

Multi-Hop Simulator for Ad-Hoc IEEE 802.11e

João M. Ferro, Fernando J. Velez, *Member, IAENG**

Abstract—This work addresses the development of a cross-layer wireless network simulator for IEEE 802.11e in a multi-hop environment. Based on a previous simulator created by our research team for physical plus MAC layers, it involved the upgrade of a previous version of that simulator to model the transmission of packets from the source to the destination using intermediate nodes. From the results obtained for an initial routing algorithm, one can conclude that the system is capable of deliver the packets regardless of the source/destination, and successfully calculate several metrics such as end-to-end delay and number of packets lost.

Keywords: IEEE 802.11e, simulator, multi-hop, Quality of Service, cross-layer

1 Introduction

Wireless networks are gaining more and more importance in our world. Cellular phones with GPRS/UMTS, Wi-Fi networks, and WiMAX networks are very common these days, and they share a common feature: they require some sort of backbone infrastructure in order to allow for packets from different communication peers reach each other. For example, if someone makes a phone call, the conversation will always pass from the cell phone to the operators' infrastructure, and then to the receivers phone, even if they are both in the same building. Resources would be certainly saved if somehow the cell phones could connect directly to each other.

In an ad-hoc network all the participants (also called nodes) can communicate directly with their neighbours. Two nodes are considered neighbours if their communication devices can reach each other. Nodes wanting to communicate to others that are not neighbours will simply send a message to another node which is located nearer the destination. As so, a centralised infrastructure to establish the connectivity is not required, since each node will determine by itself to where it should forward its data. The specification IEEE 802.11 refers to this type

of network as Independent Basic Service Set (IBSS).

The paper is organized as follows. Section 2 presents some background information about the previous simulator and about IEEE 802.11e. Section 3 describes the features of the new simulator highlighting the modifications we performed. Section 4 presents the results of the initial simulations, while Section 5 presents conclusions and suggestions for future work.

2 Previous work

This simulator is based on the one developed at the Instituto de Telecomunicações, Laboratório da Covilhã, as part of the IST-UNITE project. The purpose of that previous version was to create a trustworthy simulator to be used in the context of IST-UNITE, allowing the study of the interoperability among Wi-Fi and HSDPA. This involved the creation of an HSDPA time-driven simulator and a Wi-Fi event-driven simulator that may be either able to run separately or together, in tight cooperation for the coexistence scenario. In the latter case, there is communication between the two simulators which will run synchronously. The Wi-Fi simulator was initially built for the infrastructure mode, i.e., an architecture with one access point (AP) and several wireless stations (STAs) attached to it. It can simulate traffic from one STA to the AP and vice-versa, and calculates the throughput (total in the simulation and per unit of time), packet delay (total in the simulation and average), lost packets, retransmissions, and collisions. This computation is oriented to the support of the Quality of Service (QoS), i.e., it implements IEEE 802.11e with four access categories. As a consequence, these metrics are calculated for each traffic type: voice (VO), video (VI), background (BK) and best-effort (BE).

The IEEE 802.11e standard was conceived to improve 802.11 for Quality of Service (QoS) support. QoS refers to the way resources are reserved for certain applications, guaranteeing that the most urgent data flows will have higher priority levels. The IEEE 802.11 Medium Access and Control (MAC) coordination functions are the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). They establish the timing and sequence for each station to access the shared medium. The latter is only used in the infrastructure mode, with the AP acting as the coordinator. It provides some basic QoS support, but since it is defined as

*Manuscript received March 20, 2008. This work was partially funded by CROSSNET (a Portuguese Foundation for Science and Technology, FCT, POSC project with FEDER funding), MobileMAN, IST-UNITE, and "Projecto de Re-equipamento Científico" REEQ/1201/EEI/ 2005 (a FCT project). João Ferro acknowledges the PhD grant from FCT ref. SFRH/BD/36742/2007. Authors can be reached at Instituto de Telecomunicações, DEM-UBI, Calçada Fonte do Lameiro, 6201-001 Covilhã, Portugal. Tel/Fax: (+351) 275 329 953 Email: ferro@lx.it.pt, fjv@ubi.pt

optional and is more complex and costly to implement, only a few APs support it. The former does not present QoS guarantees at all. To ensure the QoS performance, IEEE 802.11e introduced the Hybrid Coordination Function (HCF), which defines the HCF Controlled Channel Access (HCCA) and Enhanced Distributed Channel Access (EDCA). In EDCA, a Transmission Opportunity (TXOP) is won according to the traffic type, scheduling firstly higher priority traffic, like voice and video. The HCCA is more complex, but allows greater precision in terms of QoS. All these enhancements are fully explained in [1].

