optimizer, located on the gateway node, which generates optimized distributed query processing plans after receiving user queries. Like TinyDB and Cougar, SNEE (Streaming NEtwork Engine) [39] is a complete distributed query processing framework for WSNs. This framework includes principally two components: a compiler/optimizer, named SNEE, and a continuous declarative query language over sensed data streams, named SNEEql [47], [48]. After receiving the SNEEql queries, the compiler/optimizer engine (SNEE) optimizes them by taking into account metadata such as information about the network topology, the required energy and time, the cost of nodes sampling among other relevant parameters. After the optimization phase, the compiler/optimizer engine creates, compiles, and deploys an executable code, the energy-efficient query evaluation plan, which will run into the participating nodes. These proposals define how to query sensor data in a distributed approach and use aggregation techniques to optimize the query processing, but they do not use statistical modelling techniques combined with the distributed approach to optimize the transaction processing. The work in [43] provides real-time database management techniques to deal with the processing of conflict transactions. It uses a distributed approach where the network devices act as part of the database and periodically send the acquired data to the database server. To deal with the time constraints, the devices and the server include a program that processes transaction, while handling the time constraints of the data and transactions. The PostgreSQL DBMS is used to store data provided by sensors and the Query Language for Real-Time Databases (QL-RTDB) [49] is used to access to both the server and the devices via an application interface. To deal with

Int. J., Vol. x, No. x, 200x

the eventual concurrent transactions, an algorithm based on a concurrency control protocol that takes into in consideration both the time constraints and the logic constraints is presented. This algorithm deals very well with the execution of conflict transactions, but it performs in-network processing and periodic updates towards the database server which leads a high-energy consumption in the system. Moreover, in this algorithm the transactions verify the temporal validity of the data to be accessed. If the data is not still valid, then the transaction is waited or aborted. This can lead to performance slowdown because in some situations when one uses temporal constraints directly, one can be making the mistake of performing wait or abort the transactions too many times without having variations that justify these operations.

Works have been done on approximate query answer, among them one can notice: AQUA [50], [51], [52], CONTROL [53], and BBQ [54]. AQUA is a centralized system that uses distinct sampling-based synopses to provide approximate answers for user value queries, hence it cannot profit from the in-network processing advantages [50]. CONTROL uses centralized schemes to provide an interactive method for handling complex long running queries. Usually, a query starts with a large querying and then is continually refined based on feedback with certain confidence bounds [53]. Like AQUA, these centralized schemes do not profit from distributed sensors advantages. BBQ proposed a wireless sensor network declarative query processing framework that uses a probabilistic model to answer queries about the current state of the sensor network. Through SQL query the user chooses what data he wants to acquire. This query is parsed by the query-processor and the model is used to estimate its answer on available sensor data both in spatial and temporal criterion. BBQ uses a time varying multivariate Gaussian model and if the model is not sufficiently fresh to answer the query with acceptable confidence, then the sensor network is solicited to update the data to refine the model [54]. This work in BBQ shares some architectural resemblances with the work in this paper, but the query processing algorithm proposed here defers. In fact, BBQ provides a practical algorithm of optimizing querying execution plan by selecting the sensor readings to acquire for updating the model and afterward uses the model to generate the query answer with the specified confidence. In contrast to BBQ, the query processing algorithm proposed here uses the error tolerance and the confidence interval to route and execute the query inside the network and by the sending the result to the user, the model is updated. Moreover they don't use the concept of virtual network.

Thus, the aim of this work is to complete previous studies by profiting from the distributed architecture and statistical modelling techniques to optimize the real-time query processing.

3. Characteristics of data and transactions in real-time database systems

Like a traditional database management system (DBMS), a real-time DBMS (RT-DBMS) must process transactions and ensures that the logical consistency of the data is not violated. However, unlike a traditional DBMS, a RT-DBMS emphasizes on the temporal validity of the data and the time constraints or deadlines for transactions [33], [34], [55].

The main purpose of a RT-DBMS is to process transactions on time, while maintaining logical and temporal consistency of data. The temporal consistency expresses the need to maintain consistency between the current state of the targeted environment and the state as reflect by the database contents. The temporal consistency can be measured in two ways [56]:

- Absolute consistency, which deals with the need to maintain the view representing the state of the targeted environment consistent with the real state of the environment,

- Relative consistency, which concerns data derived from other ones.

The following paragraph gives a brief overview of the characteristics of data and transactions in real-time databases. For more information, see [55], [56].

3.1 Data

The real-time data represents the capture of the current state of the targeted environment. For example, in the field of medical monitoring, micro sensors can be placed in the human body to control

Int. J., Vol. x, No. x, 200x

blood pressure, breathing, glucose, etc. These data should reflect as closely as the current state of the targeted environment. However, the environment changes constantly and the data are collected in discreet times. So, the collected data have temporal constraints.

To satisfy these temporal constraints, the structure of the data must include these attributes: (i) timestamp, which indicates the instant when the observation relating the data was made; and (ii) absolute validity interval (*avi*) that denotes the time interval following the timestamp during which the data are considered valid. For the data quality, another attribute can be considered; the imprecision or data error (*DE*), which refers to how the current state of the targeted environment may differ from the measured data [56]. The data error on a data version *d* is defined by:

$$DE(d) = 100 * \left|\frac{CurrentValue(d) - UpdateValue(d)}{CurrentValue(d)}\right| \% \qquad (1)$$

1. Transactions

In real-time databases, the transactions are subject to deadlines. A real-time transaction is considered correct if it completes successfully before its deadline. Thus, the transactions can be divided into three categories according to the importance of the deadline: hard (strict and critical), firm (strict and non-critical), and soft (non-strict). In this work the firm transactions are considered (the deadline of the transaction must be met, otherwise the transaction is rejected).

In the literature, it was proposed two types of transaction for the real time DBMS [33], [34]: *update transactions* and *user transactions*. Update transactions are executed periodically to update real-time sensor data, or sporadically to update the derived data in order to reflect the state of the real world. Derived data are the data computed using sensor data. User transactions, representing user queries, arrive aperiodically. They may read or write non real-time data, but only read real-time data. These update transactions also, for comply with temporal constraints must have these following attributes: (i) liberation time that represents the moment on which all the resources for the transaction processing is available; (ii) computing time that indicates the execution time needed for the transaction; and (iii) maximum time, which indicates the maximum time limit for the transaction execution and the periodicity that refers to the frequency with which the transaction happens [43].

To satisfy the logical consistency of the data, transactions must be process with ACID (Atomicity, Consistency, Isolation and Durability) properties. But unlike the conventional databases, in real-time databases these properties are relaxed. Firstly, the atomicity may be relaxed. It is only applied to the sub-transaction that wants to deal with completely data consistency. Secondly, since timeliness is more important than correctness, in many situations, correctness can be traded for timeliness. Thirdly, the isolation allows transactions to communicate with others to better perform control functions. Finally, in real-time databases, not all data must be permanent and some of them are temporal [43], [55], [56].

4. System Architecture

It is assumed that the topology of the physical network is stable and well known. Moreover, it is assumed that the interacting applications tolerate reading inconsistent data under some limits. This relaxation is herein used to improve real-time transaction processing. For instance, in [43], an algorithm that uses tolerated inconsistent data to deal with conflict real-time transactions is proposed.

To satisfy the user query processing in real-time, the proposed approach combines the distributed architecture and statistical modeling techniques. In the distributed architecture, the *gateway* is considered to include more functions and it often integrates, among other components, an *admission controller*, a *scheduler*, a *transactions trigger*, and a *virtual network*. This work distinguishes two kinds of transactions: update transactions, which update the real-time data in the virtual network on demand; and the user transactions, which query the current state of the targeted environment.

The goal of this work is both to process very quickly user queries in real-time and to reduce the communication cost and processing load of queries in the sensor network for a gain of energy according to the requirements (data freshness) of

Int. J., Vol. x, No. x, 200x

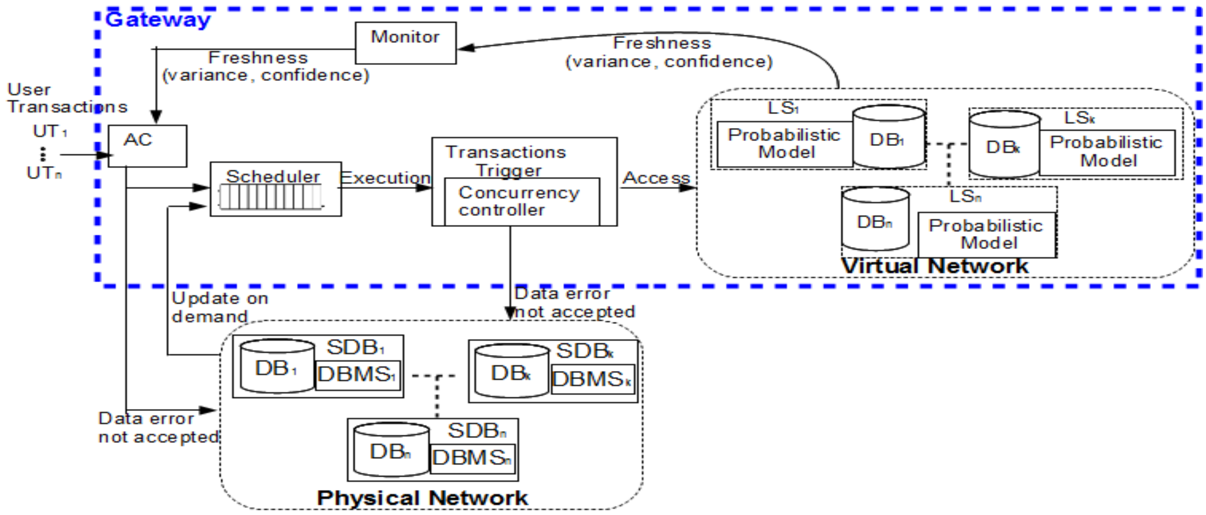


Figure 1: Proposed system architecture.

the applications. The proposed architecture is shown in the figure 1. In this architecture, the admission controller uses the variance (error) with respect to time and the confidence interval of the sensor readings to route the user query either in the virtual network, where it will be answered by a probabilistic model, or in the physical network considered more fresh. Therefore, the query processing will be optimized for both latency and energy consumption.

The description of the whole architecture is made below. Moreover, it will be described clearly how this architecture could be used to optimize the user queries by performing a query processing algorithm based on admission control that uses the error tolerance and the confidence interval as the admission parameters into the system.

4.1 Physical Network

The *physical network* is composed by real physical sensors (the case study in this work uses temperature sensors), which run directly in the targeted environment. These sensors are able to sense, process queries, and transmit information about the environment on which they are deployed. All these actions are done according to time constraints. Therefore, the sensors act as local databases named sensor databases *(SDB)*, incorporating a software component that manages the data storage and user queries with time constraints. Thus, the WSN forms a distributed database. These sensors should be solicited only when the model is not able to satisfy a query with their accepted *error* with respect to time and *confidence interval*. This will greatly reduces the activities of the sensors, increasing thus the network lifetime. Moreover, this network can be organized with an efficient protocol for enhancing the network lifetime further.

4.2 Gateway

The *gateway* is usually used to connect the wireless sensor network and the external networks (e.g. IP). It has the same processing and storage capabilities as a normal PC. In the proposed architecture, the gateway includes, among other components, an *admission controller*, a *scheduler*, a *transactions trigger*, and a *virtual network*.

4.2.1 Admission controller (AC)

The *admission controller (AC)* has the task of controlling and routing user transactions that are accepted or not in the system. It performs this control and this routing according to the freshness (variance and confidence interval) calculated on real data in the model. Its functioning is controlled by the *monitor,* which inspects the calculations in the model and provides it with its operating parameters. Thus, the admission controller intercepts the user transactions, controls and routes them according to the freshness (and possibly other parameters) either in the virtual network or in the physical network.

Int. J., Vol. x, No. x, 200x

4.2.2 Transactions trigger

The user transactions admitted in the system and the update transactions from the sensors are placed in a *waiting queue* before being sent to the *transaction trigger*. This *transactions trigger* has the function of managing the execution of transactions. Thus, it has the *concurrency controller* module that manages the concurrent transactions as follows: when a user transaction wishes to access a data currently in update, then the data error (variance with respect to time) is recomputed with the new sensor data in the current update transaction. If the new error is accepted by the user transaction, then it can be retrieved from the virtual network. Otherwise, the user transaction is sent to the physical network considered fresher. On the other hand, if an update transaction wishes to update data currently in use by another transaction, and if the second transaction is an update, then the new data is simply inserted in the database to refine and make the model stringer. Otherwise, if the second transaction is a user transaction, then the error is recomputed. If this error is still accepted by the user transaction, then the update transaction can run concurrently. Otherwise, it is queued.

