IEEE 802.11 Wireless Network Simulator Using Verilog

BABER ASLAM[1], MONIS AKHLAQ[1], SHOAB A. KHAN[2]
Information Security Department, College of Signals[1]
Computer Engineering Department, College of E&M Engineering[2]
National University of Science & Technology
Tamizuddin Road, Rawalpindi
PAKISTAN

Abstract: - This paper presents a basic IEEE 802.11 network simulator using Verilog. Basic aim of this paper was to design a simulator using a hardware description language (HDL) such as Verilig, since the functions and protocols described in a state machines are best simulated using a HDL. Many of the security vulnerabilities in IEEE 802.11 are due to unauthenticated / unencrypted management and control frames which result in a number of Denial of Service (DoS) attacks. Current IEEE 802.11 simulators deal with Physical and MAC layers and do not include the exchange of management and control frames, thus making it difficult to simulate an attack (DoS) and its possible solution. The proposed simulator includes the communication setup process and can be used for simulating DoS attacks and its possible solutions.

Key-Words: - Network Simulator, IEEE 802.11, Verilog, Wireless LAN, Link layer simulation.

1 Introduction

Due to ease of installation, expansion, modification and maintenance, wireless technology is the first choice of today's network administrators and home users. IEEE 802.11, which was ratified in 1999, is the basic and widely adopted standard for WLANs [1]. The standard has been evaluated and updated in different periods but still provides space for improvements. One example is IEEE 802.11i, which was defined to address the security vulnerabilities of WLANs [2]. However, IEEE 802.11i is still vulnerable to many Denial of Service (DoS) attacks because of its unauthenticated management and control frames [3, 4]. This makes WLANs an open area for research by communication experts. The space also exists for simulation / verification being an important step in any research work.

In IEEE 802.11 WLANs, the actions of an access point (AP) or a wireless client depends on their respective current states. These states are maintained at each end by a state machine and the transitions between these states are controlled by management / control frames. The Robust Security Network Association (RSNA) Establishment [2] is the communication setup procedure that involves transition between all states and establishment of keying material at both ends. Many DoS attacks involve spoofing of management frames used in RSNA establishment. In order to study the attacks and test the proposed solutions, simulation of RSNA establishment phase and exchange of relevant frames is required.

This paper presents a basic simulator using Verilog. The simulator includes the AP and wireless clients. RSNA establishment phase has been specially included to help the researchers to study different DoS attacks and test the proposed solutions. The simulator is the first prototype and may require improvements / enhancements of functionality.

The paper is organized into sections. Section 2 introduces some of the existing simulators and section 3 gives brief introduction to Verilog. Section 4 describes the simulator design and finally section 5 gives a sample simulation scenario.

2 Network Simulators

Some of the network simulators that can be used for simulation of wireless networks are described here. These can be broadly divided into two categories i.e., Open Source Simulators and Commercial Simulators.

2.1 Open Source Simulators

2.1.1 NS 2

Network Simulator2 (NS2) is a discrete event simulator that runs more efficiently on Linux platforms [5]. It is freely available and has an active user list to discuss the queries. Its complete source code is available which also includes decent documentations. Some tutorials on NS2 are also available on the web that can serve as a good starter. NS2 uses both C++ and OTCL. C++ is used for source coding the modules and OTCL for simulation control. These two are tightly bound with each other. NS2 has good support for wireless networks and 802.11 protocol is well supported however, NS2 provides no support for the infrastructure mode. Detailed functioning of MAC is not included and Management frames are also simply discarded. Due to its object oriented programming any modification requires understanding of complete code of the simulator and specially the C++ and OTCL binding.

2.1.2 OMNET++

OMNET++ is a component-based, modular and open-architecture simulation environment having strong Graphic User Interface (GUI) [6]. Its use is free for academic and non-profit purposes. It also has a user list which is a support forum. Its INET framework deals with simulations of wired, wireless and ad-hoc networks. However, its wireless modules are not yet fully developed and only support ad hoc mode which has no functionality regarding authentication / association etc. New modules can be added but sketchy documentation makes it very difficult.

2.1.3 GloMoSim

GloMoSim (Global Mobile Information Systems Simulation Library) is mainly a simulator for wireless networks [7]. It's Academic and research version is free to download (from a .edu domain). Its design is based on a layer approach similar to seven layers of OSI model. It makes use of Parsec (Parallel Simulation Environment for Complex Systems) [8] for parallel discrete event simulation. In order to understand and modify the source, the user need extensive knowledge of Parsec. QualNet is commercial product that is based on GloMoSim [9].

