A New Approach to the Simulation of HIPERLAN Wireless Networks

G. I. Papadimitriou*, T. D. Lagkas*, M. S. Obaidat, and A. S. Pomportsis***

*Department of Informatics, Aristotle University
Box 888, 54124, Thessaloniki, Greece

**** Corresponding Author:**
M. S. Obaidat, Dept. of Computer Science
Monmouth University
West Long Branch, NJ 07764, USA
E-mail: obaidat@monmouth.edu

KEYWORDS
HIPERSIM, HIPERLAN, modeling and simulation, WLAN, wireless, sense range, performance evaluation.

ABSTRACT

This paper presents a new simulation mechanism that is used by the simulator "HIPERSIM," which was developed to examine the behavior of the HIPERLAN wireless networks. The ETSI HIPERLAN and the IEEE 802.11 are the most popular WLAN standards. The lack of HIPERLAN specialized simulators and the need for an accurate simulation engine have led to the development of HIPERSIM. The latter is a simulation environment for the HIPERLAN Type 1 networks which uses an exhaustive simulation engine in order to simulate accurately most of the important features of a HIPERLAN wireless network. Specifically, it fully simulates the complicated EY-NPMA MAC protocol of HIPERLAN, which is based on active signaling, hidden nodes, packet forwarding mechanism, power saving process, among others. A rather original characteristic of HIPERSIM is the fact that it distinguishes between the communication range and the sense range of a node. The main focus of this work is to provide a simulation mechanism appropriate for wireless networks, and especially for the HIPERLAN WLANs. The HIPERSIM results show that the HIPERLAN protocol is effective and suitable for the wireless environment. Probably there could be some improvement in order to avert the collisions close to the receiver.

1. INTRODUCTION

The increasing interest in wireless networks has led to the development of some standards that try to find their way in the market. This technology seems to be mature, but it still tries to meet the increasing demands and needs of new applications. WLAN standards demand continuous improvement in order to support QoS successfully. The recent applications for wireless networks are quite demanding, because they involve synchronized transmission (e.g. voice and video transmission) and reliability. Beside the need for improvement of the WLAN standards, it is necessary for the manufacturers to evaluate the different WLAN solutions that are offered today. Because of the above mentioned reasons, the need for suitable WLAN simulation tools has now arisen.

The two most popular WLAN standards are the 802.11 (IEEE) and the HIPERLAN (ETSI). Most of the researchers use general-purpose network simulators with the appropriate modules for the wireless topology. Some of them are forced to create their own simulation tools in order to analyze some limited features of a wireless network. Especially for the HIPERLAN networks, there are very few suitable simulation environments. This paper examines the existing WLAN simulation methods in general, and presents the HIPERSIM simulation environment. HIPERSIM is a simulator specialized in HIPERLAN networks simulation, it is fully parameterized, and it supports most of the features of HIPERLAN protocol and wireless topology. One of the differences between the simulation method that HIPERSIM uses and the classical simulation methods is the fact that HIPERSIM uses a time-based simulator in order to simulate accurately and in an exhaustive way the complicated Elimination Yield Non-pre-emptive Priority Multiple Access (EY-NPMA) MAC protocol of HIPERLAN. Also, HIPERSIM distinguishes between the communication range and the sense range of a node, which is rather original.

2. WLAN SIMULATION

The basic principles of a WLAN simulation do not differ significantly from the principles of a wired LAN simulation. So, the primary simulation entities (e.g. node, medium, buffer, and packet) remain the same, and the results mainly concern the network throughput and the average packet delay under various conditions. However, the wireless nature of a WLAN has some special characteristics that need extra analysis and emphasis.

In a WLAN, it is usually assumed that the communication medium, that is the air, is common for all the nodes, like a bus-network. The most simulation environments adopt this assumption. However, this is not completely accurate. In a wired bus-network, all the nodes that are connected to the cable have the same view

Proceedings 15th European Simulation Symposium
Alexander Verbraeck, Vlatka Hlupic (Eds.)

ISBN 3-936150-28-1 (book) / 3-936150-29-X (CD)

of the medium status. This is not happening in a wireless network. In a WLAN, every node has its own view of the medium status, which depends on the range of its antenna. When two nodes of a WLAN are not able to communicate directly with each other, then they are called hidden nodes. The "hidden nodes" issue is very important for the WLAN operation that is why most of the simulation environments take into account the hidden nodes. HIPERSIM goes a step further by distinguishing between the communication range and the sense range of a node.

Specifically, a pair of hidden nodes can be either in sense range or not in the HIPERSIM simulation environment. This issue is analyzed later on. Also, when simulating a WLAN, the packet forwarding mechanism of the specific protocol affects significantly the overall network performance, so it deserves further analysis. Another feature that is simulated by HIPERSIM is the power saving mechanism. The nodes of a WLAN, like HIPERLAN, are usually mobile devices with a limited battery life. Since the antenna of a node consumes a great amount of energy when transmitting or receiving, the simulation of the power saving mechanism provided by the protocol is quite interesting.

Most of the network simulators, like OPNET, are event-based. This means that the network operation is divided into a number of discrete events. There is an event list where all the generated events are stored. When an event is processed, new events are generated and they are added to the list. Then, the event that takes place earlier than the others is selected from the list, and the simulation clock is updated. The simulation stops when the simulation time is over or another termination condition is true. In Figure 1, the flowchart of a typical event-based simulator is shown (Obaidat et al. 2003; Nicopolitidis et al. 2003; Sadiku and Ilyas 1995).