4.2.3 Virtual Network

The *virtual network* is composed by *logical sensors (LS)* and has the same topology with the physical network. Thus, each sensor database has a logical sensor as its correspondence in the virtual network. A logical sensor is a simple application module composed by a *probabilistic model* and a *memory (DB)*. Each memory of a logical sensor stores the historical data of the corresponding real sensor in the physical network. The probabilistic model of each logical sensor uses the data stored in its memory to estimate the answer of the real-time user queries according to their required data freshness constraints. These answers should represent the current state of the targeted environment, and they have temporal validity. As time advances the model becomes less accurate, until the time where it cannot answer some queries with acceptable error and confidence interval. When that happens the physical network should be solicited to update the sensor data in the memory of the targeted logical sensors.

4.2.3.1 Probabilistic Model

While the proposed architecture is general, the probabilistic model used depends on the physical phenomenon studied. The case study in this work focuses on the monitoring of the temperature of an environment. Therefore, this work focuses, without loss of generality and for exemplification purposes, on a model based on the Gaussian distribution [57], [58].

A random variable *X* follows a normal distribution with parameters μ and σ, denoted $N(\mu, \sigma)$, if its probability density equation is:

$$P(X) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{X-\mu}{\sigma})^2} \qquad (2)$$

whose graph is the famous *"bell curve"*, and the two parameters μ and σ respectively represent the mean and standard deviation of *X*. The probability distribution is symmetrical with respect to μ, which therefore characterizes the central tendency, and σ is the dispersion of the distribution. The larger it is, the more the distribution is spread on both sides of μ. The inflection points are located $(\mu - \sigma)$ and $(\mu + \sigma)$, with:

$$Prob\{|X - \mu| < \sigma\} = 68.26\% \qquad (3),$$
$$Prob\{|X - \mu| > 1.96\sigma\} = 5\% \qquad (4),$$
$$Prob\{|X - \mu| > 2.58\sigma\} = 1\% \qquad (5)$$

It is known that the probability density function (*pdf*) representing the variation of temperatures follows a Gaussian distribution. This property was verified by numerous works, including the one proposed in [54]. This work ensured that was true for the case study by making several tests on several real sensor data sets. Thus for exemplification purposes, the curve in figure 2 shows a *pdf* of a set of real measures of temperature with a mean μ=*23.9*.

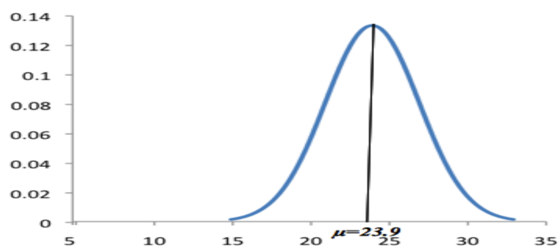


Figure 2: Example of a Gaussian curve with 23.9 of mean.

Int. J., Vol. x, No. x, 200x

This proposal consists of optimizing the user queries by combining a distributed sensor architecture and a probabilistic model. Let us consider that the model is the probability density function *(pdf)* with equation (2), *p(X)*, assigning a probability for each possible assignment to the random variable *X*, representing the variation of temperatures of a given environment. For building the representation of this *pdf* of the variation of temperatures, the proposal uses the data stored in the database of each logical sensor. Thus after construction of the representation of the *pdf*, the probabilistic model is used to estimate the answer of user queries about the current state of sensor readings according to the data freshness. When a user query arrives with accepted error and confidence interval that cannot be satisfy by the model, it is executed in the physical network and the model is updated on demand.

4.3 Query processing approach

The probabilistic model used in this work can answer many complex query types. However for exemplification purposes, this section focuses on attribute-value estimates that are used in the case study.

4.3.1 Query estimation

The objective of this query estimation is to estimate the answer of real-time user queries based on statistics. The user can query the information in one sensor or in a set of sensors located in a geographic area. For exemplification purposes, consider that a user query arrives in the system at time *t* and asks to estimate the values of temperature of the sensor database SDB_i with an error ε and a confidence $1-\theta$ ($\theta \in [0,1]$). Suppose that the readings in the memory DB_i of the logical sensor LS_i are given by the table I (the data and their corresponding timestamps are only represented here, but in reality the data model encapsulates other attributes, as explained in section 3.1) and the last sensor reading represents the last value in the table.

Table I: Representation of the data and their timestamps in the logical sensor LS_i

	Logical sensor LS_i				
Data	d_0	d_1	… d_k	…	d_N
Timestamps	t_0	t_1	… t_k	…	t_N

The late time of the query from the last reading is given by:

$$Q = t - t_N \quad (6)$$

By using the previous *pdf*, one can compute the expected answer of the query, the mean μ and the standard deviation σ with:

$$\mu = E(X) = \int_{-\infty}^{+\infty} x p(x) dx \quad (7)$$

$$\sigma^2(X) = E[(X-\mu)^2] = \int_{-\infty}^{+\infty} (x-\mu)^2 p(x) dx \quad (8)$$

Thus, the average variance per time unit is given by:

$$\sigma^2{}_{/t} = \frac{1}{\sum_{i=0}^{N} t_i} \sigma^2 \quad (9)$$

and the exact variance of the temperature between the last reading and the arrival of the query is given by:

$$\Delta = \frac{1}{\sum_{i=0}^{N} t_i} Q \sigma^2 \quad (10)$$

Thus, this calculation will help to take care of the temporal validity of the real-time data, which is intrinsically given by the error one is willing to take. After these computations, the model can answer the user query if these two conditions are met:

1. $\Delta \le \varepsilon$

2. $P(X \in [\mu-\sigma, \mu+\sigma]) \ge 1-\theta$,
With $P(X \in [\mu-\sigma, \mu+\sigma]) = \int_{\mu-\sigma}^{\mu+\sigma} p(x) dx$ (11)

If one of these conditions is not true, then the model is not sufficiently rich to answer the query. Below, the query processing algorithm is discussed to better explain how the user query and the updates are processed.

4.3.2 Query processing algorithm

Notice that a user query requests real-time information about the targeted environment. Moreover, a query is a triplet of objects constituted by the request (the targeted sensors), the tolerated error, and the targeted confidence interval. Thus, when a user query arrives, it is intercepted by the

Int. J., Vol. x, No. x, 200x

admission controller, which compares the freshness requirement of the query according to the freshness of the data in the model (calculations in previous section):

1. If the tolerance for the freshness verifies, it means that the model is sufficiently fresh to answer the query, the query is estimated in the virtual network and the results sent directly to the user.

2. If the tolerated freshness does not verify, then the query is executed in the wireless sensor network (which is supposedly more fresh), and while sending the results to the user, the model is also updated in order to refine the estimates. The algorithm of a user query processing with 95% confidence (see equation 4.2.3.1) is given below:

```
CT = GetCurrentTime() + processing_time;
TL = GetTimeOfLastMeasure(query);
VAR = GetVarianceForTimePeriod(CT,query);
if( VAR < =given_value ){
        LM = GetLastMeasure(query);
        AV = GetAvgForTimePeriod(query);
        return "last measure: " + LM + " in time: " + TL +
        "Average    for this time period: " + AV + "
        estimated error (95% confidence): " + 2*VAR;
    }else{
        go to WSN;
        return results;
        storeNewDataAsHistoricalData(values);
        updateStatistics();
          }
```

Figure 3: Query processing algorithm.

5. Performance analysis

The goal of this section is to compare performance indices of the proposed architecture and its query mechanism against the ones used in [43] named here for identification purpose *tested model*. To reach this goal, the work in [43] is implemented. Indeed, this work presents an efficient solution based on the distributed approach to deal with the processing of conflicts between real-time transactions on WSNs. The eventual conflicts of transactions are handled by an algorithm based on a relaxed concurrency control protocol which takes into consideration both the time constraints and the logic constraints. In this model, the query is intended either for sensors to perform in-network processing or for the central database server and the updates are periodically sent to the central repository, this leads a high-energy consumption in the system. Moreover, in this algorithm the transactions verify the temporal validity of the data to be accessed. If the data is not still valid, then the transaction is waited or aborted. This leads to performance slowdown when one knows that some applications tolerate reading stale data under some limits. In addition, when one uses temporal constraints directly, one can make the mistake of performing wait or abort the transactions too many times without having variations that justify these operations.

The work proposed in this paper does not deal with conflicts between transactions, but provides a solution to improve the processing of real-time user transactions in terms of latency and energy wasted according to user specified freshness.

The type of query examined in this work is the attribute value estimates. Thus, the user can query the information in one sensor or in a set of sensors located in a geographic area, within a specified error ε with a confidence interval $1\text{-}\theta$ ($\theta \in [0,1]$).

The performance indices, which are in interest in this work, are the energy cost and the latency of the user query processing. For exemplification purpose, the energy cost is determined according to the nodes visited by the query, the data sensing, processing, transmission, and the sensor's duty cycle operation. Thus, the formula 12 below gives the energy consumption of each sensor node. For more details about the calculation of each term of the formula, readers can refer to [59].

$$E_{node} = E_{sensor} + E_{ADC} + E_{\mu c} + E_{transceiver} \quad (12)$$

To capture the actual energy consumption of each sensor's activity in the simulation, each activity need to be calibrated by real measures. For that, the cost model used numbers obtained from the TelosB datasheet [60], [61], as presented in Table II.

Table II: Nominal power consumption of TelosB nodes [61]

Components	Mode	Current draw
Module	Active	1.8 mA
	Sleep	5.1 μA
	Receive	19.7 mA
RF-Transceiver	Transmit(at 0dBm)	17.4 mA
	Sleep	0.01 mA

Int. J., Vol. x, No. x, 200x

To calculate the latency of each query let us assume that N queries are sent to the system and each query Q_i has an instant time A_i that it was sent to the system and an instant time F_i that it finishes to be processed. Thus, the latency of each query is given by:

$$\bar{R} = \frac{1}{N}\sum_{i=1}^{N}(F_i - A_i) \qquad (13)$$

This work is interested in the variation of the energy cost in the network and the latency for the treatment of the real-time user queries according to the consistency and to the workload of the system. To obtain these performance criteria, the tolerated error is modified. Moreover, the nodes in the model [43] are programmed to periodically send their readings to update the real-time data in the central database located inside the gateway and the same number of query is submitted to the two models, this will represents the charge of the system.

5.1 Results Analysis

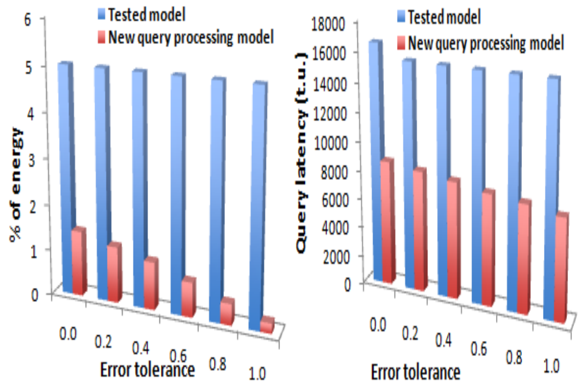
This section presents the simulation results of the proposed architecture and query mechanism to optimize the real-time user query processing. These experiments are done based on real world as well as synthetic data sets.

5.1.1 Energy cost and latency *versus* freshness relaxation

To measure the impact of relaxing freshness in the energy consumption and the query latency, the user queries are sent in a network composed by ten sensors. The queries ask the temperature readings from the sensor within a specified tolerated error and a confidence interval. In these experiments the tolerated error (ε) varies between *0* and *1* degrees Celsius and the confidence interval is fixed to *95%.*

A. A case where user queries are executed in the central database for the tested model.

In the tested model, queries are intended either for sensors or for the central database server what justify these first experiments where the queries are sent to the central database in their model. The results obtained are shown in the figure 4.



(a) (b)

Figure 4: Percentage of energy per query (a) and individual query latency (b) *versus* error tolerance with a confidence interval of 95% (queries sent to the database for the tested model).

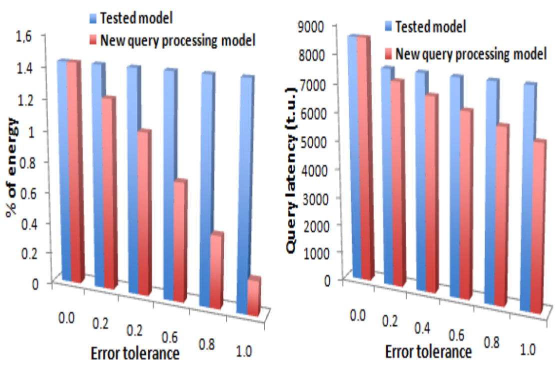
As expected, when the tolerated error (ε) increases, the percentage of energy consumed in the tested model remains big. This is due to the fact that the nodes are programmed to periodically send all their readings to the central database independently to the relaxation, what involves a huge energy consumption brought mainly by communication operations. However, the percentage of energy consumption of the proposed approximate processing remains very smaller and decreases gradually. This can be explained by the fact that in this new model the communication cost decreases gradually with respect to the relaxation; most of the queries is estimated at the virtual network what not involves much data transmission from sensors to the gateway, see figure 4 (a). The central database will be updated only if the freshness of the actual data in the virtual network is note accepted by the query. Concerning the query latency in time unit (t.u.), depicted by the figure 4 (b), as ε increases it decreases slowly in the tested model because its coherency control protocol is also based on a relaxation mechanism. However, the individual query latency in the new proposed model is much better. This is because in the new proposed model, query results are approximated. In addition, if the data freshness in the virtual network is accepted for approximating the results, the query is not put in wait mode; it is executed directly in the physical network. Instead, in the tested model if the data are outdated, the queries are put in wait mode

Int. J., Vol. x, No. x, 200x

for updates. This leads to a huge waste of time without even it has a substantial variation in the physical network.