In the EDCA simulator each station (and AP) has four buffers, one for each traffic type. By using an event-driven approach, for each new packet arrived at a machine a new event is generated. The simulator engine uses its internal clock to pick up the next event and pass the corresponding packet to the developed MAC+PHY layers. A more comprehensive description of lower layers of the simulator can be found in [2].

3 Overview

The current simulator is an evolution of the previous one. To provide support for multi-hop environment it was necessary to implement the routing algorithm above the already existing Medium Access Control (MAC) plus physical (PHY) layers, as it is presented in Fig. 1.

After the random placement of all the stations in the field, the simulator determines which can communicate directly. With this data, the selected routing algorithm determines the shortest path from a station to all the others. For these initial tests we chose the well-known Dijkstra’s algorithm [3], which calculates the least cost path from one node to all the remaining nodes. This characteristic allows the routing calculation to take place before the simulation starts, making the routing table available to all nodes since time instant 0s. Besides this initial calculation, the routing table can be accessed and modified at any time, allowing the use of cross-layer design (i.e., gather information from physical and MAC layers and use it in the routing algorithm) for optimisation purposes. In this paper, however, we present no cross-layer at all, and the routing table is not modified during the simulations.

The engine starts collecting events at the beginning of the simulation. When a packet arrives at a wireless machine, a new simulation module verifies if the packet has another destination by checking the two additional fields, now included in the packet header: besides the “payload”, “origin station”, and “destination station”, among others, each packet now includes the “initial origin” and the “final destination” fields. If it is found that the destination is another node, the routing table is accessed to determine the next hop, and the packet is added to the

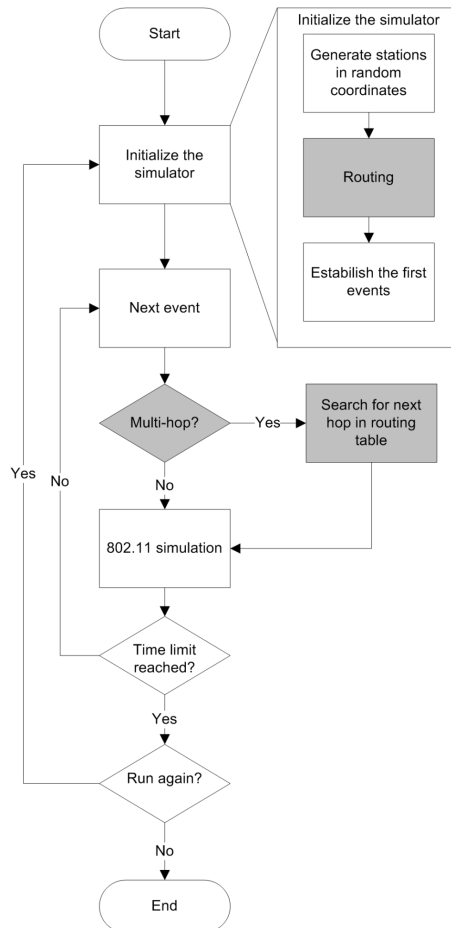


Figure 1: Simulator flowchart, modules added to the previous version are displayed in grey.

machine buffer with the new destination stamped.

The simulation will run until the time limit is reached, and will be repeated during a pre-determined number of times. The final output of the simulator is obtained as an average of the results for these simulations.

For the end user, this simulator allows for simulating a network with an unrestricted number of nodes, in any configuration type. By using the chosen routing protocol, it supports connections to every reachable station from its neighbours, allowing for a message to reach any destination. Stations are randomly placed in a field and even if a station is isolated the simulator will still run (unless this station is the source/final destination of a packet, which will terminate the simulation). Several parameters can be tuned by the user to its needs, some of them are presented in Table 1.