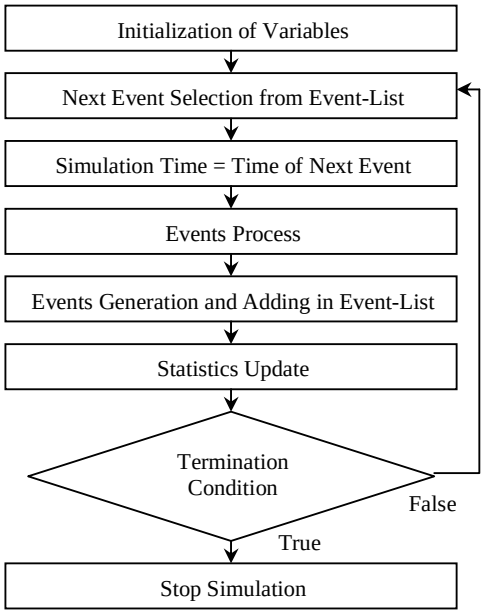

Figure 1. Typical Event-based Simulator Flowchart

The MAC protocols of the IEEE 802.11b and the HIPERLAN Type 1 standards are based on the classic carrier sense multiple access (CSMA) protocol, since a node senses the carrier before attempting to transmit. However, these WLAN MAC protocols have a lot of enhancements in order to be suitable for the wireless environment, thus they differ significantly from the classic CSMA that is used in the wired LANs. The event-based simulation models are the most popular schemes, but their great disadvantage is the fact that they demand simplifications of the system operation in order to distinguish discrete events. HIPERSIM, on the other hand, implements an exhaustive time-based simulation model in order to simulate accurately the complicated EY-NPMA MAC protocol of the HIPERLAN standard. Also, this simulation method avoids simplifications that concern the wireless topology, like the assumption that all nodes have the same view of the carrier status. In Figure 2, the flowchart of a typical time-based simulation model is presented (Obaidat et al. 2003; Nicopolitidis et al. 2003; Sadiku and Ilyas 1995).

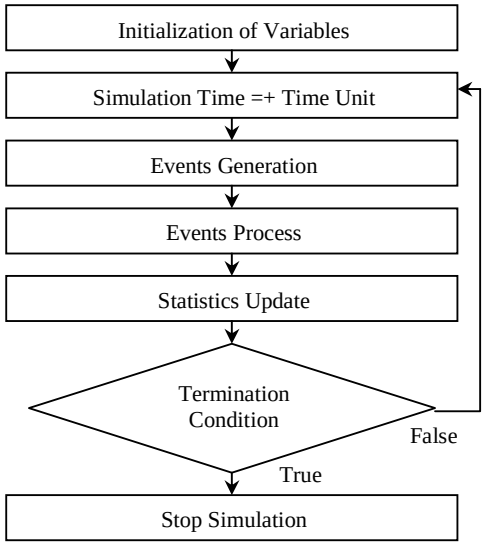

Figure 2. Typical Time-Based Simulator Flowchart

3. THE HIPERLAN NETWORK

HIPERLAN (HIgh PErformance Radio Local Area Network) is a standard for wireless LANs that was defined by the European Telecommunications Standards Institute [ETSI]. HIPERLAN is a rather short range WLAN (~50m), it supports slow moving stations (1.4 m/s), and can be of infrastructure or ad hoc type. Its high transmission rate is 23.529Mbps, and it operates at the 5GHz band.

An important characteristic of HIPERLAN is that it can support a variety of services. It combines asynchronous communication, such as file transfer, with time bounded communication, such as voice and video transmission. This is due to the fact that the Medium Access Control protocol supports Quality of Service (QoS), aided by the Channel Access Mechanism (CAM) that assigns the packet priorities.

3.1. The Priority Mechanism

The QoS support is based on two mechanisms: the user priority assigned to a packet and the lifetime of the latter. The user application sets the user priority of a packet at the value low or high. The value of the user priority and the packet lifetime give the CAM priority. In Figure 3, the CAM priority values are shown according to the user priority and the residual lifetime. The highest CAM priority is the one with the smallest value (ETSI 1998; Jacquet et al. 1996b; FU et al. 1996).

Notation:
ML: MPDU Lifetime
RML: Residual MPDU Lifetime
UP: User Priority (low = 1 , high = 0)

UP ↓	RML (msec) →	< 10	10 - 20	20 - 40	40 - 80	> 80
0		0	1	2	3	4
1		1	2	3	4	4

Figure 3. The CAM Priority Assigning

The Earliest Deadline First (EDF) is the queuing discipline of the packet buffer. This method selects the packet that needs to be sent earlier than the others, according to its CAM priority, residual lifetime and user priority. An example of this packet selection method is shown in Figure 4.

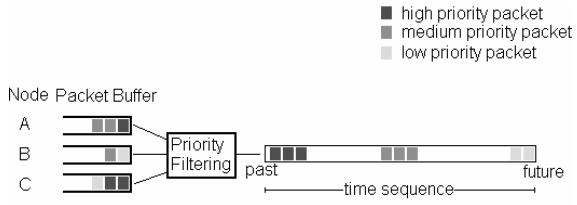

Figure 4. The Priority Scheme

HIPERSIM simulates the priority mechanism of HIPERLAN. Every packet carries a residual lifetime, a user priority and a CAM priority. These attributes get updated when the packet buffer is managed. The HIPERSIM results show the efficiency of the EDF method in contrast to the classic FIFO (ETSI 1998; Jacquet et al. 1996b).

3.2. The EY-NPMA Medium Access Protocol

The medium access protocol (MAC) that is used in HIPERLAN is the Elimination Yield Non-pre-emptive Priority Multiple Access (EY-NPMA) protocol. EY-NPMA is based on active signaling and it is suitable for wireless local area networks, like HIPERLAN. The medium access mechanism is based partially on the well known CSMA, since the node eventually "listens" to the medium to find out if it can transmit or not. However, the main mechanism differs from CSMA, since every node contests for the medium access in an active way by transmitting some special signals (Jacquet et al. 1996a). Below, the operation of the EY-NPMA protocol is presented briefly, according to the "ETSI functional specification EN 300 652 v1.2.1" (ETSI 1998).

A node that wants to transmit initially senses the channel. If it is idle, it enters the "channel free condition" and eventually transmits the packet. If the channel is not idle, then the node waits to synchronize with the others nodes at the end of the current transmission, so it enters the "synchronized channel condition." The synchronized channel access cycle consists of three phases, which define the access pattern for the competitive nodes (ETSI 1998; Jacquet et al. 1996a; Chevrel et al. 1996; LaMaire et al. 1996).