B. A case where user queries are executed in the network for the tested model.

This section deals with the case where in the tested model the queries are sent in the network to collect information, while in the new proposed model the queries are estimated in the virtual network before to be sent in the network when the virtual network is not sufficiently fresh. The results are shown in the figure 5.



(a) (b)

Figure 5: Percentage of energy per query (a) and individual query latency (b) *versus* error tolerance with a confidence interval of 95%(queries sent to the network for the tested model).

One can notice that when ε increases, the percentage of energy consumption in the tested model remains constant. This is due by the fact that all the queries are running in the network and independently to the relaxation this model keeps reporting the sensor readings, leading thus an extra communication cost. As for the percentage of energy consumption of the new proposed model, it decreases gradually. Furthermore, let us notice that when ε is equal to 0; i.e. the query asks the data with no error tolerance, the percentage of energy consumption of the proposed model gets the same value to that of the tested model. This phenomenon is normal because in that case all the queries in the proposed model are sent to the network, so the same situation in the tested model, see figure 5 (a). Concerning the individual query latency (in t.u.), when the queries ask the data with no error tolerance, the latency is more or less the same in the two models because in this case the queries have the same execution plan in both models. Furthermore, when the tolerated error increases the query latency in the two models decreases gradually, but it is better in the new proposed model, see figure 5 (b).

6. Conclusion and Future Work

This work proposed a new architecture and a query-processing algorithm to optimize the real-time user query processing for WSNs. Instead of periodically send the sensor readings to the database server for off-line processing or process the query directly into the network, the proposed approach combines a distributed architecture and statistical modeling techniques to perform a query processing based on admission control that uses the error tolerance and the confidence interval as the admission parameters. A new concept of virtual network, composed by logical sensors which, in their turn, are composed by a probabilistic model and memory, is used to approximate the answer of the query according to a given error tolerance and confidence interval. If the sensor data inside the virtual network is not sufficiently rich to answer the query, the admission controller routes the query towards the physical network. The experimental results based on real world as well as synthetic data sets show that the general proposed architecture provides, among other advantages, good individual query latency and valid data for real-time applications and energy-efficiency for WSNs.

Correlation is an important aspect between sensors and their neighborhood. Therefore in future works, the improvement of the proposed architecture and its query mechanism in order to take into account the correlation between the sensor readings and eventually other parameters will be investigated. This will further improve the estimate of readings and the power saving.

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 a Tecnologia* through the PEst-OE/EEI/LA0008/2013 Project.

REFERENCES

[1] E. H. Callaway, *"Wireless Sensor Networks: Architectures and protocols "* CRC Press, 2004.

[2] P. Baronti, P. Pillai, C: V. Chook, S. Chessa, A. Gotta, and F. Y. Hu, "*Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards* " Journal of Computer communications, 30 (7), 1655–1695, 2007.

[3] L. M. L. Oliveira and J. J. P. C. Rodrigues, "*Wireless Sensor Networks: a Survey on Environmental Monitoring*," Journal of Communications (JCM), Vol 6, No 2, pp. 143-151, April 2011, DOI: 10.4304/jcm.6.2.143-151

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "*Wireless sensor networks: a survey*," Computer networks, 38, 393-422, 2002.

[5] A. Nayak, I. Stojmenovic, "*Wireless sensor and actuator networks: algorithms and protocols for scalable coordination and data communication",* Wiley-Interscience, 2010.

[6] Akyildiz, F. I., T. Melodia, and R. K. Chowdhury. "*A survey on wireless multimedia sensor networks"*. Journal of Computer networks (Elsevier) 51, no. 4 (March 2007): 921--960.

[7] Oliveira L.M. Luís, Amaro F. de Sousa, and Joel J. P. C. Rodrigues. "*Routing and Mobility Approaches in IPv6 over LoWPAN Mesh Networks"*.in International Journal of Communication Systems, Wiley, ISSN: 1074-5351, Vol. 24, Issue 11, November 2011a, pp. 1445-1466, DOI: 10.1002/dac.1228.

[8] K. Lin, M. Chen, S. Zeadally, and J. J. P. C. Rodrigues, " *Balancing Energy Consumption with Mobile Agents in Wireless Sensor Networks "* Future Generation Computer Systems, Vol. 28, Issue 2, pp. 446-456, Feb. 2012, DOI: 10.1016/j.future.2011.03.001,

[9] K. Lin, J. J. P. C. Rodrigues, H. Ge, N. Xiong, and X. Lian "*Energy Efficiency QoS Assurance Routing in Wireless Multimedia Sensor Networks "* IEEE Systems Journal, Vol. 5, Issue 4, pp. 495-505, December 2011, DOI: 10.1109/JSYST.2011.2165599

[10] Cho, S., E. Shih, N. Ickes, and R. Min. "*Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks"*. ACM/IEEE MOBICOM. ACM, 2001. 272– 287.

[11] Madden, S. R., M. J. Franklin, J. M. Hellerstein, and W. Hong. "*TinyDB: An Aquisitional Query Processing System for Sensor Networks"*. ACM Transactions on Database Systems (TODS) (ACM) 30, no. 1 (2005): 122-173.

[12] Madden, S., M. J. Franklin, J. M. Hellerstein, and W. Hong. "*TAG:Atiny aggregation service for ad-hoc sensor networks"*. In OSDI.Citeseer, 2002.

[13] Sandra Sendra, Jaime Lloret, Miguel Garcia and Jose F. Toledo, "*Power saving and energy optimization techniques for Wireless Sensor Networks"*, Journal of Communications, Vol. 6 Issue 6, Pp. 439-459. August 2011. doi:10.4304/jcm.6.6.439-459

[14] Susumu Matsumae, "*Energy-Efficient Cell Partition of 3D Space for Sensor Networks with Location Information"*, Network Protocols and Algorithms, Vol 1, No 2 (2009), Pp. 85-98

[15] P. A. Neves, J. J. P. C. Rodrigues, M. Chen, and A. V Vasilakos "*A Multi- channel Architecture for IPv6-enabled Wireless Sensor and Actuator Networks Featuring PnP Support* " Journal of Network and Computer Applications, 2011, (in press), DOI: 10.1016/j.jnca.2011.03.033

[16] L. M. L. Oliveira, J. J. P. C. Rodrigues, A. F. de Sousa, and J. Lloret, "*Network Access Control Framework for 6LoWPAN Networks*" Sensors, Vol. 13, Issue 1, pp. 1210-1230, 2013. DOI: 10.3390/s130101210.

[17] O. Diallo, J. J. Rodrigues, and M. Sene, "*Real-time data management on wireless sensor networks: A survey* " Journal of Network and Computer Applications, 35 (3), pp. 1013-1021, May 2012.

[18] Cranor C, Gao Y, Johnson T, Shkapenyuk V, Spatscheck O. "*Gigascope: High performance network monitoring with an SQL interface"*.Proceedings of the 2002 ACM SIGMOD international conference on Management of data. New York: ACM Press; 2002. p. 623.

[19] Cortes C, Fisher K, Pregibon D, Rogers A. "*Hancock: a language for extracting signatures from data streams"*. Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. New York: ACM Press; 2000. p. 9–17.

[20] Chen J, Ferreira L. "*Human Motion Tracking using Wireless Sensor Networks.*" Technical Report HURRAY-TR-090407, available online from /www.cister.isep.ipp.ptS, 2009.

[21] Zhu Yunyue, Shasha Dennis. "*Statstream: Statistical monitoring of thousands of data streams in real time*". Proceedings of the 28th international conference on Very Large Data Bases.VLDB Endowment, 2002. p. 358–69.

[22] Chen J, DeWitt David J, Tian F, Wang Y. "*NiagaraCQ: A scalable continuous query system for internet databases"*. Proceedings of the 2000 ACM SIGMOD international conference on Management of data. New York: ACM Press; 2000. p. 379–90.

[23] Mainwaring A, Culler D, Polastre J, Szewczyk R, Anderson J. "*Wireless sensor networks for habitat monitoring"*. Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications. New York: ACM Press; 2002. p. 88–97.

[24] Ulmer C, Yalamanchili S, Alkalai L. "*Wireless distributed sensor networks for in-situ exploration of mars"*. Work in Progress for NASA Technical Report (Citeseer), 2003.

[25] Arasu A, Babcock B, Babu S, Datar M, Ito K, Motwani R, et al. "*STREAM: The Stanford Stream Data Manager"*. IEEE Data Engineering Bulletin. Citeseer 2003:19–26.

[26] Silva FA, Braga TRM, Ruiz LB, Nogueira JMS. "*Tecnologia de Nós Sensores Sem Fio"*. Contrôle & Instrumentação; agosto 2004;92:76–87.

[27] Bonnet Ph., Seshadri P. "*Device Database Systems"*. In: Proceedings of the International Conference on Data Engineering ICDE'99. San Diego, CA: IEEE Computer Society; 2000.

[28] Bonnet Ph, Gehrke J, Seshadri P. "*Querying the Physical World"*. Journal of Personal Communications, IEEE 2000;7(5):10–5.

[29] Servigne S., Noel G, "*Real time and spatiotemporal data indexing for sensor based databases*", *Journal of Geospatial Information Technology for Emergency Response* (Taylor & Francis Group) 2008 ; 6: 123-142.

[30] O. Diallo, J. J. P. C. Rodrigues, M. Sene, J. Lloret, "*Distributed Database Management Techniques for Wireless Sensor Networks*", *IEEE Transactions on Parallel and Distributed Systems, 19 Aug. 2013. IEEE Computer Society Digital Library. IEEE Computer Society, <http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.207>*

[31] Y. Yao and J. Gehrke, "*Query Processing in Sensor Networks*," In First Biennial Conference on Innovative Data Systems Research (CIDR 2003), Asilomar, California, January 5-8, 2003.

[32] M. Demirbas, X. Lu, and P. Singla, "*An in-network querying framework for wireless sensor networks*," IEEE Trans. on Parallel and Distributed Systems (TPDS), vol. 20, no. 8, pp., 1202-1215, 2009.

[33] Idoudi Nizar, Duvallet Claude, Sadeg Bruno, Bouaziz Rafik, Gargouri Faiez. "*How to model a real-time database?"*. Proceedings of 12th IEEE International Symposium on Object-oriented Real-time distributed