2.2 Commercial Simulators

2.2.1 OPNET

The network simulator OPNET (Optimized Network Engineering Tool) is easy to use and is having excellent graphical user interface [10]. It has different versions such as academic, research, complete etc. Only academic version is freely available with limited features, less documentation and no support. The current academic version (IT Guru Academic Edition Build 1996) does not have detailed functions about wireless networks especially infrastructure mode. It is impossible to make modification / additions in academic version.

2.2.2 NetSim

NetSim is a discrete event simulator with an object based modeling approach [11]. It includes models for various LAN and WLAN protocols. It has two versions, a standard version and an academic version. It is a good teaching and learning tool. Modifications in its source code are however difficult being a commercial product.

3 Verilog

Verilog is a Hardware Description Language (HDL). Hardware Description Language is used to describe a digital system, it can be as simple as an adder or as complex as a microprocessor. Further, HDL can define the system at different levels of abstraction. The basic difference between a programming language and a HDL is that HDL has modules that run concurrently like a hardware, further, timings can also be coded in the description.

To standardize Verilog a IEEE Standards Group for Verilog (VSG) was established in 1993. The first standard, IEEE 1364-1995, was released in 1995 [12]. The latest standard is Version C of IEEE 1364-2001 [13]. The standardization has further increased the use of Verilog.

The basic purpose of Verilog or any HDL is to simulate a digital system. This simulation is carried out to test the design before going for final hardware prototyping. AP and wireless network cards are essentially a digital system and can be best simulated in HDL specially those with hardware acceleration functions. Further in a WLAN, each AP or wireless client has to operate independently and concurrently as a separate hardware device. This can be best simulated in Verilog.

Designing in Verilog involves four steps (Fig 1). Step 1 define the system using Verilog. In this step already defined modules can also be used. These can be vendor defined modules, for example Microprocessor 8051. Step 2 is compiling the design file. This is followed by step 3 which is simulation. Simulation results may be available in textual or graphical form. Debugging step involves checking/ analyzing the simulation results and modifying the design files.

Step 1 can be performed using any text editor, for step 2,3 and 4 many tools are available. A list of free tools for Verilog can be found at Verilog. Net [14].

For our simulation we used ModelSim Xilinx Edition III v6.1e which is a free version with decent functionality. ModelSim is a verification and simulation tool for hardware. The hardware to be verified or simulated is described using hardware description languages such as VHDL, Verilog etc. ModelSim has many supporting functions such as waveforms that can be observed to confirm the functioning of a device.

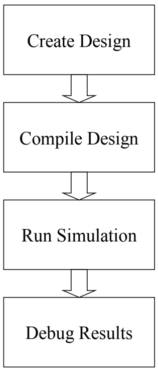

Fig 1: Steps in System Design with Verilog

4 Simulator Design

The simulator consists of a number of modules that have been integrated through a project in ModelSim called WLAN. The main modules are media, carrier sense, receive, transmit, test bench and log modules. The relationship between different modules is shown in Fig 2. Test bench controls the simulation and log gives the results of simulation. Media module makes communication possible between AP and wireless nodes. The details of the simulator are explained in next sub sections.

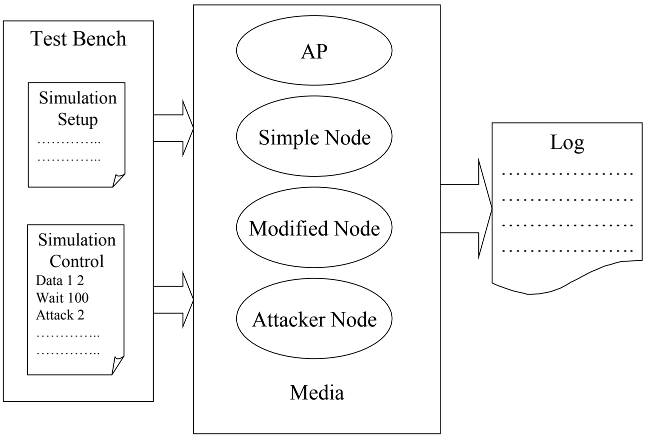

Fig 2: Simulator Design

4.1 Frame Format

The frame format used is given in Fig 3. It is essentially the IEEE 802.11 standard frame with some simplification.