The first phase is the "Prioritization Phase" where the nodes carrying the highest CAM priority win and make the rest defer. In Figure 5, it is shown how node B "wins" during the Prioritization Phase and makes node A defer. CAM priority of node A packet is 2, while CAM priority of node B packet is 1 (higher priority).

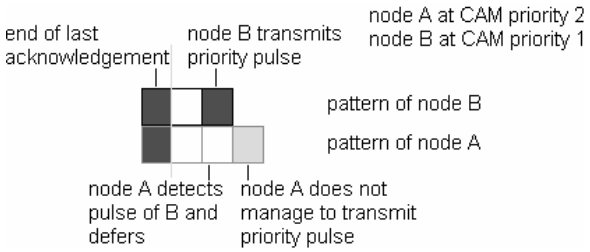

Figure 5. The Priorization Phase

The nodes that "survive" the Prioritization Phase proceed to the Elimination Phase. During the Elimination Phase, a great percentage of the contending nodes is eliminated, but at least one of them survives. This takes place in a random manner and according to a geometric distribution of probability p = ½. In Figure 6, it is shown how node B "wins" during the Elimination Phase and makes node A defer. Node A transmits an elimination pulse 1 slot long, while node B transmits an elimination pulse 2 slots long.

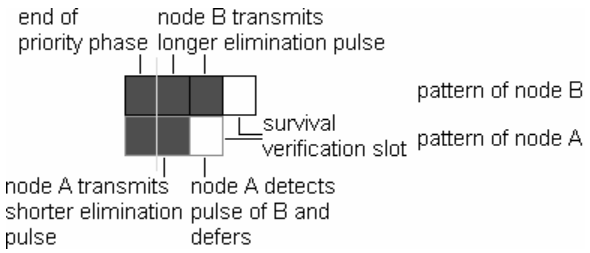

Figure 6. The Elimination Phase

The Yield Phase is the last phase before the transmission of a data packet and it is the last effort to reduce the number of the contending nodes. The nodes "win" randomly and according to a uniform distribution. If more than one node survive this phase, they will start transmitting simultaneously and a collision will take place. In Figure 7, it is shown how node B "wins" during the Yield Phase and makes node A defer. Node A allows 3 idle slots, while node B allows 2 idle slots.

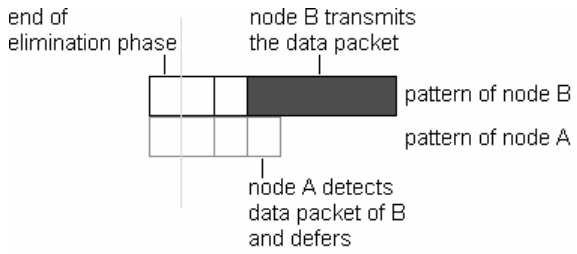

Figure 7. The Yield Phase

In Figure 8, an overview of the EY-NPMA protocol is shown.

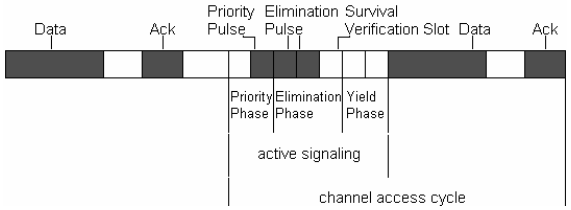

Figure 8. Overview of the EY-NPMA Protocol

4. THE HIPERSIM SIMULATION ENVIRONMENT

4.1. The HIPERSIM Features

4.1.1. Communication Range Issues

In a WLAN, not all of the nodes are expected to be able to communicate directly with each other. This is due to the fact that the antenna of a node has a limited range, and the obstacles that might intercept the communication. Specifically, the range of HIPERLAN is approximately 50 meters when the nodes are moving slower than 1.4m/s.

A special characteristic of the wireless environment is the fact that when two nodes are not able to communicate directly, they might be able to sense the signal transmitted by each other. More specifically, in a wireless network, like HIPERLAN, two nodes are able to exchange data packets when they are able to detect the transmitted signal and identify the bits sent. This happens when the signal attenuation due to distance and the signal fading are not intense enough to make the communication between the two nodes impossible. In this case, the two nodes are assumed to be in communication range. When the quality of the received signal is not good enough to allow data transmission, but it can still be detected, then the two nodes are assumed to be in sense range (signal detection range). Obviously, the sense range is greater than the communication range, since the former includes the latter, so it is likely that in a WLAN two nodes can detect each other even when they are not able to communicate directly. A special feature of HIPERSIM is that it distinguishes between these two kinds of ranges, when it simulates the hidden nodes (Wilkinson b).

The hidden nodes are pairs of nodes where one of them cannot receive data from the other, because of the distance or the obstacles between them. They cause overall reduction of the system performance. The range issues mentioned before are related to the "hidden nodes." We assume that two nodes are hidden when they are out of communication range. Two hidden nodes might be either in sense range or not. The simulation results show that a great reduction of the network performance is caused mainly by the hidden nodes that are out of sense range. That is the reason why it is important to make the distinction between the communication range and the sense range. In Figure 9, we can see three nodes of a HIPERLAN network. Every C_i area represents the communication range (data reception range) for node i and every S_i area represents the signal detection range (sense range) for node i. As we can see, node B is able to send and receive data from nodes A and C, while nodes A and C are hidden from each other, that is to say they are not in communication range. All three nodes are able to detect the signal of the others. In general, EY-NPMA takes advantage of the fact that two hidden nodes in sense range are able to detect the transmitted signal (FU et al. 1996; Weinmiller et al.; Moh et al. 1998).