Computing (IEEE ISORC'2009). Tokyo, Japan: IEEE, 2009a. p. 321–5.
[34] Idoudi Nizar, Louati Nada, Duvallet Claude, Bouaziz Rafik, Sadeg Bruno, Faiez Gargouri A. "*Framework to Model Real-Time Databases*". International Journal of Computing & Information Sciences 2009b;7:1–11.
[35] O. Diallo, J. J. P. C. Rodrigues, M. Sene, J. Lloret. Simulation framework for real-time database on WSNs. Journal of Network and Computer Applications (2013), http://dx.doi.org/10.1016/j.jnca.2013.07.001
[36] Pailler, S. "*Analyse Hors Ligne d'Ordonnançabilité d' Applications Temps Réel comportant des Tâches Conditionnelles et Sporadiques*". 2006, 21-30.
[37] P. Bonnet, J. Gehrke, and P. Seshadri, "*Towards sensor database systems*," 2nd International Conference on Mobile Data Management, Hong Kong, China, January 8-10, 2001.
[38] Y. Yao and J. Gehrke, "*The cougar approach to in-network query processing in sensor network*," ACM SIGMOD Record , 31 (3), 9- 18, 2002.
[39] I. Galpin, C. Y. Brenninkmeijer, A. J. Gray, F. Jabeen, A. A. Fernandes, N. W. Paton, "*SNEE: a query processor for wireless sensor networks*," Distributed and Parallel Databases, 29, 31–85, 2011.
[40] I. Galpin, C. Y. Brenninkmeijer, A. J. Gray, F. Jabeen, A. A. Fernandes, N. W. Paton, "*An architecture for query optimization in sensor networks*," IEEE 24th International Conference on Data Engineering (ICDE 2008), Cancun, Mexico, 7-12 April 2008. Pp. 1439 - 1441. DOI: 10.1109/ICDE.2008.4497582
[41] I. Galpin, C. Y. Brenninkmeijer, F. Jabeen, A. A. Fernandes, and N. W. Paton, "*Comprehensive optimization of declarative sensor network queries*," 21st International Conference on Scientific and Statistical Database Management (SSDBM 2009), Lecture Notes in Computer Science, Vol. 5566. Pages 339-360.
[42] G. Valkanas, D. Gunopulos, I. Galpin, A. J. G. Gray, and A. A. A. Fernandes, "*Extending query languages for in-network query processing*," 10th ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE '11), Athens, Greece, June 2011. pp. 34-41.
[43] Chagas Lunas Dantas, Lima Eduardo Pereira, Ribeiro Neto Pedro Fernandes. "*Real-Time Databases Techniques in Wireless Sensor Networks*".Sixth International Conference on Networking and Services.IEEE, 2010.p. 182–7.
[44] S. Madden, W. Hong, J. Hellerstein, and K. Stanek, "*TinyDB: A Declarative Database for Sensor Networks*," accessed 25/01/2013 Available: http://telegraph.cs.berkeley.edu/tinydb/overview.html
[45] Berkeley. *TinyOS*. Available: http://www.tinyos.net/, 1999.
[46] P. Bonnet, J. Gehrke, and P. Seshadri, "*Cougar: The Sensor Network is the Database*," accessed 25/01/2013 Available:www.cs.cornelle.edu/database/cougar/, 2001.
[47] C. Y. Brenninkmeijer, I. Galpin, A. A. Fernandes, and N. W. Paton, "*A semantics for a query language over sensors, streams and relations*," 25th British National Conference on Databases, BNCOD 25, Cardiff, UK, July 7-10, 2008. Lecture Notes in Computer Science Vol. 5071, Pp. 87–99.
[48] C. Y. Brenninkmeijer, I. Galpin, A. A. Fernandes, and N. W. Paton, "*Validated cost models for sensor network queries*," Sixth International Workshop on Data Management for Sensor Networks (DMSN '09), Lyon, FRANCE, August 24, 2009.
[49] Leite CRM. "*Linguagem de Consulta para Aplicações em Tempo-real*".Dissertação, Programa de Pós-Graduação em Engenharia Eléctrica. Campina Grande: Universidade Federal de Campina Grande; 2005.
[50] P. B. Gibbons. "*Distinct sampling for highly-accurate answers to distinct values queries and event reports*". In Proc. of VLDB, Sept 2001.
[51] P. B. Gibbons and M. Garofalakis. "*Approximate query processing: Taming the terabytes* (tutorial), September 2001.
[52] P. B. Gibbons and Y. Matias. "*New sampling-based summary statistics for improving approximate query answers*". In SIGMOD, 1998.
[53] J. M. Hellerstein, R. Avnur, A. Chou, C. Hidber, C. Olston, V. Raman, T. Roth, and P. J. Haas. "*Interactive data analysis with CONTROL*". IEEE Computer, 32(8), August 1999.
[54] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. "*Model-Driven Data Acquisition in Sensor Networks*". In VLDB Conference, September 2004.
[55] DiPippo LC, Wolfe VF. "*Real-time databases*".Journal of Database Systems Hand- book, Multiscience Press (Citeseer) 1997.
[56] Ramamritham K. "*Real-time databases*". Journal of Distributed and Parallel Data- bases (Springer) 1993;1(2):199–226.
[57] Chris Belsom. "*The Normal distribution*".Cambridge University Press, 1992.
[58] Wlodzimierz Bryc. "*The normal distribution: characterizations with applications*". Springer-Verlag, 1995.
[59] R. Briand, O. Arrijuria, V. Dupé, C. Richon, G. Terrasson , "*Evaluation de la consummation énergétique d'un nœud: Du hard au soft*", *RESSACS 2010*. 2010.
[60] CrossBowInc, TelosB (TPR2400) Product Datasheet, available online from *http://www.willow.co.uk/TelosB_Datasheet.pdf.* Retrieved: 03/07/2013.
[61] A. H. Nguyen, A. Foerster, D. Puccinelli, and S. Giordano, "*Sensor node lifetime: an experimental study*", in 7th IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing, pp. 146-151, 2011.

Chapter 6

Real-Time Query Processing Optimization for Cloud-based Wireless Body Area Networks

This chapter consists of the following article:

Real-Time Query Processing Optimization for Cloud-based Wireless Body Area Networks

O. Diallo, J. J. Rodrigues, M. Sene, N. Jianwei

Information Sciences, Elsevier, ISSN: 0020-0255, doi: http://dx.doi.org/10.1016/j.ins. 2014.03.081 (in press).

ISI Impact Factor (2012): 3.643

ISI 5-Years Impact Factor (2012): 3.676

Real-Time Query Processing Optimization for Cloud-based Wireless Body Area Networks

Ousmane Diallo
Instituto de Telecomunicações, University of Beira Interior, Portugal
Department of Mathematics and Informatics, UCAD, Senegal
E-mail: ousmane.diallo@it.ubi.pt

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

Mbaye Sene
Department of Mathematics and Informatics, UCAD, Senegal
Email: mbaye.sene@ucad.sn

Jianwei Niu
Beihang University, Beijing, China
Email: niujianwei@buaa.edu.cn

Abstract

Wireless body area networks (WBANs) have received a lot of attention from both academia and industry due to the increasing need of ubiquitous computing for eHealth applications, the continuous advances in miniaturization of electronic devices, and the ultra-low-power wireless technologies. In these networks, various sensors are attached either on clothes, on human body or even implanted under the skin for real-time health monitoring of patients in order to improve their independent daily lives. The energy constraints of sensors, the vital and large amount of data collected by WBAN nodes require powerful and secure storage, and a query processing mechanism that takes into account both real-time and energy constraints. This paper addresses these challenges and proposes a new architecture that combines a cloud-based WBANs with statistical modeling techniques in order to provide a secure storage infrastructure and optimize the real-time user query processing in terms of energy minimization and query latency. Such statistical model provides good approximate answers to queries with a given probabilistic confidence. Furthermore, the combination of the model with the cloud-based WBAN allows performing a query processing algorithm that uses the error tolerance and the probabilistic confidence interval as query execution criterions. The performance analysis and the experiments based on both real and synthetic data sets demonstrate that the new architecture and its underlying proposed algorithm optimize the real-time query processing to achieve minimal energy consumption and query latency, and provide secure and powerful storage infrastructure.

Keywords: *WBAN; Wireless body area networks; Cloud computing; Real-time database management; Query estimation; Query optimization.*

1. Introduction

The observed aging of population in many developed countries and the rising costs of health care, has been receiving a considerable interest from researchers, system designers, and application developers on a new type of network generally known as wireless body area networks (WBANs) [1, 5, 15, 27]. A wireless body area network is composed by small and intelligent devices, generally called body sensors, attached either on clothes, on body, or even implanted under the human skin. These devices are able to sense, sample, process, and wirelessly communicate continuous user's physiological state monitoring [11, 39]. The sensed data is sent to a nearby personal server (PS) device, e.g., a Personal Digital Assistant (PDA) or a Smartphone, which acts as a sink. Then, through a Bluetooth/WLAN connection, this data is streamed remotely either to a medical personnel's site for real-time diagnosis, or to a medical database for storage, or even to the corresponding equipment that issues an emergency alert [6, 15, 47]. Fig. 1 illustrates a WBAN and exemplifies the above-mentioned connection scenarios.

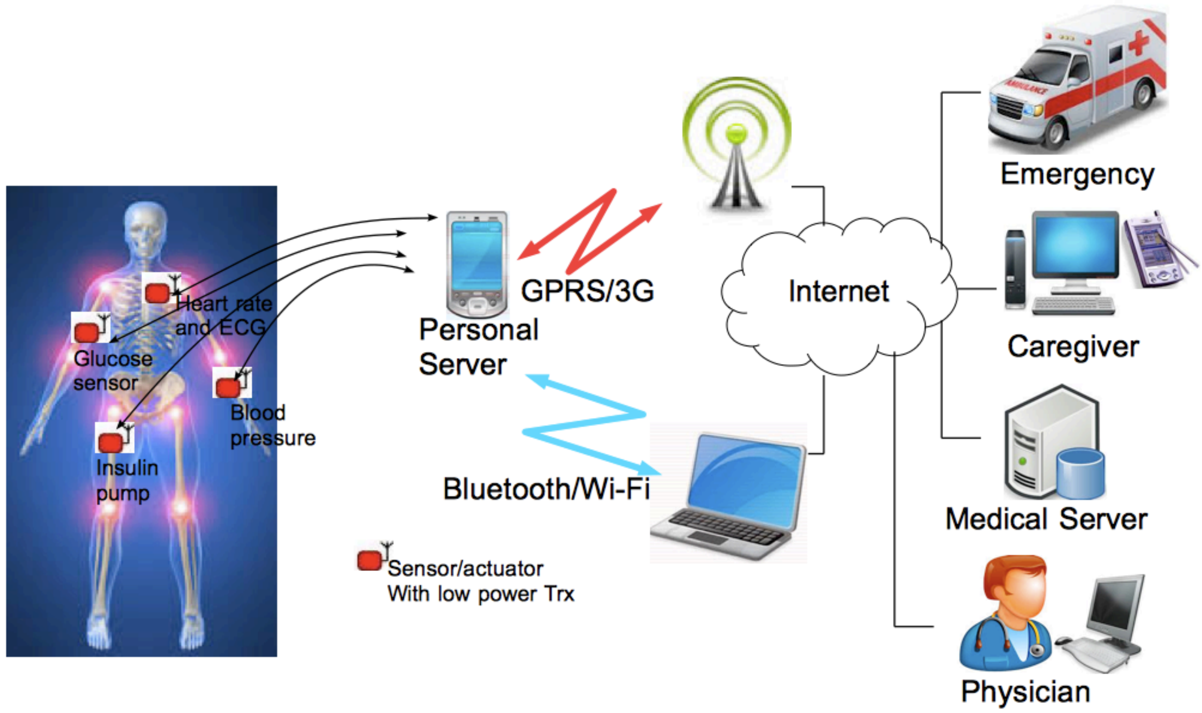


Fig. 1. Illustration of a WBAN architecture.

Generally, a wireless body area network uses two main types of devices, sensors and actuators. Sensors are used to measure certain parameters of the human body such as heartbeat, blood pressure, body temperature, recording prolonged electrocardiogram (ECG), etc. Actuators perform some specific actions according to the data they receive from the sensors or through interaction with users. For example, an actuator equipped with a built-in reservoir and pump administers the correct dose of insulin to diabetics based on the glucose level measurements from body sensors [27]. However, these tiny sensors are severely energy-constrained since they run either off a small battery or use energy scavenging techniques [34]. Therefore, optimizing energy consumption is an essential focus point in WBANs research [33, 44, 49].

Energy-efficient solutions for WBANs can be obtained at different levels [30], such as low power RF transceiver design [4], network architecture [35], energy scavenging [37], energy-efficient communication protocols [46], and node level [14] [2]. Since wireless communication activities are more expensive in energy consumption than the sensing and processing activities [20, 32, 33], saving energy by minimizing the data communication activities is one of the main research topics on this area and the main driver of this work.

Systems based on body sensor networks are getting increasingly used in many areas of application including vaginal temperature monitoring [10, 40], healthcare, athletic training, workplace safety, consumer electronics secure authentication, and safeguarding of uniformed personnel [12]. However, the processing time becomes increasingly critical for such applications. These applications must query and analyze the data in limited time periods in order to take real-time decisions and to react efficiently [19].

The data collected from a WBAN must closely reflect the current physiological state of the targeted patient. However, the patient's physiological state changes constantly and the data is collected in discreet moments of time. Then, the collected data has a temporal validity and as time advances it becomes less accurate until it does not reflect the physiological state of the patient any longer [23, 24]. It is then fundamental that responses of queries from medical applications ensure that returned data complies with logical and temporal constraints. Moreover, the energy constraints of sensors, the vital and large amount of data collected from WBAN nodes requires powerful and secure storage, and a query processing mechanism that minimizes the energy consumption and latency. In this context, the design of a secure and powerful storage infrastructure, and a query processing mechanism that takes into account both time-constraints of data and energy consumption is fundamentally important due to the real-time requirements of medical data and the resource limitations of WBAN nodes. To reach these goals, this work proposes a new architecture that combines a WBAN, which uses cloud services [9, 29, 48, 50] with statistical modeling techniques in order to optimize real-time user queries to achieve minimal energy consumption and latency, and provide a secure storage infrastructure. The main contributions of this paper are the following:

1. Proposal of a new architecture that combines a WBAN that uses cloud services with statistical modeling techniques in order to efficiently extract and store patient data, and provide to nearby or remote user/medical personnel a real-time diagnosis.
2. Proposal of a query processing algorithm, which is based on the new architecture and optimizes the real-time user query processing for minimizing the energy consumption and the query latency. The query processing algorithm uses a statistical model to compute the error and the confidence interval on the patient data stored in the medical database server located in the cloud. Thus, as long as these parameters are accepted by the query, the model can answer the query at the medical database server. Otherwise, the query must be satisfied in the WBAN and the model updated. Therefore, the body sensors are solicited only when the patient data in the medical database server is not fresh to answer the query with acceptable confidence. The error tolerance and confidence interval inform how much uncertainty the user can tolerate and are controlled by the applications, which define their constraints. One can notice that this information will be intuitive to many physicians, as they are the same as the error and confidence interval used for results in most scientific areas. In addition to gains in terms of latency and energy consumption, this proposal may also improve fault tolerance because the model can estimate some readings of unreachable body sensors for some time.
3. Performance evaluation study of the proposed architecture and its underlying query processing algorithm. The experiment results based on real data sets as well as synthetic data sets demonstrate that this combination and the proposed algorithm optimize the real-time user query processing to save more energy while guaranteeing good individual query latency.