2 Bits	4 Bits	1 Bit	1 Bit	48 Bits	48 Bits	48 Bits	4 Bits	12 Bits	48 Bits	0 – 2312 Bytes
Type	Sub Type	To DS	From DS	Addr 1	Addr 2	Addr 3	Frag no	Seq No	Addr 4	Frame Body

Fig 3: Frame Format

4.2 Media Module

The media module simulates wireless media. It's a shared space that is used for writing the sent data. When a node needs to send a frame, it writes it to the media space. All nodes then check the media for their addresses and read the frame accept if it's for them else discard it. Two status flags are used, first to indicate whether media is busy or idle and second to indicate whether node is sending or receiving a frame.

4.3 Carrier Sense Module

This module is implemented in all nodes and AP. It is basically a simplified version of carrier sense multiple access / collision avoidance. Any node that has a frame to send monitors the media, when media is free it generates a random backoff and waits for that backoff before sending the frame. If during the wait state the media gets busy, the node stops the backoff down counter and waits for the media to be free again. When media is free it again starts its backoff down counter from the stopped state and waits till counter is zero after that it sends the frame. The flow is shown if Fig 4.

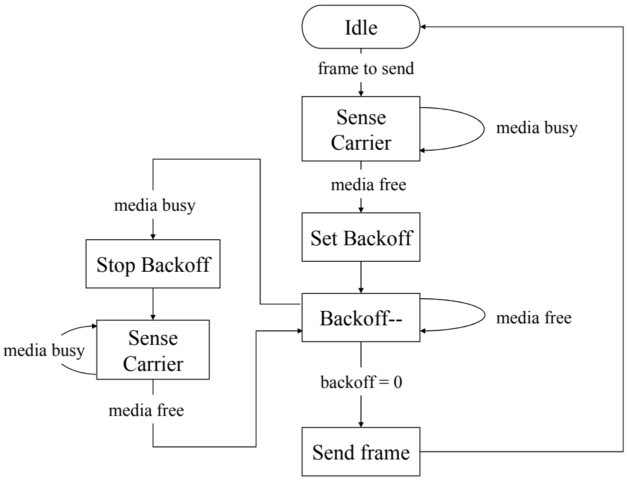

Fig 4: Flow Diagram – Carrier Sense Module

4.4 802.11 /802.1x State Machine

The state machine maintains authentication /association state of wireless node and AP. AP maintains a separate state machine for each of its authenticated / associated wireless nodes. Both the send and receive modules make use of state machine while processing the frames. AP and wireless client both start with state 1. Client requests authentication using Open System authentication to AP. On successful completion of authentication, both AP and wireless client transit to state 2. In state 2, wireless client initiates association request and on successful association both transit to state 3. In state 3, depending on authentication mechanism, 802.1x authentication messages will be exchanged. Pair wise Master Key (PMK) is established both at wireless node and AP after successful authentication. The 4-way handshake follows to generate keys for data encryption and data integrity. Both the AP and client are in state 4 and can use data encryption using generated keys. As authentication server has not been implemented in simulator therefore the transition to state 4 and establishment of keys is implicit to transition to state 3. State transitions are shown in Table 1

Table 1: 802.11 / 802.1x State Transition Table

		Next State		
		State 1 Unauthenticated-Unassociated	State 2 Authenticated-Unassociated	State 3 Authenticated -Associated
Current State	State 1 Unauthenticated-Unassociated		Successful Authentication	
	State 2 Authenticated-Unassociated	Deauthentication Notification		Successful Association
	State 3 Authenticated-Associated	Deauthentication Notification	Disassociation Notification	

4.5 Send Module

Send module is implemented in AP and wireless nodes. When there is packet to send and carrier sense module has set send ready flag then the node is ready to send frame. Before sending the frame the current state of receiving node is checked and if current frame can be sent according to MAC protocol then send action is performed else the frame is discarded and associated actions are performed.

4.6 Receive Module

Receive module is implemented in AP and wireless nodes. After transmitting the frame the transmitting node changes the status of media to receiving. On this all nodes check the frame and if receiving address matches to that of the node then frame is received. After receiving the frame it is processed or dropped according to the current state of 802.11 state machines. Table 2 and 3 show the actions performed on receiving a frame by wireless node and AP respectively.