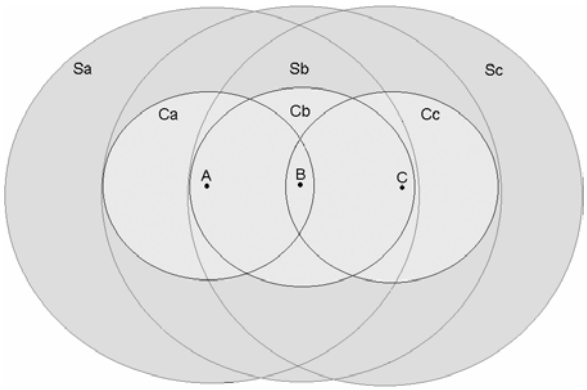

Figure 9. Representation of the Communication Range (C_i) and the Sense Range (S_i)

The simulation engine of HIPERSIM initially defines the pairs of the hidden nodes of the network. Then, it defines the hidden pairs that are out of sense range and those that are in sense range. The object that represents the carrier carries the information about all the transmitting nodes. The nodes that check the carrier can find out which signal they are able to detect, and whether they can identify the

bits sent. If the transmitter is not hidden, then the node can detect the signal and identify the bits sent (communication range). If the transmitter is hidden but in sense range, then the node can just detect the transmitted signal, but it cannot establish a direct communication. Lastly, if the transmitter is hidden and out of sense range, then the node that checks the carrier cannot even detect the signal.

4.1.2. Packet Forwarding

There are two types of nodes in HIPERLAN that are simulated in HIPERSIM: the forwarders and the non-forwarders. The non-forwarders know only their direct neighbors; nodes which are in communication range. On the other hand, the forwarders are aware of the network topology. In case a non-forwarder wants to transmit a packet to a node that is not in communication range (hidden node), it sends it to a forwarder by setting the latter as the intermediate node of transmission. Packet forwarding in HIPERLAN is based on a table-driven routing protocol. This forwarding mechanism increases the system complexity since it requires the continuous watch of the network topology, which changes dynamically. In Figure 10, three nodes (A, B, C) of a HIPERLAN network are represented, where nodes A and B are in communication range, nodes B and C are in communication range, while nodes A and C cannot communicate directly with each other. Nodes A and C constitute a hidden pair, while node B is a forwarder. This means that it can forward packets from node A to B and vice-versa. The packet forwarding mechanism is simulated in HIPERSIM. Initially, the forwarders are defined. Both the forwarders and the non-forwarders are represented by the "Node" class, where there is a property that defines whether the specific node is a forwarder or not. So, in HIPERSIM, the forwarders and the non-forwarders have the same structure, except from the fact that a forwarder can communicate directly with any other

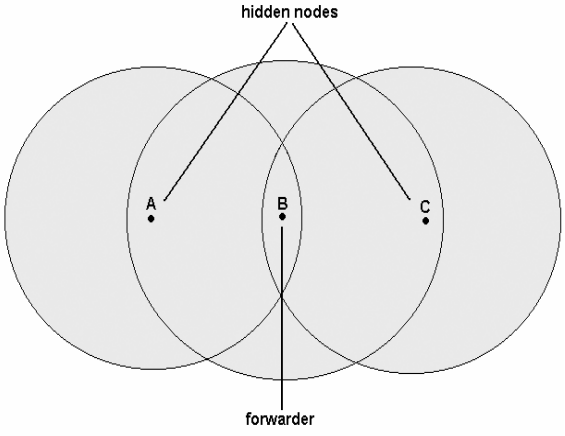

Figure 10. Representation of the Packet Forwarding

4.1.3 Power Saving

The mobile devices that constitute a WLAN have limited energy autonomy. When an antenna transmits or receives a signal, it consumes a great amount of energy. Therefore, a possible solution to the power problem is to turn the antenna off when there is no data transmission or node of the network and it works as an intermediate node that forwards packets between the hidden nodes. All the nodes are aware of the forwarders (ETSI 1998; Weinmiller et al.; Zeng 2000).

reception. In HIPERLAN, some nodes are P_Savers, while some others are P_Supporters. P_Savers are set at status "OFF" for specific time intervals. During these time intervals, they are not able to receive data packets. P_Supporters have the responsibility to collect data packets that have as destination a P_Saver which is "OFF." When the P_Saver returns to normal operation status, the P_Supporters forward the packets to it. Power saving is optional and it is not fully defined by the HIPERLAN protocol. The HIPERSIM simulates the power saving mechanism and shows that it has satisfactory results. Initially, the P_Savers and the P_Supporters are defined. These nodes have the same structure with any other node of the network. The difference is that during the simulation the P_Savers are turned off from time to time, and the P_Supporters collect the packets and send them to the P_Savers when they are back on (ETSI 1998; Zeng 2000)

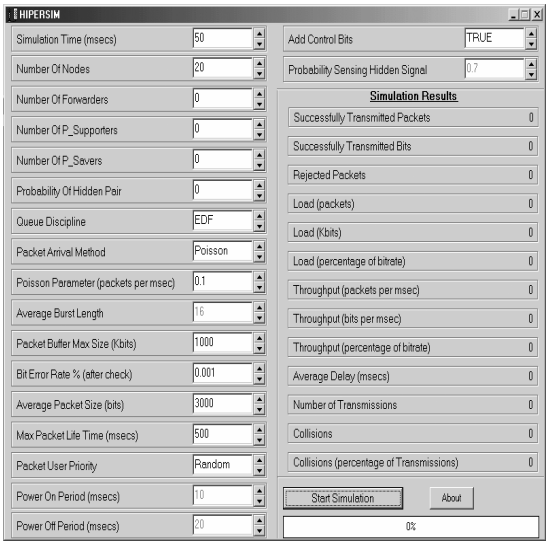

Figure 11. The Graphical User Interface of HIPERSIM

4.2. The HIPERSIM Environment

4.2.1. Environment Description

Figure 11 shows the environment of HIPERSIM. The user is able to set the simulation parameters, start the simulation, watch the progress bar and finally get the

results that are shown on the right side of the HIPERSIM window. The results of every simulation experiment and the current values of the parameters are automatically saved in a database file by using the ADO mechanism. Every record stores the values of the parameters and the results of the corresponding simulation. HIPERSIM was developed in C++, and it is a W32 application which uses the respective Graphical User Interface (GUI).