The rest of this paper is organized as follows. Section 2 presents important related work about WBANs. Section 3 briefly presents the characteristics of real-time data, while the proposed architecture and its underlying query processing algorithm are described in Section 4. The performance analysis and results of the new architecture with its query processing mechanism for secure and powerful storage, and real-time data diagnosis are presented in Section 5. Finally, Section 6 concludes the paper and points suggestions for further works.

2. Related work

In the literature, several works were dedicated to improve the WBANs in terms of either energy minimization or latency, or secure and powerful information storage. Among these published works the most relevant for this study are included in this section.

The authors of [34] proposed a convex optimization framework to design a communication schedule in a body sensor network in order to achieve an optimal trade-off between energy consumption and latency. Their main idea is to propose a polling based algorithm that uses time-varying features of individual sensors and gives a decision-tree that helps to resolve scheduling conflicts among devices. In this kind of protocols each sensor is assigned periodic slots separated by polling interval for data update, on the contrary the mechanism proposed in this paper does not perform periodic data transfer. It estimates the query results and the sensors only transfer the data if the error is higher than a threshold defined by the query.

In [6], the authors proposed a prototype of cloud mobile health monitoring system. The proposed system uses a WBASN and a smart phone application that are based on cloud computing, location data, and a neural network to determine the patient's physiological state. In contrast to the proposed approach in this paper, the WBASN nodes of [6] continually senses and reports sensor readings according to a coordinator, leading then to fast energy depletion.

The WBANs transmit useful and life-critical information to a cloud in order to provide feedback decision support. Unfortunately, it also leads to some risks such as clients' privacy violation, malicious interactions to the storage infrastructure, etc. To prevent from this kind of problem, the work in [28] proposed a cloud-assisted privacy preserving mobile health monitoring system in order to protect the privacy of the involved entities and their data. This work focused on security aspects in cloud-assisted WBAN, but did not take into account the energy constraints of sensors and the real-time requirements of this kind of application.

In [13], the authors discussed the implementation of cloud computing solutions over e-Health services. For that they highlighted the requisites, issues, benefits, and barriers to deploy a cloud-based solution on e-Health services. Moreover, two scenarios of implementation of cloud-based solution of electronic health records (EHRs) management system have been provided, such as a large hospital and several primary care centers.

Another research work presented in [36] took advantage on mobility and proposed an autonomic system that integrates mobile computing [41, 42] and cloud computing in order to analyze ECG data. For that, the authors provided a cloud environment that collects people's health data, such as ECG data, and disseminates them to a cloud-based information repository, facilitating then the analysis of the data using software services hosted in the Cloud.

The research work in [45] proposed an energy-saving MAC scheme with dynamic transmission thresholds for body sensor networks. Their main idea is to limit the transmitted data based on dynamic transmission thresholds. Thus, in this approach a central control unit (CCU) performs a capital role in the WBAN. It receives the acceptable thresholds (e.g., alarm latencies, alarm thresholds, and transmission thresholds) from medical application, forms a Time Division Multiple Access (TDMA) frame including beacon, and broadcasts it into the WBAN. Finally, the CCU executes the data polling algorithm to choose data from body sensor nodes and transmits collected data to medical application. In contrast, the approach in this paper estimates the query answers according to the data error thresholds from medical applications at the cloud and the nodes are solicited only if these thresholds are exceeded by the error on the data.

In [43], the authors tried to enhance the processes of patients' vital data collection in healthcare institutions. Their main idea is to automate this process by using sensors attached to existing medical equipment that are inter-connected to exchange service. Thus, the information is reported in the cloud from where it can be processed by expert systems and/or distributed to medical staff. This work provides a full end-to-end telemedicine system based on cloud services and wireless sensors, but it does not tackle the problem of data transfer from the sensors and resource constraints.

Table I provides a summary of the above-described research works related to WBANs highlighting their characteristics and scientific contributions.

Table 1
Summary of the main WBAN proposed approaches and their features.

Authors	**Performances metrics**	**Data communication approach**	**Cloud-based design**	**Real-time constraints aware**
Nabar *et al*. [34]	Tradeoff b/w energy and latency	Polling based method with periodic slots of data transfer	X	X
Bourouis *et al*. [6]	X	Continual report of sensor readings	✓	✓
Lin *et al*. [28]	Privacy preserving of data	X	✓	X
Fernandez-Cardenosa *et al*. [13]	Economic metrics	X	✓	X
Pandey *et al*. [36]	Scalability, Economic metrics	X	✓	✓
Shahrokhi and He	energy	Data reduction	X	X

[45]		techniques		
Rolim *et al*. [43]	latency	X	✓	✓

In all the research that was carried out within this framework, none of the work that combines cloud-based WBANs with statistical modeling techniques for providing powerful shared storage infrastructure, as well as a real-time data processing and energy-efficient has been found. Thus, the aim of this work is to complete previous studies by profiting from the integration of WBANs with cloud computing and statistical modeling techniques to optimize real-time user queries to achieve minimal energy consumption and latency and provide secure storage infrastructure.

3. Characteristics of real-time data

The main purpose of a real-time database management system (RT-DBMS) is to process transactions within time, while maintaining logical and temporal consistency of data. The temporal consistency seeks to maintain consistency between the current state of the targeted environment and the state as reflected by the database contents. There are two ways to measure the temporal consistency [21, 22, 23, 24, 38]:

1. **Absolute consistency**: the absolute consistency seeks to maintain the view representing the state of the targeted environment consistent with the current state of the environment;
2. **Relative consistency**: the relative consistency concerns data derived from other ones.

The real-time data represents the capture of the current state of the targeted environment. For example, in the field of medical monitoring, micro sensors can be placed in the human body to control blood pressure, breathing, glucose, etc. These data should reflect as closely as the current state of the target environment. However, the environment changes constantly and the data are collected in discreet times. So, the collected data have temporal constraints. To satisfy these temporal constraints, the structure of the data must include these attributes: *i*) the time label, which indicates the time instant when the observation relating the data was made, *ii*) the absolute validity interval (avi) which denotes the time interval following the time label during which data is considered valid. For the data quality, another attribute that should be considered is the imprecision or data error (*DE*), which indicates how the current state of the targeted environment may differ from the measured data [38]. The data error on a data version *d* is defined by the equation (1). More detailed information can be found in [22, 38].

$$DE(d) = 100 * \left|\frac{CurrentValue(d) - UpdateValue(d)}{CurrentValue(d)}\right| (\%) \qquad (1)$$

4. System Architecture

It is assumed that the topology of the WBAN is stable, well known and the person which wears the body sensors is not moving too much, e.g. patient in a hospital bed or at home. Moreover, it is assumed that interacting applications can tolerate to read inconsistent data under some limits. This relaxation is herein used to improve real-time transaction processing. For instance, in [17], an algorithm that uses tolerated inconsistent data to deal with conflicted real-time transactions is proposed.

Generally, queries on data from a WBAN request real-time information about the physiological activities and actions of the human body and its surrounding environment (e.g., health status and motion pattern). Therefore, one of the goals of this work is to process queries in real-time intended to a cloud-based WBAN and to reduce the communication cost and the processing load in terms of energy. This will provide real-time diagnosis of data in remote sites of users (nurse, medical doctor, etc.) and improve the lifetime of body sensors according to the requirements (data freshness) of the applications. Fig. 2. illustrates the proposed architecture, which is composed of three components. Tier 1 represents the WBAN, Tier 2 represents the communication between the personal server (PS) and the access points (APs), and Tier 3 represents the cloud environment, which includes among other components series of medical servers accessed through the Internet.

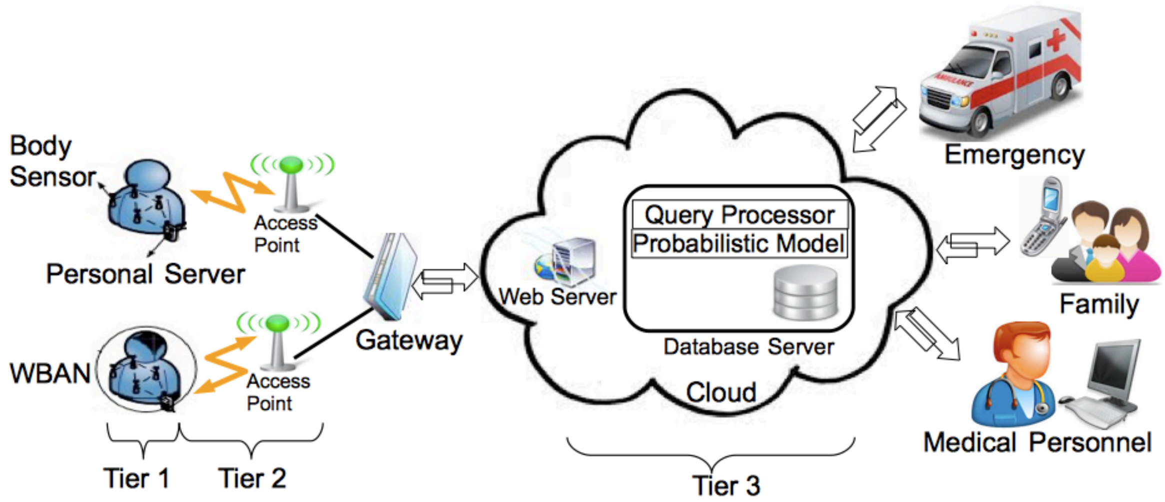


Fig. 2. Illustration of a three-tier architecture of Cloud-based WBAN.

The description of the whole system architecture is performed below. Moreover, it clearly describes how this architecture could be used to optimize the user queries for real-time diagnosis, decision-making, and energy saving.

4.1 Tier 1: Wireless body area networks

A WBAN refers to radio communications [25] of about 2 meters around the human body, which considers the communications among body sensors and the communications between body sensors and a PS. It can include a heart sensor and motion sensors, among others. For example, these two sensors can be used in a monitoring system of cardiac patients during a rehabilitation period in which the heart sensor will report either raw ECG signals, time stamped heart beats, or averaged heart rate over a certain period of time and motion sensors, each one equipped with a 3D accelerometer, will report either raw acceleration signals for x, y, and z axes, extracted features (e.g., time stamped steps), or an estimated level of activity [6]. A body sensor includes, among other components, a power unit, a processor, a memory, and a transceiver. The body sensors wait for demand from the PS and transmit the sensor readings in order to update the data in medical databases. This demand is performed only when the model in a database server is not able to satisfy a query with acceptable error. This will greatly reduce the communication activities of the body sensors, increasing then the network lifetime.

4.2 Tier 2:Communication between the personal server and acces points

The Tier 2 represents the communication between the PS and one or more APs, and it is used to interconnect WBANs with the cloud. Thus, the PS represents the sink node and it can be a dedicated unit, such as a personal digital assistant (PDA) or a smartphone. The main purpose of this unit is to collect all the data from the body sensors, aggregate them if necessary, establish a secure connection to the cloud, and transmit the data about users' health status through the gateway. This data is recorded into medical database servers in order to be accessible by remote users (nurse, medical doctor etc.). The main components of this device are a power unit, a large processor, a large memory, and a transceiver.

4.3 Tier 3: Cloud environment

Besides medical database servers, the cloud may encompass other servers, such as commercial health care providers, emergency services, etc. accessible via Internet. In this proposed architecture, a medical database server is composed, among other components, of a *query processor*, a *probabilistic model*, and a *database*. The *query processor* may process either historical query or real-time query with respect to query's parameters. A database stores medical data of registered patients. The *probabilistic model* uses the data stored in a database to estimate the answer of the real-time queries according to their required data freshness constraints. These answers should represent the data about the current human's physiological activities and actions, and they have temporal validity. As time advances the model becomes less accurate, until the time where it cannot answer some queries with acceptable error and confidence interval. When that happens the WBAN should be solicited to answer the query and update the body sensor data in the medical database server.

4.3.1 Probabilistic Model

The proposed architecture is general, however the probabilistic model used depends on the physiological phenomenon studied. The case study considered in this work focuses on the monitoring of the temperature of the human body. Therefore, this work focuses, without loss of generality and for exemplification purposes, on a model based on the Gaussian distribution [3, 8].