Table 2: Actions by Wireless Node on receiving a frame

		Current State		
		Unauthenticated Unassociated - UU	Authenticated Unassociated - AU	Authenticated Associated - AA
Received Frame	Association Response	1. Drop frame 2. Send Deauthentication	1. Change state to AA	1. Drop frame
	Disassociation	1. Drop frame 2. Send Deauthentication	1. Drop frame	1. Change state to AU
	Authentication Response	1. Change state to AU	1. Drop frame	1. Drop frame
	Deauthentication	1. Drop frame	1. Change state to UU	1. Change state to UU
	Data	1. Drop frame 2. Send Deauthentication	1. Drop frame 2. Send Disassociation	1. Process frame

Table 3: Actions by AP on receiving a frame

		Current State		
		Unauthenticated Unassociated - UU	Authenticated Unassociated - AU	Authenticated Associated - AA
Received Frame	Association Request	1. Drop frame 2. Send Deauthentication	1. Send Association Response	1. Send Association Response 2. Change state to AU
	Disassociation	1. Drop frame 2. Send Deauthentication	1. Drop frame	1. Change state to AU
	Authentication Request	1. Send Authentication Response	1. Send Authentication Response 2. Change state to UU	1. Send Authentication Response 2. Change state to UU
	Deauthentication	1. Drop frame	1. Change state to UU	1. Change state to UU
	Data	1. Drop frame 2. Send Deauthentication	1. Drop frame 2. Send Disassociation	1. Process frame according to Destination's state

4.7 Test Bench Module

Test bench module is the main / top-level module of simulator. Wireless network to be simulated is defined in this module. It instantiates and interconnects different devices in simulation such as AP, media, wireless nodes etc. The simulation to be performed is also defined in the test bench. It also serves as application layer providing data to nodes for transmission. Test bench can access and modify internal states of different module thus making comprehensive testing / simulation possible.

The sequences of events that take place are defined in test bench, such as, start of communication setup process, Data to be sent by a wireless client, timings of different events etc.

4.8 Log Module

Log module defines the logs that are generated by each wireless node and AP. Every device makes an entry to log file whenever it performs some action. On termination of simulation the log gives detailed account of the sequence of events performed by various devices. The log can be observed to check the behavior of any device. The layout and legend of log are given in Fig 5 and Table 4 respectively.

Time	Action	At node	Type	Sub type	To DS	From DS	RA	TA	DA	SA	Seq No	Remarks

Fig 5: Log - Layout

Table 4: Log - Legend

S/ No	Action Symbol	Meaning
1	r	Frame received
2	s	Frame sent
3	d	Frame dropped
4	p	Frame processed
5	Q	Frame enqueued
6	D	Frame dequeued
7	c	Collision detected
8	x	No frame to send

5 Sample Simulation

5.1 Simulation Scenario

The simulation scenario comprises of wireless network in infrastructure mode having an access point (AP) and three wireless nodes. The network is shown in Fig 6.