4 Results

In order to verify if the modified simulator was able to simulate a multi-hop environment we ran a few tests. In the first one we tried to verify if the simulation time was

Table 1: Some values that can be modified by the user.

Parameter	Default value	Description
Side	60m	Side of the square where the stations are placed
Range	35.365m	Transmission range of each station
Simulation Time	10000ms	Period of time to be simulated
Payload XX	- bits	Length of each packet of the type XX
Inter Arrival XX	- ms	Interval of time to generate a new packet of the type XX

going to affect the end-to-end delay. For this purpose we considered a fixed configuration of 14 stations with a video stream being generated from station 1 to station 3, Fig. 2.

The following parameters were used:

- Video traffic starts at: 1ms;
- New packet generated each: 10ms;
- Simulation ends at: 0.1/0.5/1/5/10/15s;
- Video payload size: 10240bits (each packet will be sent in 3 fragments).

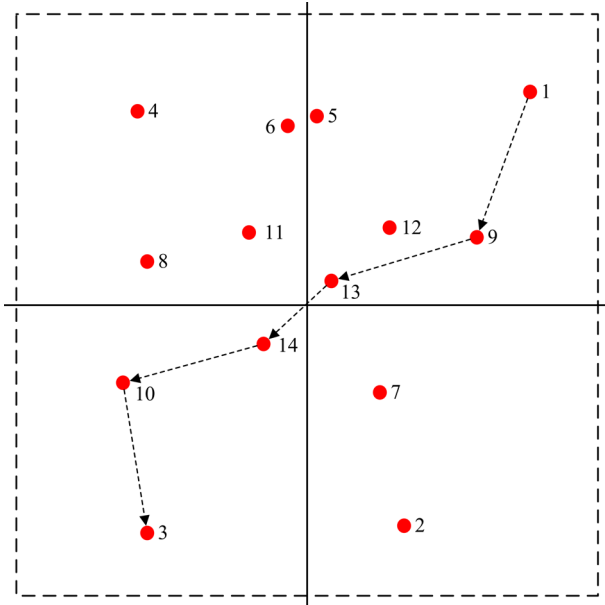


Figure 2: Video traffic being transmitted from station 1 to station 3.

Fig. 3 presents the average delay for each simulation time. As one can observe, the delay is constant with

the simulation time, despite the slight differences for the shortest simulations.

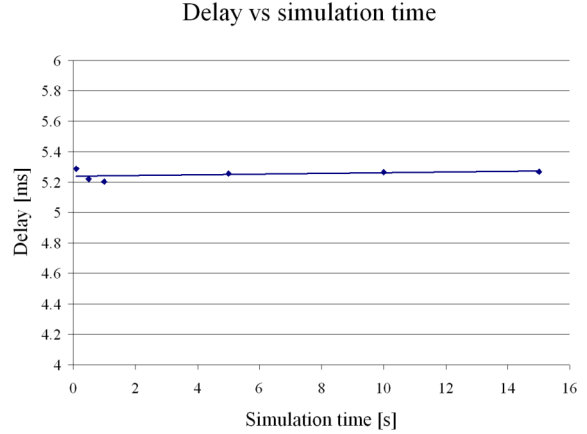


Figure 3: Delay versus simulation time.

After this initial test, 10s was assumed as a reasonable simulation time, and the subsequent tests were made with this value for the duration. Besides, further tests were made to evaluate the influence of multiple streams in the delay and packet loss. For these tests we used the same deployment of stations as in the previous test, Fig. 4.