It is known that the probability density function (*pdf*) representing the variation of temperatures follows a Gaussian distribution [18]. Thus, a random variable X follows a normal distribution with parameters μ and σ, denoted $N(\mu, \sigma)$, if its probability density equation is the equation (2).

$$P(X) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{X-\mu}{\sigma})^2} \qquad (2)$$

whose graph is the famous *"bell curve"*, and the two parameters μ and σ respectively represent the mean and standard deviation of X. The probability distribution is symmetrical with respect to μ, which therefore characterizes the central tendency, and σ is the dispersion of the distribution with:

$$Prob\{|X-\mu| < \sigma\} = 68.26\% \qquad (3)$$

$$Prob\{|X-\mu| > 1.96\sigma\} = 5\% \qquad (4)$$

$$Prob\{|X-\mu| > 2.58\sigma\} = 1\% \qquad (5)$$

This proposal focuses on optimizing user queries by combining a cloud-based WBANs with a probabilistic model. Let us consider that the model is the probability density function *(pdf)* with equation (2), $p(X)$ assigning a probability for each possible assignment to the random variable X, representing the variation of temperatures of a given human body. For building the representation of this *pdf* of the temperatures variation, the proposal uses the data stored in the medical database server. Thus, after construction of the representation of the *pdf*, the probabilistic model is used to estimate the answer of user queries about the current state of body sensor readings according to the data freshness. When a user query arrives with its error and confidence interval that cannot be satisfied by the model, it is executed in the WBAN and the model is updated on demand.

4.4 Query processing optimization for energy minimization and latency

The probabilistic model used in this work can answer many complex query types. However, for exemplification purposes, this section focuses on attribute-value estimates that are used in the case study.

4.4.1 Query estimation

The objective of this query estimation is to estimate the answer of real-time user queries based on statistics. The users can query the information in one body sensor or in a set of body sensors among the sensors composed the considered WBAN. For exemplification purposes, consider that a SQL query submitted to the medical database server arrives at time t_2 and asks to estimate the temperature values of a specific body sensor S_i in the WBAN with an error ε and a confidence $1-\theta$ ($\theta \in [0,1]$). This query is parsed and translated by the query processor into probabilistic computations over the model. Suppose that the last reading of the body sensor S_i in the medical database server was acquired at time t_1 and the total sum of acquisition times of all the readings of S_i in that database is T. The difference t_2-t_1 gives us the late time of the query from the last reading. Let us denote this time interval Q. By using the previous *pdf*, one can compute the expected answer of the query, the mean μ

and the standard deviation σ. Then, the average variance per time unit is given by σ^2/T and the exact variance of the temperature between the last reading and the arrival of the query is given by the equation (6) below:

$$\varphi = Q * \sigma^2/T \tag{6}$$

Thus, this calculation will help to take care of the temporal validity of the real-time data in the medical database server, which is intrinsically given by the error one is willing to take. After these computations, the model can answer the user query if these two conditions are met:

3. $\varphi \leq \varepsilon$ (7)
4. $P(X \in [\mu - \sigma, \mu + \sigma]) \geq 1 - \theta$, with $P(X \in [\mu - \sigma, \mu + \sigma]) = \int_{\mu-\sigma}^{\mu+\sigma} p(x)dx$ (8)

If one of these conditions is not true, then the model is not sufficiently rich to answer the query. Below, the query routing algorithm is discussed to better explain how the user query and the updates are processed.

4.4.2 Query processing algorithm

Notice that a user query requests real-time information about the physiological activities and actions of the human body and its surrounding environment. Moreover, a query is a triplet of objects constituted by the request (the required data and the targeted body sensors), the tolerated error, and the targeted confidence interval that specify how much uncertainty the user is willing to tolerate. Thus, when a user query arrives, it is intercepted by the query processor, which verifies the freshness requirement of the query according to the freshness of the body sensor data in the model:

1. If the tolerance for the freshness is verified, it means that the model is sufficiently fresh to answer the query, then the query is estimated in the medical database server and the results sent back to the user.

2. If the tolerated freshness is not verified, then the query is executed in the WBAN (which is supposedly more fresh), and while sending back the results to the user, the model is also updated in order to refine the estimates. The algorithm of a user query processing with 95% confidence (see equations in the sub-section 4.3.1) is given at Fig. 3.

```
CT = GetCurrentTime() + processing_time;
TL = GetTimeOfLastMeasure(query);
VAR = GetVarianceForTimePeriod(CT,query);
if( VAR <=given_value ){
LM = GetLastMeasure(query);
AV = GetAvgForTimePeriod(query);
return "last measure: " + LM + " in time: " + TL + "Average for this
time period: " + AV + " estimated error (95% confidence): " + 2*VAR;
}else{
go to WBAN;
return results;
storeNewDataAsHistoricalData(values);
updateStatistics();
}
```

Fig. 3. Query processing algorithm.

5. Performance analysis

The goal of this section is to compare the performance indices of the cloud-based WBAN architecture and its query processing algorithm proposed in this paper against a same architecture called *compared model* here for distinction purposes, but with a query processing where all the body sensors must send their readings to the medical database server periodically in order to refresh the sensor data to be accessed by user queries. This last query processing method is the most used one in WBANs architectures, however it leads to a high-energy consumption in the system. Moreover, in this compared model, the real-time user queries must wait for updates if the data in the medical database server is not sufficiently rich to answer the queries, this leads to performance slowdown in terms of latency.

5.1 Scenario description

The work presented in this paper provides a secure and powerful storage, and a query processing mechanism that takes into account both the temporal-constraints of data and the energy consumption in WBAN architectures. In this architecture, the query processor uses the variance and the confidence interval of the current sensor readings in the medical server in order to either answer the query or route it in the physical WBAN considered fresher. Furthermore, the data in the medical database server will be update only on demand. Therefore, the communication cost is reduced and the latency of the query is improved.

The type of query examined in this work is the attribute value estimates. Thus, the user can query the information in one body sensor or in a set of body sensors, within a specified error ε and a confidence interval $1-\theta$ ($\theta \in [0, 1]$).

The performance indices, which are in interest in this work, are the energy cost and the latency of the user query processing. For exemplification purposes, the energy cost is determined according to the nodes visited by the query, the data sensing, processing, transmission, and the sensor's duty cycle operation. Thus, the equation 9 gives the energy consumption of the sensor node. More details about the calculation of each term of the equation can be found in [7].

$$E_{node} = E_{sensor} + E_{ADC} + E_{\mu c} + E_{transceiver} \qquad (9)$$

To capture the current energy consumption of each sensor activity in the simulation, each activity needs to be calibrated by real measures. For that, the cost model uses numbers obtained from the Shimmer mote datasheet [16, 26, 31]. Fig. 4. shows a Shimmer (Sensing Health with Intelligence, Modularity, Mobility and Experimental Reusability) mote, whereas the Table 2 provides an overview of the characteristics of a Shimmer mote.

Fig. 4. Photo of a SHIMMER mote.

Table 2
Nominal characteristics of Shimmer nodes.

Processor	Bus	Flash	RAM	EEPROM	ADC	Serial Communication	Current Drawn		Communication		
							Active	Sleep			
MSP430	16bit	48 K	10 K	None	12bit	UART	1.8mA	5.1 µA	IEEE 802.15.4		
									Sleep	Idle/Rx	Tx
									1-426 µA	18.8mA	17.4mA
									Bluetooth		

To calculate the latency of each query let us assume that N queries are sent to the system and each query Q_i has an instant time A_i that it was sent to the system and an instant time F_i that it finishes to be processed. Then, the latency of each query is given by equation (10).

$$\bar{R} = \frac{1}{N}\sum_{i=1}^{N}(F_i - A_i) \qquad (10)$$

This work is interested in the variation of the energy cost in the WBAN and the latency for the treatment of the real-time user queries according to the data consistency and to the workload of the system. To obtain these performance criteria, the tolerated error is modified along the experiments. Moreover, the body sensor nodes in the compared model are programmed to periodically send their readings to update the real-time data in the medical database server located inside the cloud and the same number of query is submitted to the two models, this will represent the charge of the system.

5.2 Results analysis

This section presents the simulation results of the proposed architecture and its query mechanism to optimize the real-time query processing. These experiments are based on real data sets as well as synthetic data sets.

5.2.1 Energy cost *versus* freshness relaxation

To measure the impact of relaxing the freshness in the energy consumption, the simulation uses a WBAN composed of ten sensors and the user queries are submitted to the medical database server. The queries ask the temperature readings from the sensors within a specified tolerated error and a confidence interval. In these experiments the tolerated error varies between *0* and *1* degrees Celsius and the confidence interval is fixed to *95%*. The results obtained are shown in Fig. 5.

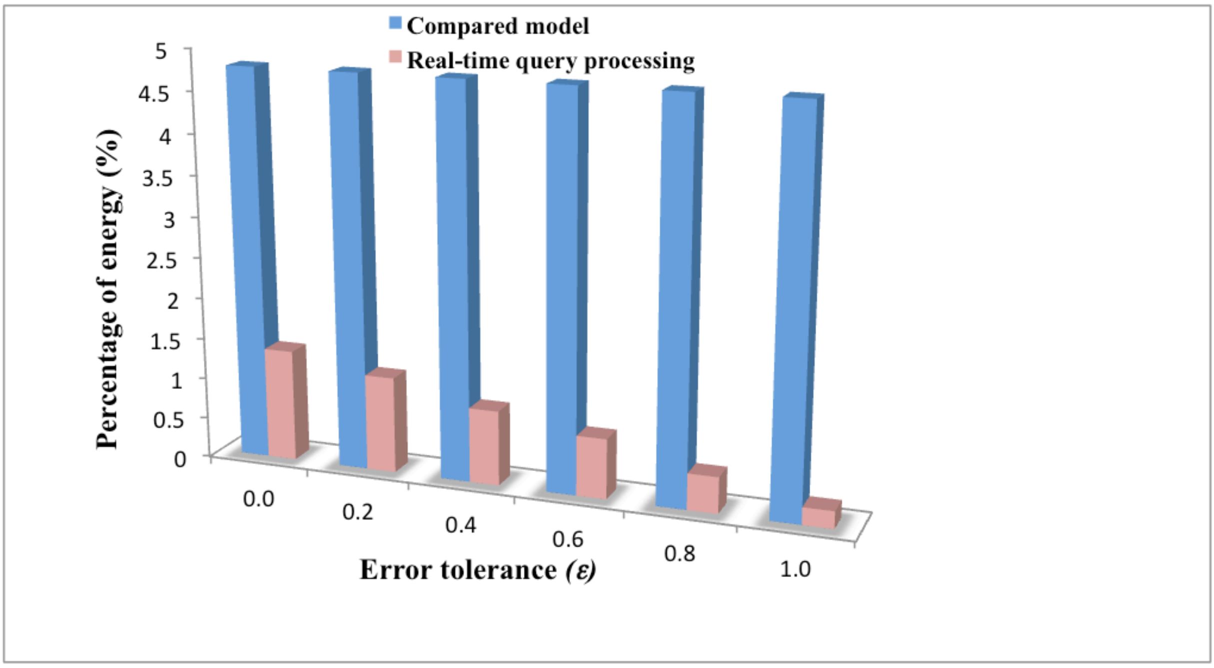


Fig. 5. Percentage of energy per query *versus* error tolerance with a confidence interval of 95%.

As expected, when the tolerated error (ε) increases the percentage of energy consumed in the compared model remains high. This is due to the fact that the nodes are programmed to periodically send all their readings to the medical database server independently to the relaxation, what involves big energy consumption in the body sensors. However, the percentage of energy consumption of the new proposed model remains very smaller and decreases gradually. This can be explained by the fact that, in this new model, the communication cost decreases gradually with respect to the relaxation; most of the queries are estimated in the medical database server that not involves much data transmission from the body sensors to the server. The medical database server will be updated only if the error and the confidence interval of the actual data in the model are not accepted by the query.

5.2.2 Individual query latency *versus* freshness relaxation

This study tries to show the impact of relaxing the freshness in the query latency. The simulation experiments are based on the same network configuration as the previous one. In addition, the queries are sent to the medical database server to ask the temperature readings from the sensors within a specified tolerated error and a confidence interval. The tolerated error varies between *0* and *1* degrees Celsius and the confidence interval is fixed to *95%*. The results obtained are shown in Fig. 6.