4.2.2. Code Structure

The implemented classes in the simulation mechanism are: "Packet", "Carrier" and "Node." The class "Packet" represents the common packet in the network. Whatever transmitted in HIPERLAN is a "Packet" object. Among the properties of the "Packet" are size, generation time, sender's ID, receiver's ID, lifetime, priority etc. When there is an intermediate node, this might be either a forwarder or a P_Supporter. The basic methods are the "GetRML" that returns the residual packet lifetime (Residual ML), and the "GetCAM_Priority," which returns the CAM priority of the packet. In Figure 12, the structure of the class "Packet" is presented.

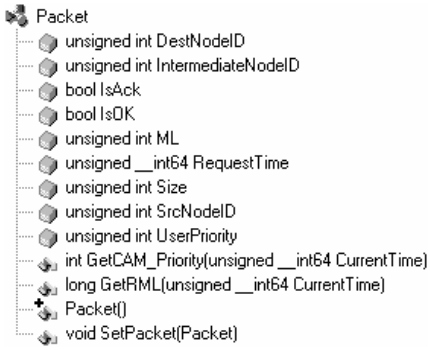

Figure 12. The Class "Packet"

The class "Carrier" represents the communication medium of the network. In general, the class "Carrier," which produces a single object, "stores" current transmitted packet, status of the medium (idle or data transmitting or in collision et al), nodes transmitting the current moment, and time that the current transmission will be completed. Every node checks the carrier to find out if there is a transmission for it. More than one node might transmit simultaneously during the simulation. These nodes might be in the communication range or the sense range or out of the sense range of the "listener" node. The basic methods are the "PutPacketOnCarrier" which "puts" a packet on the medium, and the "AddNodeInTransmittingInfo," which adds the node that starts transmitting in the list of the nodes that are currently transmitting. A view of the class "Carrier" is shown in Figure 13.

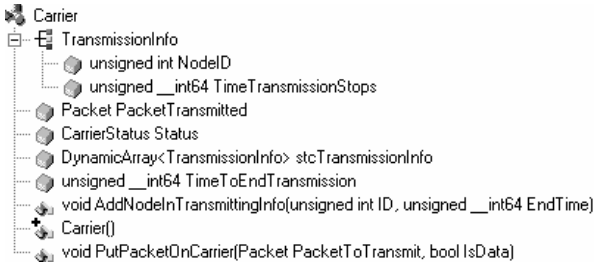

Figure 13. The Class "Carrier"

The class "Node" is definitely the most significant class and it implements most of the operations of the simulator. Every node is represented by an object of the class "Node" and carries a unique ID. There are a large number of properties and methods in the class "Node." The most important properties are the ID of the node, type of node (Simple or Forwarder or P_Supporter or P_Saver), list of nodes that are hidden from it, list of the hidden nodes that it can sense, Packet Buffer, packet to be sent, and a large number of variables that are time indicators and which help to implement the medium access protocol of HIPERLAN. The methods of the class "Node" constitute the "heart" of the simulation mechanism. Below is a list and brief description of each method.

-AddPacketInBuffer: It adds a packet in the buffer.

-ChangeP_Status: It concerns only the P_Savers. It checks the conditions and changes the status of the node from On to Off and vice-versa.

-ChannelAccess: This is the basic function that implements the MAC protocol of HIPERLAN, the "EY-NPMA." When a node has a packet to transmit, it uses this method to gain access to the channel.

-CheckExpiredPackets: It checks the packets in the buffer to discover packets whose lifetimes have expired. After that, it rejects them.

-CheckNode: This is the method that is called to check the node state and decide which actions must be executed by calling other methods.

-GetCurrentBufferSize: It returns the current size of the packet buffer in bits. It is used to find out if there is enough space in the buffer to add a packet.

-InternalPacketArrival: It implements the generation of a new packet inside the node. It decides the size of the packet, destination, lifetime etc. .

-IsNodeHidden: The Boolean returned result shows if the node, which is the argument of the method, is hidden from the node that calls the method.

-IsTransmissionListened: If the node that is calling this method can detect the current transmission, then the method returns the time that this transmission is completed.

-NextArrival_Burst: It computes the time that the next packet generation takes place, when the selected packet arrival method is the "Bursting."

-NextArrival_Poisson: It computes the time that the next packet generation takes place according to the Poisson distribution.

-ReceiveData: This method receives the transmitted data packet which is destinated to the calling node. If the calling node is an intermediate (Forwarder or P_Supporter) for this packet, then the packet is added to the buffer so that it is sent later to its final destination.
-RetransmitData: After the transmission of a data packet and the non-reception of the corresponding acknowledgement, this method is called in order to reschedule the transmission of the packet that was not acknowledged.
-SelectPacketToSend: The packet to be sent is selected from the buffer. The selection is made using the "EDF" or the "FIFO" method, according to the user's choice.
-SendAck: It implements the acknowledgement sending after the successful reception of a data packet.
-SendData: This method is responsible for the transmission of a data packet. If it is necessary, an intermediate node for this transmission is set.
In Figure 14, the whole structure of the class "Node" with its properties and methods is presented.

Figure 14. The Class "Node"

4.2.3. Operation Sequence

Here we describe briefly the sequence of the actions that take place during the simulation. First of all, we must make it clear that every simulation experiment is independent from others since the simulation mechanism is initiated every time.

According to the parameters set, packets are generated inside the nodes, and if there is enough space they are stored in the buffer. The destination of every packet is another node of the LAN. The source node tries to gain access to the channel, according to EY-NPMA, in order to send the packet selected from the buffer. In an exhaustive way, the nodes are checked one by one every time unit. The simulation clock step or time unit is equal to the high rate bit-period, which is the time needed to transmit a bit in high transmission rate (23.529 Mbps). The simulation is terminated when the simulation time is over and the results are calculated and stored.