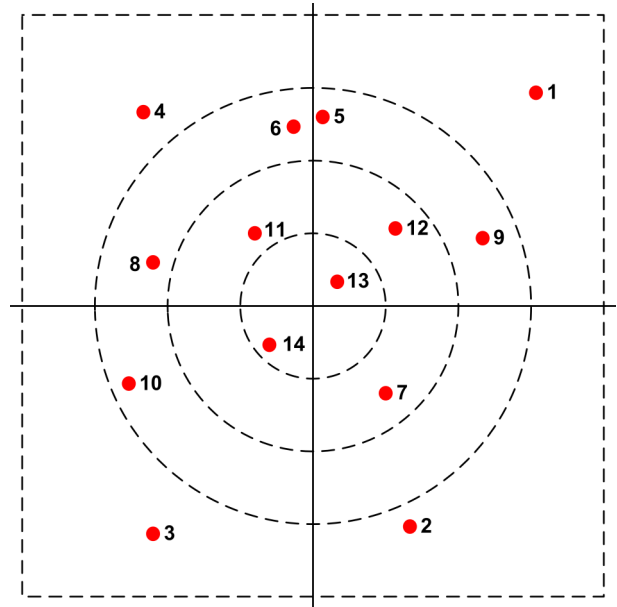


Figure 4: Station deployment for the tests.

The inner circle has a radius of 15 meters, the next one 30m, and the outer one 45m. The square has $120 \times 120 \text{m}^2$. In the next set of tests we added sequentially video streams between the following stations:

1. From station 1 to 3;
2. From station 4 to 2;

3. From station 5 to 6;
4. From station 8 to 9.

For each of these streams the routing algorithm established the following paths:

1. $1 \rightarrow 9 \rightarrow 13 \rightarrow 14 \rightarrow 10 \rightarrow 3$;
2. $4 \rightarrow 11 \rightarrow 14 \rightarrow 7 \rightarrow 2$;
3. $5 \rightarrow 6$;
4. $8 \rightarrow 14 \rightarrow 13 \rightarrow 9$.

Fig. 5 presents the results obtained for the delay of video stream from stations 1 to 3. By increasing the number of stations that generate packets, the end-to-end delay (latency) increases.

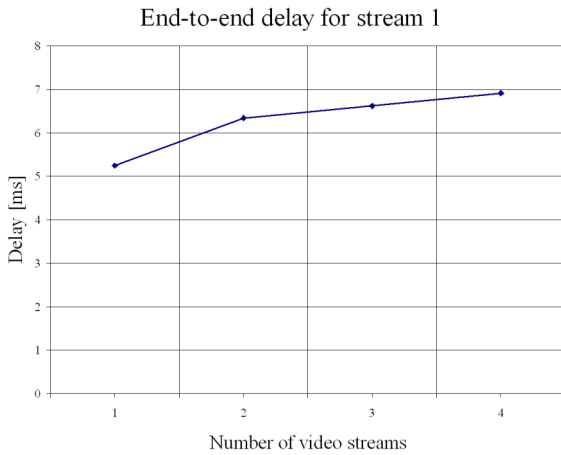


Figure 5: End-to-end delay for video stream 1.

Another interesting metric is the number of lost packets, which is presented in Fig. 6. When the number of packets to be transmitted increases there are more collisions and packet losses.

One interesting characteristic of this simulator is the possibility of simulating 4 different types of traffic and their mixtures. By using this feature we produced a new set of tests, keeping the same topology but modifying the type of packets in each stream:

1. Video stream from station 1 to 3;
2. Background traffic from station 4 to 2;
3. Voice traffic from station 5 to 6.

The details for the configuration of each traffic type are presented in Table 2, while the results for the number of

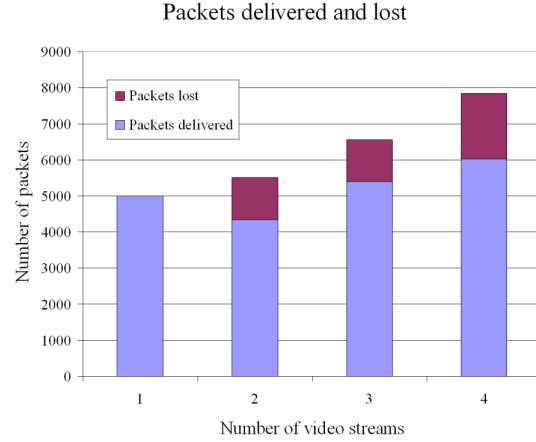


Figure 6: Total of packets received/lost.

Table 2: Configuration of each traffic type.

