International Journal of Wireless Communications and Mobile Computing
2014; 2(4): 42-51
Published online December 02, 2014 (http://www.sciencepublishinggroup.com/j/wcmc)
doi: 10.11648/j.wcmc.20140204.11
ISSN: 2330-1007 (Print); ISSN: 2330-1015 (Online)

Design and Implementation of a Testbed for Mobile Adhoc Network Protocols

Akhtar Hussain, Aimel Khan, Abdul Rehman Qaiser, Muhammad Mohsin Akhtar, Obaidullah Khalid, Muhammad Faisal Khan*

Dept. of Electrical Engineering, National University of Sciences and Technology, Islamabad, Pakistan

Email address:
mfaisalkhan@mcs.edu.pk (M. F. Khan)

To cite this article:
Akhtar Hussain, Aimel Khan, Abdul Rehman Qaiser, Muhammad Mohsin Akhtar, Obaidullah Khalid, Muhammad Faisal Khan. Design and Implementation of a Testbed for Mobile Adhoc Network Protocols. *International Journal of Wireless Communications and Mobile Computing*. Vol. 2, No. 4, 2014, pp. 42-51. doi: 10.11648/j.wcmc.20140204.11

Abstract: Due to the significant growth in the area of wireless communication in the last few years, Quality of Service (QoS) has become an important consideration for supporting variety of applications. Elaborate testing of newly developed Mobile Adhoc Network's routing protocols in real world scenarios is a key step for providing QoS to the users. In order to test the developed Adhoc Routing Protocols, an IEEE 802.11 based testbed has been developed in the paper. Some well-known Adhoc network protocols were implemented and tested in user space daemon in Linux. All the key parameters necessary for assessing an Adhoc Network Protocol have been analyzed. A user application was developed in Java NetBeans Integrated Development Environment (IDE) to ensure repeatability and to provide a mobility model to the users in an efficient way. Customized and special purpose nodes have been developed by integrating various hardware components, in order to improve the efficiency and robustness of the testbed. Some of the well-known protocols have been tested by exposing the configured nodes to an outdoor atmosphere to cater all the unforeseen environmental factors, which affect the performance of Adhoc Protocols. Different performance metrics like overhead, throughput, end to end delay, average jitter and packet loss were evaluated by varying mobility, number of hops, packet size and pause time.

Keywords: Ad-hoc Routing Protocols, AODV, B.A.T.M.A.N, File Transfer Protocol, Mobile Ad-hoc Networks, Test-Bed, Network Time Protocol, OLSR

1. Introduction

Mobile Ad hoc Network (MANET) is an infrastructure-less network and can be deployed instantly to serve temporary or urgent purposes. MANETs promises to be useful in disaster management, for military in combat zones, for spontaneous meetings and for all those scenarios where the existing infrastructure is damaged or difficult to deploy a trivial communication infrastructure. Several protocols have been proposed and many of them are being practiced in real world scenarios. Bluetooth and Wi-Fi are among the common communication links used for mobile adhoc networks. Whenever new protocols are developed, their viability and efficiency is being evaluated, to assess their key features, conquer their shortcomings and add more appealing features.

There are several methods for testing the developed protocols. Theoretical evaluation provides elementary insight into the features of the investigated approach. Software simulations can be used for initial design and for an estimation of the results, but they do not realistically duplicate the physical layer. What emulator testbeds have in common is that they try to address the problems of scaling, management and test repeatability; but emulator testbeds cannot always be a substitute for real world scenarios. Testbeds with indoor experiments by using MAC filtering are also common. Indoor testing of protocols doesn't cater to the environmental conditions and open-air interruptions. MANETs are usually used for emergency purposes, and they find their major usages in open-air scenarios.

An emulated testbed was proposed in [1] which compresses the network and emulates the mobility without actually moving the nodes. An End to end delay analysis model was proposed in [2] with special consideration to MAC delay contention time while [3] compares the end to end delay in pervasive multimedia networks. A testbed for evaluating various performance parameters of Adhoc protocols was suggested by [4] with varying mobility and node congestion. Jitter of different Adhoc protocols was compared by [5]. Some security issues related to Adhoc networks were discussed in [6] and a quantitative analysis of a full-scale multi-hop Adhoc network testbed was carried out by [7].

The requirements for a testbed to be reliable and successful for testing the newly developed protocols are numerous. Some of them are scalability, cost-effectiveness, reproducibility, reliability and management [2]. A testbed with a user application and dedicated hardware will serve this purpose in an efficient way.

This paper proposes an IEEE 802.11 based testbed for adhoc network protocols. Some well-known Adhoc network Protocols i.e AODV, OLSR and B.A.T.M.A.N were implemented in user space daemon in Ubuntu. All the tests have been carried out in open air to consider all the possible interferences and unseen environmental factors, which influence the performance of the routing protocols. The developed testbed has been equipped with an application which was developed in Java NetBeans IDE. Testing nodes have been developed by integrating various hardware components to make a customized hardware to facilitate the users in testing and to ensure reproducibility in an efficient and robust way.