In the beginning of every simulation, all the nodes are initiated as elements of a dynamic list of "Node" objects. The carrier, which is represented by the single object of the class "Carrier" (objCarrier), is also initiated. Next, the "forwarders" and the "P_Supporters" are selected, provided that the user has made the corresponding choices. After that, the P_Savers, the pairs of hidden nodes and the hidden nodes that are in the sense range are selected. As it was mention earlier, a special feature of HIPERSIM is that it distinguishes hidden nodes from those that are in the sense range and those that are out of sense range. In the next section, the HIPERSIM results show that hidden nodes that are out of sense range affect significantly the HIPERLAN performance. The time the first packet generation takes place is calculated for every node. Afterwards, the application enters the main loop of the simulation, which checks every node at every time moment by using the method "CheckNode."

The function "CheckNode" decides whether some other methods of the class "Node" will be called. Initially, it checks if there is a packet arrival from another node, so as to call the function "ReceiveData." Then, it checks if a data packet retransmission is needed, and if it does, it calls the function "RetransmitData." After that, the function "ChannelAccess" is called. If there is no packet to be sent at the current moment, the "ChannelAccess" function performs no action. Then, the function "CheckNode" checks if an acknowledgement must be sent, so as to call the function "SendAck." Afterwards, if it is time to generate a new data packet, the function "InternalPacketArrival" is called. Lastly, the function "CheckNode" checks if the node status must change from On to Off and vice-versa by using the function "ChangeP_Status." It is obvious that the latter function can be called only when the calling node is a P_Saver.

When the simulation time is over and the simulation is completed, the results are calculated and presented. A new record is created in the table of the "results.mdb," and the values of all the parameters and results are recorded there. In order to get these results, some special variables are inserted in different places in the code and their values are updated during the simulation (statistics update). There are also some global, assistant functions, which are called when needed. For example, one of these

functions is the msecs conversion to high rate bit-periods (1 msec = 23529 high rate bit-periods). The flowchart of the simulation mechanism is shown in Figure 15.

4.3. HIPERSIM Simulation Results

This section analyses the HIPERSIM results of the HIPERLAN under various conditions.

In every simulation experiment, the packet generation inside the nodes is Poissonian. The packet size is variable and exponential. Its default average value is 3000 bits, without counting in the added control bits. The packet lifetime is randomly selected between 10 and 800 msecs

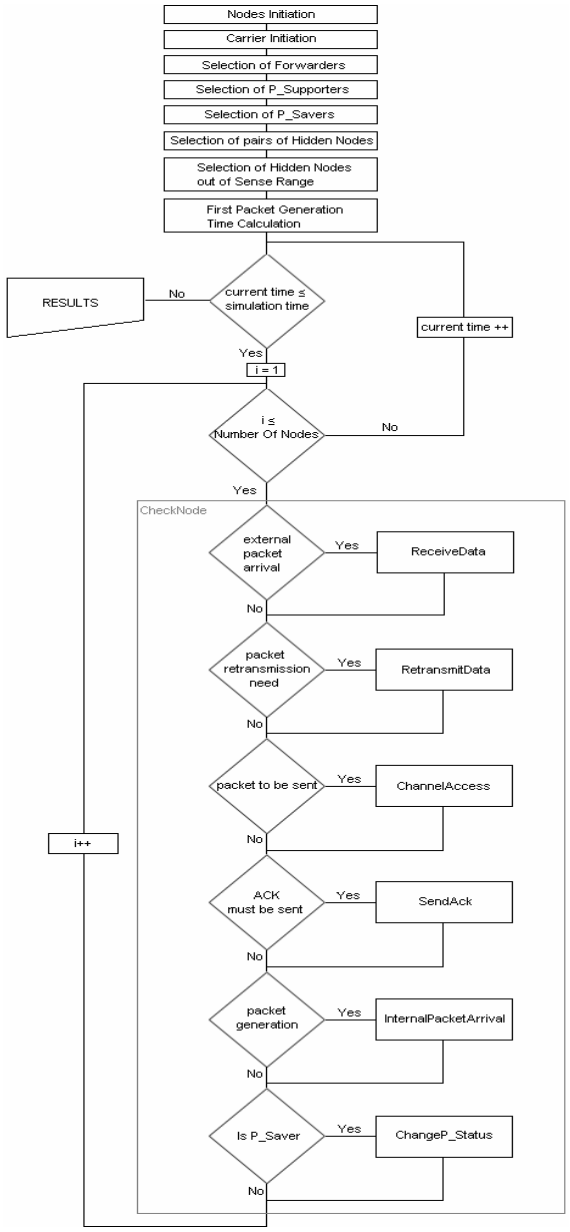

Figure 15. Flowchart of the HIPERSIM Mechanism

Initially, we examine the system throughput depending on the number of nodes of the WLAN. In Figure 16, we notice that the throughput value remains almost stable when the number of nodes increases. This satisfactory performance is due to the fact that EY-NPMA deters the great increase of the collisions even when the number of nodes increases significantly.

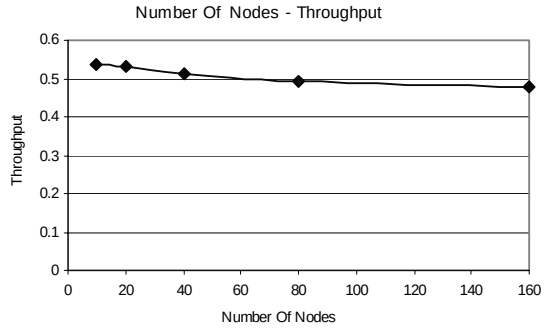

Figure 16. System Throughput versus Number of Nodes

Likewise, the average packet delay, that is the average time the packet remains in the system from its generation time till its successful transmission and reception, is almost stable while the number of nodes increases; see Figure 17.