Type	Packet size [bits]	Period [ms]
Video	10240	10
Voice	1280	20 (bi-directional)
Background	18430	12.5

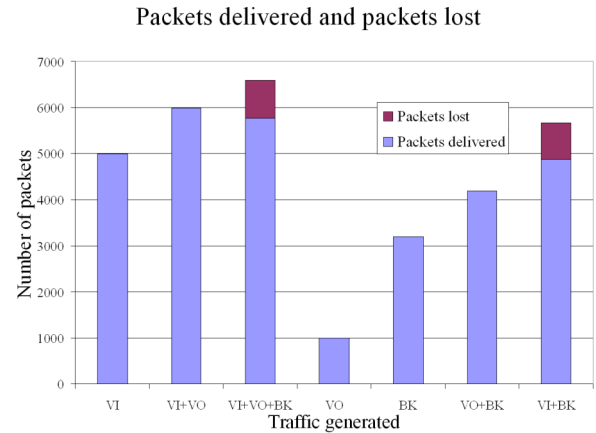


Figure 7: Total of packets received/lost.

packet successfully delivered or lost are presented in Figs. 7 and 8.

While Fig. 7 presents the results for each packet, Fig. 8 addresses each stream, i.e., the latter only counts a successful packet when this one arrives at the final destination, while the former presents results for the accounting of packets that arrive at any station. For this reason, and looking at the results for the video stream (VI) alone, in the former case, 5000 packets were successfully delivered, while for the latter only 1000 packets arrived at the final destination. Remember that this stream is transmitted in a 5-hop path, and that the most relevant result is the

latter.

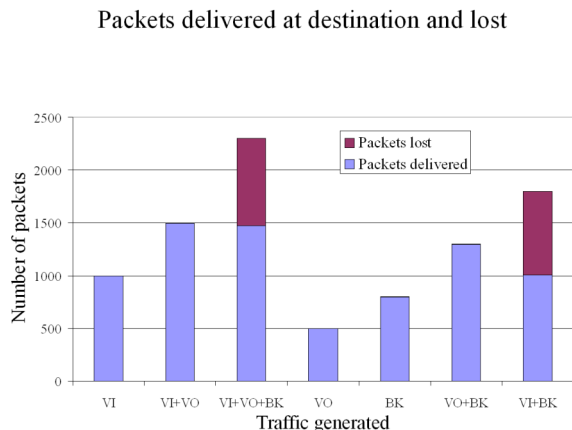


Figure 8: Total of packets lost/received at destination.

With only one stream, the system behaves perfectly, and all of the packets are correctly delivered. This happens because there are no collisions, since the period is longer than the end-to-end delay for every string. However, when more than one type of traffic is being generated simultaneously, the packets start to collide and some of them are considered lost (current policy establishes that a station will drop a packet after it experiences 2 collisions). The extension of this problem is larger for “background” traffic, which can be explained by the largest size of each packet, which will require the fragmentation into four smaller packets. For voice traffic, the delay is very low and it seems that it does not affect the remaining traffic at all. This is due to the small packets generated by this traffic type (no fragmentation required) and because these stations are in line-of-sight, so no multi-hop is necessary.

5 Conclusions

In this paper we presented a custom-made IEEE 802.11e simulator, with multi-hop capabilities. Based on a previous work that developed a MAC plus PHY layer simulator for this standard, we added above these layers a routing protocol in order to allow the simulation of traffic among stations that can not communicate directly. We performed some initial tests to verify if the simulator was performing as expected, and the results were encouraging. A few improvements for this simulator are being considered; after more validation tests a visual interface is being planned as well as the release of the code to the scientific community for research purposes.

References

[1] Prasad, A.R., Prasad, N.R., *802.11 WLANs and IP Networking*, Artech House, London, UK, 2005.

[2] Cabral, O., Segarra, A., Velez, F.J., “Event-Driven Simulation for IEEE 802.11e Optimization,” *IAENG International Journal of Computer Science*, V35, N1, Feb. 2008.

[3] Dijkstra, E.W., “A Note on Two Problems in Connexion with Graphs,” *Numerische Mathematik*, V1, pp. 269-271, 1959.