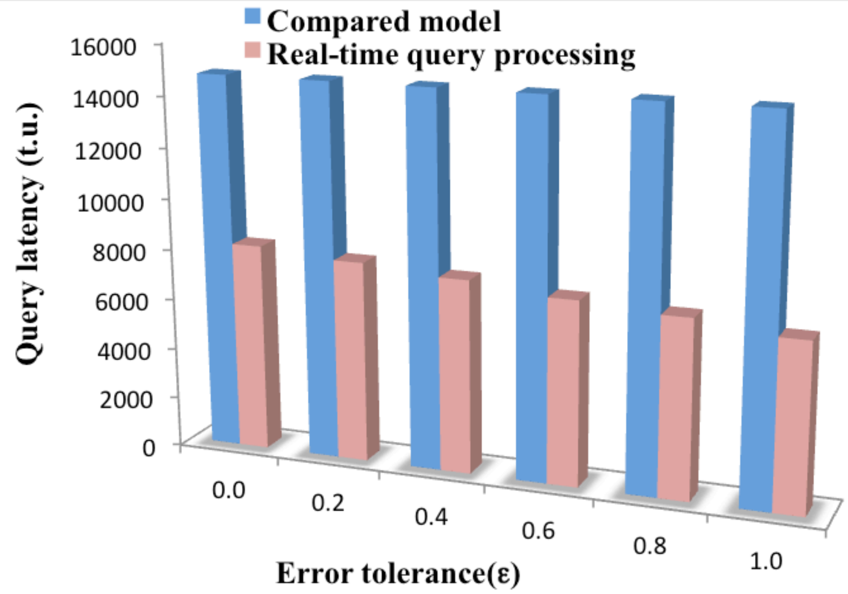


Fig. 6. Individual query latency *versus* error tolerance with a confidence interval of 95%.

As depicted in Fig. 6., when the tolerated error (ε) increases, the individual query latency in time unit (t.u.) for the compared model remains more or less big and constant because this model does not tolerate freshness relaxation. However, the individual query latency in the new proposed model is better and it decreases progressively. One has this result because in the proposed model query results are approximated according to data freshness requirements of queries. In addition, if the data in the medical database server is not sufficiently fresh to approximate the results, the query is not put in a wait mode; it is executed directly in the physical WBAN. In contrast, in the compared model if the data is outdated, the queries are put in a wait mode for updates. This leads to a great waste of time without even it has a substantial variation in the WBAN.

6. Conclusion and future work

This work proposed a new architecture of cloud-based WBAN with its underlying query processing algorithm for secure and powerful storage, energy-efficient and real-time data processing. Instead of periodically sending the sensor readings from the WBAN to the medical database server in the cloud for off-line processing that involves time delay and energy wasting, the proposed approach combines a WBAN with cloud services and statistical modeling techniques to perform a query processing optimization that uses the error tolerance and the probabilistic confidence interval as query execution criterions. Thus, the probabilistic model uses the patient data stored in the medical database server to approximate the answer of the query according to a given error tolerance and confidence interval. If the patient data inside the medical database server is not sufficiently rich and fresh to answer the query, the query processor routes the query towards the physical WBAN to obtain the needed real-time data. The experimental results based on real data sets as well as synthetic data sets show that the proposed framework provides, among other advantages, good individual query latency with valid data for real-time medical diagnosis and energy-efficient processing.

Mobility is an important aspect in WBANs. Thus, many patients need physical mobility and are no longer compelled to stay in a healthcare structure. For that, future work can focus on extending this proposed architecture and its query mechanism in order to take into account mobility of patients and corresponding devices.

AKNOWLEDGMENTS

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 a Tecnologia* through the PEst-OE/EEI/LA0008/2013 Project, and by the AAL4ALL (Ambient Assisted Living for All), project co-funded by COMPETE under FEDER via QREN Programme, and by NSFC (61170296 and 61190125).

REFERENCES

[1] B. Antonescu, S. Basagni, Wireless body area networks: challenges, trends and emerging technologies, in: BodyNets '13 Proceedings of the 8th International Conference on Body Area Networks, ICST, Brussels, Belgium, 2013, pp. 1-7, ISBN: 978-1-936968-89-3, doi>10.4108/icst.bodynets.2013.253722.
[2] S. Bahanfar, L. Darougaran, H. Kousha, S. Babaie, Reliable Communication in Wireless Body Area Sensor Network for Health Monitoring, IJCSI International Journal of Computer Science Issues, 8 (3) (September 2011), Issue 5, ISSN (Online): 1694-0814, pp. 366-372.
[3] C. Belsom, The Normal distribution, Cambridge University Press, 1992.
[4] J. L. Bohorquez, A. P. Chandrakasan, J. L. Dawson, A 350 µW CMOS MSK Transmitter and 400 µW OOK Super-Regenerative Receiver for Medical Implant Communications, IEEE JOURNAL OF SOLID-STATE CIRCUITS, 44 (4) (APRIL 2009), pp. 32–44.
[5] A. Boulis, D. Smith, D. Miniutti, L. Libman, Y. Tselishchev, Challenges in body area networks for healthcare: the mac, IEEE Communications Magazine, 50 (5) (May 2012), pp. 100–106.
[6] A. Bourouis, M. Feham, A. Bouchachia, A New Architecture of a Ubiquitous Health Monitoring System: A Prototype of Cloud Mobile Health Monitoring System, International Journal of Computer Science Issues (IJCSI), 9 (2) (March 2012), pp. 434-439
[7] R. Briand, O. Arrijuria, V. Dupé, C. Richon, G. Terrasson, Evaluation de la Consommation énergétique d'un nœud: Du hard au soft, in: RESSACS 2010, 2010.
[8] W. Bryc, The normal distribution: characterizations with applications, Springer-Verlag, 1995.
[9] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility, Future Generation Computer Systems, 25 (6) (2009), pp. 599-616.
[10] J. M. P. L. Caldeira, J. J. P. C. Rodrigues, J. F. R. Garcia, I. de la Torre, A New Wireless Biosensor for Intra-Vaginal Temperature Monitoring, Sensors Journal, MDPI, Basel, Switzerland, 10 (11) (November 2010), pp. 10314-10327, DOI: 10.3390/s101110314.
[11] J. M. L. P. Caldeira, J. J. P. C. Rodrigues, P. Lorenz, Towards Ubiquitous Mobility Solutions for Body Sensor Networks on HealtCare, IEEE Communications Magazine, IEEE, 50 (5) (May 2012), pp. 108-115, ISSN: 0163-6804, DOI: 10.1109/MCOM.2012.6194390.
[12] H. Cao, C. Chow, H. Chan, V. Leung, Enabling technologies for wireless body area networks: a survey and outlook, IEEE Wireless Communication Mag, 47(12) (2009), pp. 84–93.
[13] G. F. Cardenosa, I. de la Torre-Diez, M. Lopez-Coronado, J. J. Rodrigues, Analysis of cloud-based solutions on EHRs systems in different scenarios, Journal of Medical Systems, 36 (6) (2012), pp. 3777–3782.
[14] F. Casamassima, E. Farella, L. Benini, Power saving policies for multipurpose WBAN, in: 23rd International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), (IEEE 2013), pp. 83–90, DOI 10.1109/PATMOS.2013.6662159.

[15] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, V.C.M. Leung, Body area networks: A survey, ACM/Springer Mobile Networks Applications, 16(2) (2011), pp. 171–193, URL http://dx.doi.org/10.1007/s11036-010-0260-8.
[16] Intel Corporation, The SHIMMER Sensor Node Platform, Available online from http://www.eecs.harvard.edu/~konrad/projects/shimmer/SHIMMER-GettingStartedGuide.html, 2013.
[17] C. L. Dantas, L. E. Pereira, R. N. P. Fernandes, Real-Time Databases Techniques in Wireless Sensor Networks, in: Sixth International Conference on Networking and Services, (IEEE 2010), 2010, pp. 182–188.
[18] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, W. Hong, Model-Driven Data Acquisition in Sensor Networks, in: 30th VLDB Conference, Toronto, Canada, September 2004, pp. 588-599.
[19] O. Diallo, J. J. P. C. Rodrigues, M. Sene, Real-time data management on wireless sensor networks: A survey, Journal of Network and Computer Applications, 35 (3) (May 2012), pp. 1013-1021, http://dx.doi.org/10.1016/j.jnca.2011.12.006.
[20] O. Diallo, J. J. P. C. Rodrigues, M. Sene, J. Lloret, Distributed Database Management Techniques for Wireless Sensor Networks, IEEE Transactions on Parallel and Distributed Systems, IEEE computer Society Digital Library, IEEE Computer Society, (19 Aug. 2013a), http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.207
[21] O. Diallo, J. J. P. C. Rodrigues, M. Sene, J. Lloret, Simulation Framework for Real-Time Database on WSNs, Journal of Network and Computer Applications, 39 (2014), pp. 191–201, http://dx.doi.org/10.1016/j.jnca.2013.07.001.
[22] L.C. DiPippo, V. F. Wolfe, Real-time databases, Journal of Database Systems Hand-book, Multiscience Press (Citeseer 1997), pp. 1-57.
[23] N. Idoudi, C. Duvallet, B. Sadeg, R. Bouaziz, F. Gargouri, How to model a real-time database?, in: Proceedings of 12th IEEE International Symposium on Object-oriented Real-time distributed Computing (IEEE ISORC'2009). Tokyo, Japan: IEEE, 2009a, pp. 321-325.
[24] N. Idoudi, N. Louati, C. Duvallet, R. Bouaziz, B. Sadeg, F. Gargouri, A Framework to Model Real-Time Databases, International Journal of Computing and Information Sciences 7 (January 2009b), pp. 1-11.
[25] R. H. Jacobsen, K. Kortermand, Q. Zhang, T. S. Toftegaard, Understanding Link Behavior of Non-intrusive Wireless Body Sensor Networks Wireless Personal Communications, Wireless Pers Commun, 64 (3) (June 2012), pp 561-582, DOI 10.1007/s11277-012-0601-y
[26] M. Johnson, M. Healy, V. de Ven, M. Hayes, J. Nelson, T. Newe, E. Lewis, A Comparative Review of Wireless Sensor Network Mote Technologies, in: IEEE Sensors Conference, 2009, pp. 1439 – 1442, 10.1109/ICSENS.2009.5398442.
[27] B. Latre, B. Braem, I. Moerman, C. Blondia, P. Demeester, A survey on wireless body area networks, Journal of Wireless Networks, 17(1) (2011), pp. 1–18, URL http://dx.doi.org/10.1007/s11276-010-0252-4.
[28] H. Lin, J. Shao, C. Zhang, Y. Fang, CAM: Cloud-Assisted Privacy Preserving Mobile Health Monitoring, Information Forensics and Security, IEEE Transactions on 8(6) (June 2013), pp. 985 – 997.
[29] J. Lloret, M. Garcia, J. Tomas, J. J.P.C. Rodrigues, Architecture and protocol for intercloud communication, Information Sciences, Volume 258, (February 2014), pp. 434-451.
[30] B. Lo, G. Yang, Body Sensor Networks-Research Challenges and Opportunities, in: IET Seminar on Antennas and Propagation for Body-Centric Wireless Communications, 2007, pp. 26–32.
[31] V. Madan, SRN Reddy, Review of wireless sensor mote platforms, VSRD international journal of Electrical, Electronic & communication Engineering, 2(2) (2012), pp. 50-55.
[32] S. Madden, M. J. Franklin, J. M. Hellerstein, W. Hong, TAG: A tiny aggregation service for ad-hoc sensor networks, in: OSDI, Citeseer, 2002.
[33] F. Marcelloni, M. Vecchio, Enabling energy-efficient and lossy-aware data compression in wireless sensor networks by multi-objective evolutionary optimization, Information Sciences 180 (10) (2010), pp. 1924–1941.
[34] S. Nabar, J. Walling, R. Poovendran, Minimizing energy consumption in body sensor networks via convex optimization, in: BSN'10: Proceedings of the 2010 International Conference on Body Sensor Networks, IEEE Computer Society, Washington, DC, USA, 2010, pp. 62–67.
[35] A. Natarajan, B. Silva, K. Yap, M. Motani, To hop or not to hop: Network architecture for body sensor networks, in: IEEE SECON, 2009.
[36] S. Pandey, W. Voorsluys, S. Niu, A. Khandoker, R. Buyya, An autonomic cloud environment for hosting ECG data analysis services, Future Generation Computer Systems 28(1) (2012), pp. 147–154, http://dx.doi.org/10.1016/j.future.2011.04.022.
[37] J. Paradiso, T. Starner, Energy scavenging for mobile and wireless electronics, IEEE Pervasive Computing, 4 (1) (2005), pp. 18–27, 10.1109/MPRV.2005.9.
[38] K. Ramamritham, Real-time databases, Journal of Distributed and Parallel Databases (Springer 1993), 1(2), pp.199–226.