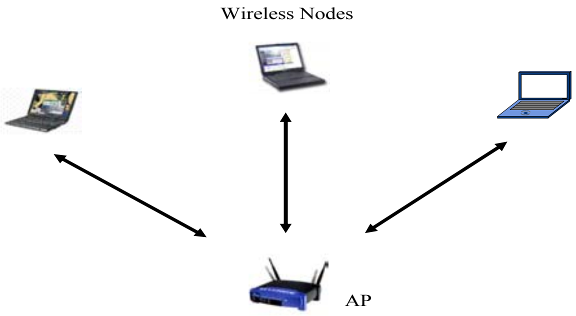

Fig 6: Simulation Scenario

5.2 Simulation Conducted

First of all the normal functioning of AP and wireless nodes was checked. For this a wireless network consisting of two wireless nodes and AP was simulated. Data was transferred between two wireless nodes via AP. This involved completion of communication setup process. The recovery from different intermediate states was also checked by controlling states via test bench. The results were checked by both the waveform and log. Waveform graphically shows the transition to states, sending / receiving of frames, updating of different variables/ states. The log is more concise and only provides the actions performed at various devices along with time and other important data. A partial log output is shown in Fig 7.

```
  49,  Q,     1,  2, 0,                         2,     1,          Data to send
 180,  D,     1,  2, 0,                         2,     1,          Frame to send
 180,  Q,     1,  2, 0,                         2,     1,          Cannot send: UU
 180,  Q,     1,  0,11,                       255,     1,          AuthReq
 181,  D,     1,  0,11,                       255,     1,          Frame to send
 183,  s,     1,  0,11,  0,0,   255,     1,   255,     1,  3714,   Data sent
 185,  r,   255,  0,11,  0,0,   255,     1,   255,     1,  3714,
 185,  p,   255,  0,11,  0,0,   255,     1,   255,     1,  3714,   AuthReq
 185,  Q,   255,  0, 8,                         1,   255,          AuthResp
 240,  D,   255,  0, 8,                         1,   255,          Frame to send
 242,  s,   255,  0, 8,  0,0,     1,   255,     1,   255,  1636,   Data sent
 245,  p,   255,  0, 8,                         1,   255,          New state AU
 245,  r,     1,  0, 8,  0,0,     1,   255,     1,   255,  1636,
 245,  p,     1,  0, 8,  0,0,     1,   255,     1,   255,  1636,   New state AU
 330,  D,     1,  2, 0,                         2,     1,          Frame to send
 330,  Q,     1,  2, 0,                         2,     1,          Cannot send: AU
 330,  Q,     1,  0, 0,                       255,     1,          AssocReq
 331,  D,     1,  0, 0,                       255,     1,          Frame to send
 333,  s,     1,  0, 0,  0,0,   255,     1,   255,     1,  3715,   Data sent
 335,  r,   255,  0, 0,  0,0,   255,     1,   255,     1,  3715,
 335,  p,   255,  0, 0,  0,0,   255,     1,   255,     1,  3715,   AssocReq
 335,  Q,   255,  0, 1,                         1,   255,          AssocResp
 350,  D,   255,  0, 1,                         1,   255,          Frame to send
 352,  s,   255,  0, 1,  0,0,     1,   255,     1,   255,  1637,   Data sent
```

Fig 6: Partial Log

Second set of simulations were conducted with AP and three wireless nodes. Data transfers between three wireless nodes were simulated. The internal states of wireless clients were then altered to check their response and recovery. Internal states of AP were also modified to confirm its response. The log was observed which confirmed that the network was performing as per design parameters.

Further simulations were also conducted by changing the sequence of events in Test bench module. The network was then expanded to five wireless clients. The functionality of Carrier sense module was specially checked to confirm that each wireless client has equal probability of getting the access.

6 Conclusion

The WLAN standards are still in development phase, there are still many open and interesting research areas in this field. One of the important step in research is to check the results through simulation. In this paper we have designed a basic WLAN simulator using a HDL. WLAN standard is based on state machines and state machines can be best simulated using a HDL. Further if a modification / solution is tested through HDL then there are all the more chances that it will be implement able in hardware. The simulator can be expanded by adding reusable libraries such as vendor specific library/ module for an AP or a wireless client. This makes simulation more realistic.

References:

[1] IEEE Standard 802.11-1999. Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control and Physical Layer Specifications. 1999.

[2] IEEE P802.11i – 2004. IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks— Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 6: Medium Access Control (MAC) Security Enhancements. July 2004.

[3] J. Bellardo, and S. Savage. 802.11 Denial-of-Service attacks: real vulnerabilities and practical solutions. In *Proceedings of the USENIX Security Symposium*, pages 15-28, August, 2003.

[4] AusCERT. Denial of Service Vulnerability in IEEE 802.11 Wireless Devices . *AusCERT AA-2004.02*, Australian Computer Emergency Response Team, [online] www.auscert.org.au/ render.html?it=4091, 13 May 2004.

[5] The Network Simulator – ns-2 [online] www.isi.edu/nsnam/ns/

[6] OMNET++ Discrete Event Simulation System [online] www.omnetpp.org/

[7] GloMoSim, [online] pcl.cs.ucla.edu/ projects/ glomosim

[8] Parsec, Parallel Simulation Environment for Complex Systems, [online] pcl.cs.ucla.edu/ projects/ parsec/

[9] QualNet, [online] www.qualnet.com/products/ QualNet/

[10]OPNET Technologies, Inc [online] www.opnet.com

[11] NetSim, tetcos, [online] www.tetcos.com

[12] IEEE Std 1364-1995, IEEE Standard Hardware Description Language Based on the Verilog® Hardware Description Language,.The Institute of Electrical and Electronics Engineers, Inc. 345 East 47th Street, New York, NY

[13] IEEE Std p1364-2001, IEEE Standard Hardware Description Language Based on the Verilog® Hardware Description Language,.The Institute of Electrical and Electronics Engineers, Inc. 345 East 47th Street, New York, NY

[14] Verilog.Net Free Tools, [online] www.verilog.net/ free.html