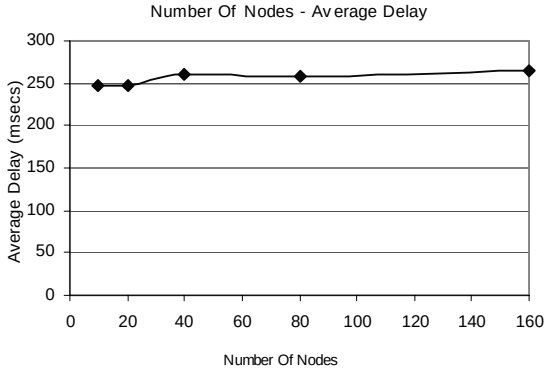

Figure 17. Average Delay versus Number of Nodes

Figure 18 presents the average packet delay versus the Poisson parameter which concerns the packet generation in every node every msec. Actually, the increase of the value of the Poisson parameter causes the increase of the system load, since the packet generation rate increases. In this graph, we examine the performance of the Earliest Deadline First (EDF) buffer discipline in comparison to the traditional First In First Out (FIFO). As it was expected, the average delay is greater when FIFO is used, especially when the load increases. HIPERSIM can simulate both the EDF and the FIFO queue disciplines.

We examine the way that the average packet size affects the throughput, assuming two different values for the Bit Error Rate (after check). The “bit error rate” is the rate a bit error occurs, after the application of all the predefined error detection and error correction checks (that is why

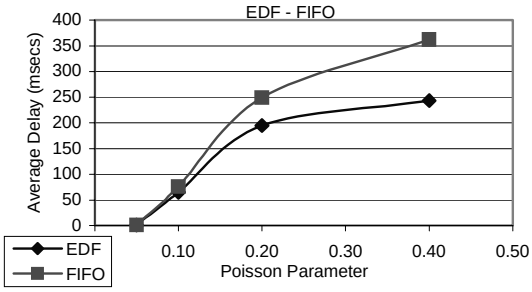

Figure 18. Comparison between the EDF(and the FIFO Queuing Discipline

we assume so low bit error rates and we add the "after check" characterization). According to the value of the bit error rate and current packet length, HIPERSIM decides whether a bit error occurs during a packet transmission. Specifically, this decision is made when a packet is "put" on the carrier using the method "PutPacketOnCarrier" of the object "objCarrier." In Figure 19, we see that the increase of the average packet size initially causes some increase in the throughput (because of the smaller overhead), which eventually stabilizes. But when we increase the bit error rate, the increase of the average packet size eventually causes a decrement in the throughput.

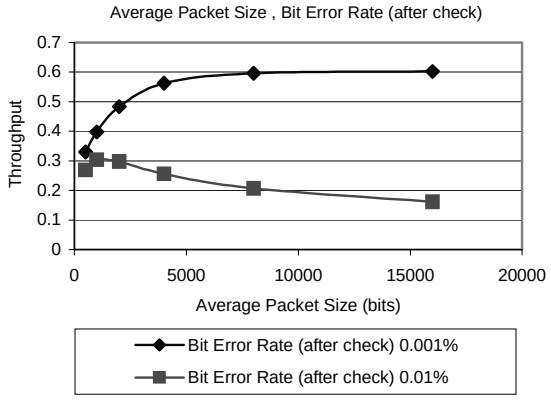

Figure 19. System Throughput as a function of the Average Packet Size and the Bit Error Rate

Another issue is the WLAN performance when there are nodes that operate as P_Savers. It is found that when the number of the P_Savers increases, the system throughput decreases. HIPERSIM simulates the "ON" and "OFF" P_Saver periods. In Figure 20, the simulation results are presented for different numbers of P_Savers, when the total number of nodes is 30 and the number of P_Supporters is 2. Every P_Saver can randomly use any P_Supporter.

HIPERSIM simulates active signaling, a feature of EY-NPMA, according to which, a node that can just sense the signal of another node (sense range) is able to participate

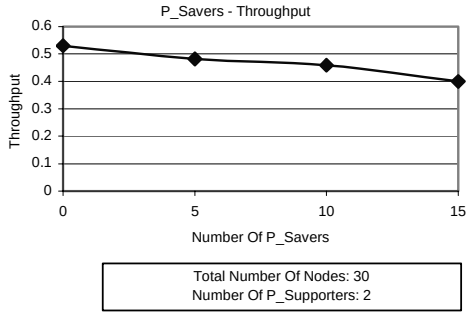

Figure 20. System Throughput versus Number of P_Savers

successfully with the latter in the synchronized channel access cycle. In case there are two hidden nodes that cannot even detect the signal of one another (they are not in sense range), there is a high possibility that collisions may occur. This would lead to a significant performance degradation. The reason for this performance reduction is the fact that these two nodes would not be able to synchronize directly since each one would have a different view of the channel status. In HIPERSIM, every node has its own view of the channel status, depending on the network topology.

In Figures 21 and 22, we use the term "Probability of Sensing Hidden Signal." This is the probability that two hidden nodes are in sense range. The "Probability of Hidden Pair" is the probability that two nodes are hidden from each other. As we can see in Figures 21, when the Probability of Hidden Pair increases, the network throughput decreases. If the Probability of Sensing Hidden Signal is low (less than 0.5), the system performance can be characterized as unacceptable. But when the value of the Probability of Sensing Hidden Signal is high and close to one, then the throughput is sufficiently high. The same behavior can be seen in Figures 22, by studying collisions rate. As it was expected, the Percentage of Collisions is high when the Probability of Hidden Pair is high. When the Probability of Sensing Hidden Signal increases, the collisions rate decreases.