[39] S. N. Ramli, R. ahmad, Surveying the Wireless Body Area Network in the realm of wireless communication, in: 7th International Conference on Information Assurance and Security (IAS), IEEE 2011, pp. 58–61, DOI 10.1109/ISIAS.2011.6122845.
[40] J. J. P. C. Rodrigues, J. Caldeira, B. Vaidya, A Novel Intra-body Sensor for Vaginal Temperature Monitoring, Sensors Journal, MDPI, Basel, Switzerland, 9(4) (2009), pp. 2797-2808, ISSN 1424-8220.
[41] J. J. P. C. Rodrigues, I. M. C. Lopes, B. M. C. Silva, I. de la Torre, A New Mobile Ubiquitous Computing Application to Control Obesity: SapoFit, Informatics for Health and Social Care, Information Healthcare, 38 (1) (January 2013), pp. 1-29, ISSN (online): 1753-8165, ISSN (print): 1753-8157, DOI: 10.3109/17538157.2012.674586.
[42] J. J. P. C. Rodrigues, L. M. C. C. Pedro, T. Vardasca, I. de la Torre-Díez, H. M. G. Martins, Mobile Health Platform for Pressure Ulcer Monitoring with EHR Integration, Health Informatics Journal, SAGE Publications, UK, 19(4) (2013), pp. 300-311, ISSN: 1460-4582, Online ISSN: 1741-2811, doi:10.1177/1460458212474909
[43] C. O. Rolim, F. L. Koch, C. B. Westphall, J. Werner, A. Fracalossi, G. S. Salvador, A Cloud Computing Solution for Patient's Data Collection in Health Care Institutions, in: Proc. of ETELEMED. IEEE, 2010, pp. 95–99.
[44] M. Saleem, I. Ullah, M. Farooq, BeeSensor: An energy-efficient and scalable routing protocol for wireless sensor networks, Information Sciences, 200 (1) (October 2012), Pages 38–56, http://dx.doi.org/10.1016/j.ins.2012.02.024.
[45] S. M. Shahrokhi, Y. He, Energy-Saving MAC Scheme with Dynamic Transmission Thresholds for Body Sensor Networks, International Journal of Sensors and Sensor Networks 1(6) (2013), pp. 69-74, doi: 10.11648/j.ijssn.20130106.11.
[46] V. Shankar, A. Natarajan, S. Gupta, L. Schwiebert, Energy-efficient protocols for wireless communication in biosensor networks, in: Proceedings of the 12th IEEE Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC01), vol. 1, 2001, pp. 114–118, 10.1109/PIMRC.2001.965503.
[47] S. Ullah, H. Higgins, B. Braem, B. Latre, C. Blondia, I. Moerman, S. Saleem, Z. Rahman, K. Kwak, A comprehensive survey of wireless body area networks, Journal of Medical Systems, Journal of Medical Systems, 36 (3) (2012), pp. 1065-1094, DOI: 10.1007/s10916-010-9571-3.
[48] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, A. V. Vasilakos, Security and privacy for storage and computation in cloud computing, Information Sciences, 258 (2014), pp. 371-386, http://dx.doi.org/10.1016/j.ins.2013.04.028.
[49] Q. Zhang, Energy Saving Efficiency Comparison of Transmit Power Control and Link Adaptation in BANs, in: IEEE International Conference on Communications (ICC) 2013, Budapest, Hungary, June 9-13 2013, pp. 1672 – 1677, 10.1109/ICC.2013.6654757.
[50] Y. Zhuang, N. Jiang, Z. Wu, Q. Li, D. K.W. Chiu, H. Hu, Efficient and robust large medical image retrieval in mobile cloud computing environment. Information Sciences, 263 (2014), pp. 60-86, http://dx.doi.org/10.1016/j.ins.2013.10.013.

Chapter 7

Conclusions and Future Work

This chapter presents the main conclusions resulting from the research work of this thesis. Furthermore, it provides some open research issues that may be addressed in the future.

1. Final Conclusions

This thesis is focused on the performance assessment of real-time data management on wireless sensor networks and presents the research work accomplished with the purpose of providing a new query processing solution that meets the real-time requirements of real-time applications and the resources limitation of WSNs and a new tool that can help to simulate real-time database protocols for WSNs and approximately test the temporal and logical validity of data and transactions before deployment with real measures. The research work was divided as follows: the study of various techniques of real-time database management in wireless sensor networks as well as in traditional networks, the analysis of problems and relevant applications as base support of the problematic, the analysis of different distributed database management techniques for wireless sensor networks, the proposal and implementation of a new real-time database management model for wireless sensor networks, and finally the proposal of a new real-time query processing technique that optimizes both the latency and the energy consumption. These four research stages resulted in contributions of this thesis.

The possible integration of real-time applications and WSNs has given raise challenges to handle real-time data storage and query in energy-efficient way. In fact, in real-time applications data and tasks have temporal constraints. Moreover, since in WSNs energy is one of the most crucial resource and the on-going interaction between the network devices and the environment results in huge amounts of data, it is challenging to design a query processing mechanism that meets both time constraints and energy. Hence, the researchers have proposed some techniques. However, many of them are limited because of the underlying architecture used, either warehousing or distributed, or complex algorithms used that can have a negative impact in either the temporal constraints or the energy constraints. Therefore, the main goal of this thesis was to propose a new query processing method that meets the real-time requirements of real-time applications and the resources limitation of WSNs. The intermediate objectives were defined so as to divide the necessary research work to reach the main objective. The first part of the research work was described in chapter 2 and consisted in the study of various techniques of real-time database management in wireless sensor networks as well as in traditional networks and the analysis of their limitations. The literature review revealed that after the acquisition of the information, a major problem of data storage and exploitation arises, particularly for systems that deal with real-time constraints. In fact, building mechanisms that meet the requirements of this kind of applications is very difficult. Moreover, the data management techniques used in traditional databases are not generally suitable for sensor networks because of their specificities. Most of the proposals are based on the warehousing approach, which is a centralized approach that requires to periodically updating the central database. This can cause, among other drawbacks, delay on the response time and great energy depletion.

On the other hand, other proposals focus on the distributed approach that seems, based on its data access mode, provide little help on the temporal constraints, however with some risk because there may be a sudden failure of sensors because sensors usually operate in unstable environment. This may lead to a delay or lack of information that influences the analysis time or even on the blocking of the system.

Some other studies used complex algorithms mainly to satisfy temporal constraints with sometimes the lack of mechanism aware of the energy depletion. Moreover, in some of these algorithms the waiting or aborting of the transactions is determined by the temporal validity of the data to be accessed, yet when one uses temporal constraints directly, one can make the mistake of performing wait or abort the transactions too many times without having variations that justify these operations. This can lead to performance slowdown when one knows that, based on this study, in real-time systems, for some applications, the accuracy of results may be sacrificed under some limit to reduce the response time.

The different distributed database management techniques for wireless sensor networks were further studied and explored in the second part of the research work and were presented in chapter 3. The literature review showed that this approach is an alternative regarding the warehousing approach drawbacks and enforced the conclusions depicted in the previous research work. The sensor nodes do not need to send periodically the collected data to the base station. The sensed data remains on the sensor nodes and some queries are distributed and processed within the sensor nodes. This reduces the data communication, minimizing then the energy consumption in the network. Moreover, this approach offers quasi real-time query processing, support of long-running and instant queries. However, According to the unstable and generally harsh environment, there may be sudden failures of sensors. This can lead to information loss that greatly influences the result analysis or even the system blocking. In addition, in this approach the time delay is distance-sensitive and depends on the depth of the nodes that have the required information.

The third part of the research work was presented in chapter 4 and proposed a simulation framework for real-time database on WSNs. The research work made in the two previous chapter, mainly in the chapter 2 showed that several simulation tools for WSNs have been proposed and they can be divided into two classes according to the nature of specified constraints: the first class of simulators is oriented network and studies the network in the point of view communication;

whereas the second class of simulators is oriented node and emphasizes on the function of a single node with simple communication models. However, there is no specific tool for testing and validating real-time database techniques for WSNs. The model is based on the distributed architecture regarding its advantages and uses the Earliest Dead-line First (EDF) protocol to schedule transactions and the Epsilon Serialisability techniques to allow conflicting transactions to execute simultaneously in way that their scheduling doesn't lead an imprecision that is higher than the one accepted in the data. The full model has been implemented in Java. The object-oriented model was chosen because, based on the previous research work, it is more suitable to model the complex real-world objects with time constraints than the relational model. After running one simulation, the results of the execution of real-time transactions configured in the configuration file, the percentage of energy consumed as well as the percentage of remained energy are shown. Moreover, the simulator is configurable and the protocols may be changed as required.

The fourth part of this research work, which was described in chapter 5 and 6, included the proposal of a new method to optimize the real-time query processing on WSNs for both latency and energy minimization and a new proposal of real-time query processing optimization for cloud-based wireless body area Networks (WBANs). The second main contribution of this thesis was accomplished by presented a new real-time query processing optimization for WSNs. According to the previous research work, the distributed approach allows performing in-network query processing that diminishes the data communication activities, which cause the most energy depletion in the network. In addition, it supports instant-queries and long-running queries processing, which are quasi-real-time queries processing. Therefore, this proposal combines statistical modeling techniques with the distributed approach to provide a new architecture and a query processing algorithm for optimizing the real-time user query processing for both latency and energy minimization with valid data. This valid data is stained of some uncertainty *(ε)* the user/application is willing to tolerate. In fact, the previous study reveled that in real-time systems, for some applications, the accuracy of results may be sacrificed under some limit to reduce the response time. Thus, Instead of periodically send the sensor readings to the database server for off-line processing or process the query directly into the network, the proposed hybrid approach uses statistical modeling techniques to perform a query processing based on admission control that uses the error tolerance and the confidence interval as the admission parameters to the system. A new concept of virtual network, composed by logical sensors which, in their turn, are composed by a probabilistic model and memory, is used to approximate in the gateway the answers of the query according to a given error tolerance and confidence interval. If the sensor data inside the virtual network is not sufficiently rich to answer the query, the admission controller routes the query towards the physical network. The experimental results based on real world as well as synthetic data sets show that the general proposed architecture provides, among other advantages, good individual query latency and valid data for real-time applications and energy-efficiency for WSNs.

The probabilistic model used in the previous research work is specific to the studied phenomenon, however the architecture and the query processing algorithm are general. Then, the last research work of this thesis is the extension and the adaption of the previous work to the area of wireless body area Networks (WBANs). A new architecture of cloud-based WBAN with its underlying query processing algorithm for secure and powerful storage, energy-efficient and real-time data processing is proposed. Instead of periodically sending the sensor readings from the WBAN to the cloud medical server for off-line processing what involves time delay and energy wasting, the proposed approach combines a cloud-based WBAN services and statistical modelling techniques to perform a query processing optimization that uses the error tolerance and the probabilistic confidence interval as query execution criterions. Thus, the probabilistic model uses the patient data information stored in the cloud medical server to approximate the answer of the query according to a given error tolerance and confidence interval. If the patient data information inside the medical database server is not sufficiently rich to answer the query, the query processor routes the query towards the physical WBAN considered fresher. The experimental results based on real as well as synthetic data sets show that the general proposed architecture provides, among other advantages, good individual query latency and valid data for real-time medical diagnosis and energy-efficiency processing of body sensor nodes.

2. Future Work

The design of a real-time databases model for WSNs is very complex because of its specificities. The first main contribution of this thesis consists of the proposal of a model for a simulation framework for performing validations of real-time databases with data acquired form WSN at early stages of software development. However, in a WSN the medium access and duty cycles of the nodes may impose a delay in data forwarding. Besides, this delay is distance sensitive; i.e. it increases in each hop of a multi-hop communication, yet this model is based on the distributed approach that involves in-network query processing. Therefore, although the simulator was mostly built to validate the applications with real-time databases protocols for WSN, it may be improved by adding the simulation of protocols or place part of this simulator on another existing simulator with acceptable characteristics of network protocols, layout, task scheduling by the operating system, etc. which can affect the precision of the simulator.

Using a graphical user interface (GUI) can facilitate and speed the establishment of the network topology and the composition of basic modules. It can also allow the quick visualization of the simulation results and help to trace and debug the simulation at real-time. Moreover by using a GUI, non-specialist users can get an easier control of the simulation. Therefore, this simulator may be revised and improved in order to add an appropriate graphical user interface.

The second major contribution of this thesis is the proposal of a new architecture and a query processing algorithm to optimize the real-time user query processing for WSNs in order to reduce the query latency and to save more energy. This approach combines the distributed approach and statistical modeling techniques to approximate query answers. The distributed approach involves in-network processing, which usually bases on other data reduction techniques such as data aggregation, packet merging, data compression techniques, data fusion, etc. However, this proposal does not profit from these above data reduction techniques. In fact, the approximation based techniques are useful in many applications which don't require the exact readings, resulting in the form of less energy consumption. However, in order to get more good results in terms of energy savings, both aggregation and approximation techniques could be used in combination with each others where possible. The WSNs are application specific, so one technique could not be always efficient for all kinds of applications. Thus, it is desirable to have a generalized database management system so that all various applications can customize and tailor it according to their needs.

Besides, correlation is an important aspect between sensors and their neighborhood. Therefore in future works, the improvement of the proposed architecture and its query mechanism in order to take into account the correlation between the sensor readings and eventually other parameters can be investigated. This will further improve the estimate of readings and the power saving.