The rest of the paper is organized as follows. In section 2, a brief introduction of the tested protocols and performance metrics for assessing Adhoc Protocols has been presented. In section 3, the developed testbed has been discussed in detail; while in section 4, the results of the tested protocols and discussions regarding each result have been presented and are followed by conclusions.

2. Adhoc Network Protocols and Performance Metrics

IEEE has standardized the 802.11 protocol for Wireless Local Area Networks. It works in two modes: DCF (Distributed Coordination Function) and PCF (Point Coordination Function). 802.11 DCF can be used as a MAC scheme for multi-hop Wireless Adhoc Networks. It is CSMA/CA with binary slotted exponential bakeoffs. Data transmission by a mobile node in DCF mode using RTS and CTS is shown in figure 1.

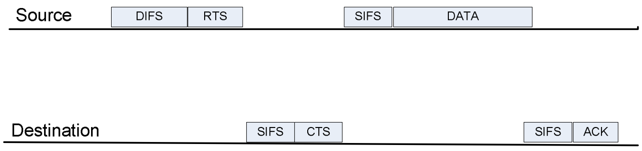


Fig. 1. *Data Transmission in IEEE 802.11in DCF mode*

Protocols governing the routing of packets within a MANET's scenario can be categorized as Reactive, Proactive and Hybrid Protocols. In proactive protocols, the route information is kept up to date by exchanging the control packets between the nodes periodically. While proactive protocols initiate a route discovery only when a node requires a route to a certain node. Hybrid protocols are the result of an intelligent combination of reactive and proactive protocols, in order to increase scalability and reduce the overhead.

2.1. Tested Adhoc Network Protocols

The protocols which have been used for testing the viability of the developed testbed in this paper are Adhoc On demand Distance routing Vector (AODV), Optimized Link State Routing (OLSR) and Better Approach To Mobile Adhoc Networking (B.A.T.M.A.N). The basic working principle of each protocol is discussed in the following section.

2.1.1. Adhoc On Demand Distance routing Vector

AODV is a reactive protocol and its working is based on the Distance Vector algorithm, which tells the information about its neighbors and number of hops required to reach them. Figure 2 shows the different types of messages used by the AODV for finding a route to a destination [7]. Where Hello messages are used to detect and monitor links to the neighbors.

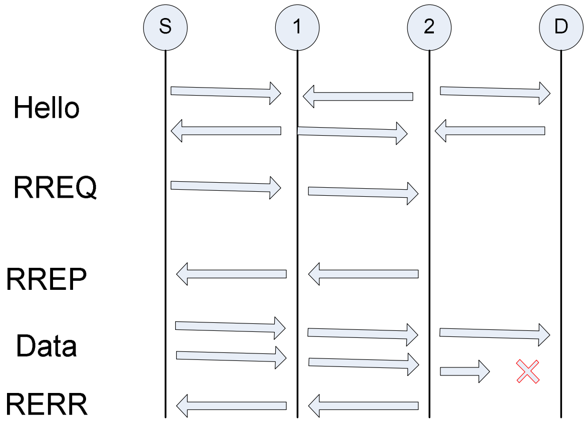


Fig. 2. *AODV Protocol Messaging*

When the source has to send data to an unknown destination, it broadcasts a Route Request (RREQ) message to that destination. The destination generates a Route Reply (RREP) and unicasts to the source. If the data is flowing and a link failure is detected, Route Error (RERR) is sent to the source.

2.1.2. Optimized Link State Routing

OLSR is an optimized version of Link State Routing Protocol. The protocol inherits the stability of Link State Algorithm and has the routes readily available pertaining to its proactive nature. In order to reduce the overhead, only the nodes designated as MPRs (Multipoint Relays) are responsible for forwarding the control information. Figure 3 shows the selection of the MPRs [8]. The rule for selecting an MPR is "For all 2 hops neighbors' n there must exist an MPR m so that n can be connected via m".

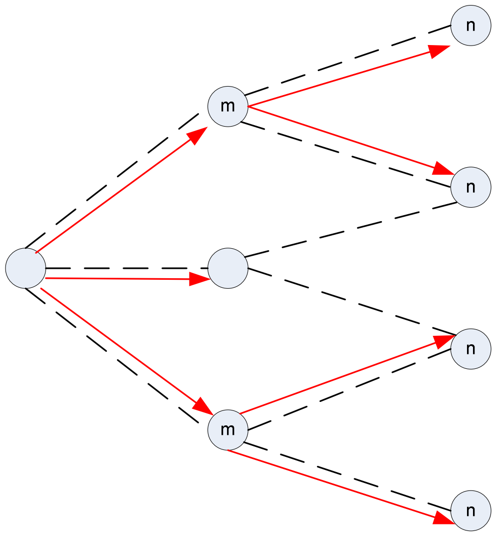


Fig. 3. *MPR Selection in OLSR.*

2.1.3. Better Approach to Mobile Adhoc Networking

B.A.T.M.A.N is also a proactive protocol and is especially designed for networks where the view of the topology is ambiguous and altering constantly. It finds the optimal gateway node to the destination instead of a complete route. It uses OMGs (Originator Messages) to calculate link quality and to identify the path. Figure 4 shows the OMGs and its contents.