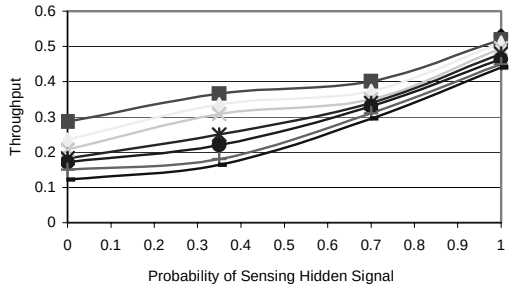

Figure 21. Throughput as a function of the "Probability Of Sensing Hidden Signal" and the "Probability Of Hidden Pair"

5. CONCLUSIONS

Simulation of a wireless local area networks (WLANs) has some special features that are different from that for

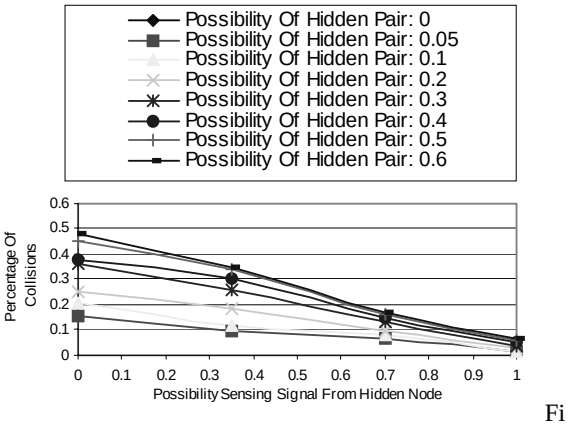

Figure 22. Collisions as a function of the "Probability Of Sensing Hidden Signal" and the "Probability Of Hidden Pair"

wired networks. The simulation environments that are used for the traditional wired LANs might be inappropriate for the WLANs. Specifically, a WLAN protocol, like HIPERLAN, has a complicated MAC protocol, which differs significantly from the classical CSMA that is used in most wired LANs. The "hidden nodes" problem is another challenge of the wireless networks which needs extra analysis. HIPERSIM is a simulation environment for the HIPERLAN wireless networks, and it simulates most of the special features of the wireless environment. It is fully parameterized and able to test the behavior of HIPERLAN networks under various conditions and operating environments. The simulation mechanism of HIPERSIM is rather original in the fact that it distinguishes between the communication range and the sense range of a node. Specifically, this work assumes that the communication range is the area where the signal can be detected and the transmitted bits can be identified, while the sense range is the area where the transmitted signal can just be detected. The simulator works in an exhaustive way in order to be accurate. The code structure is object oriented and it is developed on the W32 platform.

The HIPERSIM results have shown that HIPERLAN is an efficient WLAN standard. Probably, some improvement of the protocol is necessary, so that the frequent collisions between nodes that are out of the sense range are avoided. Basically, the "hidden nodes" problem concerns the collisions that take place close to the receiver and not in the sender's region. So it would be efficient if there were a mechanism that informed the neighbors of the receiver about the oncoming transmission. In that case, the receiver's neighbors would not collide, so the overall system performance would improve.

REFERENCES

Anastasi, G.; L. Lenzini; and E. Mingozzi, "*Stability and Performance Analysis of HIPERLAN*," Dept. of Information Engineering, University of Piza, Italy.

Chevrel, S.; A.H. Aghvami; H.-Y. Lach; and L. Taylor. 1996. "Analysis and Optimization of the HIPERLAN Channel Access Contention Scheme," *Wireless Personal Communications*, Vol. 4, pp. 27-39.

Coutras, C.; Peng-Jun Wan; O. Frieder. 2000. "Analytical modeling and performance evaluation of the HIPERLAN CAC layer protocol for real-time traffic", *25th Annual IEEE Conference on Local Computer Networks (LCN'00)* November 08 - 10, Tampa, Florida.

ETSI. 1998. EN 300 652 V1.2.1 (1998-07), ETSI, Broadband Radio Access Network (BRAN); High Performance Radio Local Area Network (HIPERLAN) Type 1; Functional Specification.

FU, K.; Y.J. GUO; and S.K. Barton. 1996. "Performance of the EY-NPMA Protocol, "*Wireless Personal Communications* Vol. 4, pp. 41-50.

Halls G.A. 1994. "HIPERLAN: the high performance radio local area network standard", *Electronics and Communication Engineering Journal* 6(6) (December 1994) 289-296.

Jacquet, P.; P. Minet; P. Muhlethaler; and N. Rivierre. 1996. "Priority and Collision Detection with Active Signaling – The Channel Access Mechanism of HIPERLAN," *Wireless Personal Communications*, Vol. 4, pp. 11-26.

Jacquet, P.; P. Minet; P. Muhlethaler; and N. Rivierre. 1996. "Data Transfer for HIPERLAN," *Wireless Personal Communications* Vol. 4, pp. 65-80.

LaMaire, R. O.; A. Krishna; P. Bhagwat; and J. Panian. 1996. "Wireless LANs and Mobile Networking: Standards and Future Directions", *IEEE Communication*, 34(8):86-94.

Moh, W. M.; D. Yao; and K. Makki. 1998. "Wireless LAN: Study of hidden terminal effect and multimedia support," *Proc. Computer Communications and Networks*, pp. 422-431, October 12-15.

Nicopolitidis, P.; M.S. Obaidat; G.I. Papadimitriou; and A.S. Pomportsis. 2003. " Wireless Networks," Wiley.

Obaidat, M. S.; and D. Green. 2003. "Simulation of Wireless Networks," in Applied Systems Simulation: Methodologies and Applications (M. S. Obaidat and G. I. Papadimitriou, Eds.), Kluwer (in press).

Papadimitriou, G. I.; and A. S. Pomportsis. 2000. "Learning-Automata-Based TDMA Protocols for Broadcast Communication Systems with Bursty Traffic," *IEEE Communication Letters*, Vol. 4, No.3.

Sadiku, M.; and M. Ilyas. 1995. "Simulation of Local Area Networks," *CRC Press*.

Tanenbaum Andrew S. 1996. Computer Networks, 3d Edition, Prentice-Hall, Inc.

Weinmiller, J.; M. Schlager; A. Festag; and A.Wolisz, "Performance Study of Access Control in Wireless LANs – IEEE 802.11 DFWMAC and ETSI RES 10 HIPERLAN," Technical University Berlin, Germany.

Wilkinson, T.; T. G. C. Phipps; and S. K. Barton. 1995. "A report on HIPERLAN standardization", International Journal of Wireless Information Networks, Vol. 2, No. 2, pp.99-120.

Wilkinson Tim. "HIPERLAN", HP Labs Europe.

Zeng J. 2000. "Wireless LAN Standards: HIPERLAN and IEEE802.11, *ECPE 6504 Wireless Networks and Mobile Computing*, Individual Project Report, 04-24-2000.