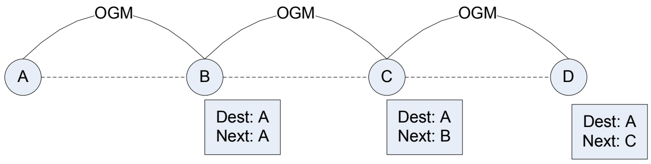


Fig. 4. *B.A.T.M.A.N Originator Messages.*

The approach used by BATMAN is to generate an Originator Message periodically. The source name and sequence number is used to uniquely identify a packet and is used to detect duplicates. On receiving an OMG, the routing tables are updated [9].

2.2. Performance Metrics

Performance Metrics encompasses QoS (Quality of service) to the end users in terms of several general parameters [10]. The perceived Quality of Service can be measured in terms of several parameters. The key parameters for evaluating an Adhoc Network Protocol are Overhead, Throughput, Jitter, End to End Delay and Packet Loss. Several papers have been written on these performance metrics [10], [11], [12], [13]. All of these five parameters have been analyzed by the developed testbed in this paper. Details of each parameter are in the following sections.

2.2.1. Overhead

Control traffic overhead mainly depends upon the topology and the data traffic in addition to the protocol being used. AODV generate overhead only when a new route is needed while OLSR and BATMAN continuously generate control traffic. The formula for calculating control traffic in both cases is shown in table 1[14].

Table 1. *Equations for Overhead Computation*

	Reactive Protocols	
	Packets	**Bandwidth**
Fixed	ßôN^2+ÞN	ÞPN+ßôRQrN^2
Mobile	ôµaLN^2	ôpµaRQrN^2
	Proactive Protocols	
	Packets	**Bandwidth**
Fixed	ßN+ôptpN^2	ÞPN+ôptpTpN^2
Mobile	ôµANpN^2	ôpµANpTpN^2

Given the parameters shown in table 2, overhead of both types of protocols can be calculated. Table 1 contains different equation for computing overhead for fixed and mobile nodes. Overhead is a key parameter in analyzing adhoc routing protocols and has been discussed in detail in different papers [14], [16].

Table 2. *Parameters for equation in table 1.*

Reactive Protocol Parameters	
ß	Route creation rate per node
ô	Route request optimization factor
N	Number of Nodes
Þ	Hello rate
RQr	Average size of route request
µ	Link breakage rate
L	Number of route reply messages
µa	Active node link breakage rate
ôp	Broadcast optimization factor
Proactive Protocol Parameters	
ß	Route creation rate per node
Ôp	Broadcast optimization factor
N	Number of Nodes
Þ	Hello rate
ANp	Active next hop ratio
Tp	Average size of topology broadcast packet
P	Number of route reply messages
P	Active node link breakage rate
tp	Topology broadcast rate

2.2.2. Throughput

A networks end to end throughput is a measure of the network's successful transmission rate and is defined as bytes per second [20]. End to end throughput can be calculated by using equation (1) [15].

$$TP = \sum_{n=1}^{N} \left(\frac{PS(n)}{PAT(n) - PST(n)} \right) \tag{1}$$

Where TP: Throughput, PS: Packet Size, PAT: Packet Arrival Time, PST: Packet Sending Time and N is the number of packets transmitted. In order to calculate the throughput, the size of each data packet was added. This gives the total data transferred [15]. Throughput is an important metric for evaluation of any Adhoc network protocols. Depending on the type of data used for transmission, the acceptable thresholds of throughput will be different. Throughput is an index for Quality of Service in all types of data.

2.2.3. End to End Delay

End to end delay is the average time between generation and successful delivery of the packets for all nodes in the network [17], [18]. The major sources of delay could be processing delay, network delay, propagation delay and destination processing delay. Equation for calculating end to end delay is given as (2) [15].

$$E2ED = \sum_{n=1}^{N} (\frac{PAT(n) - PST(n)}{N}) \qquad (2)$$

Where E2ED is end to end delay and remaining parameters are the same as in the equation (1).

Packet arrival is the time when a packet reaches the destination. End to end delay may be larger in open air due to unforeseen environmental effects.

2.2.4. Average Jitter

Jitter is defined as the variation of data communication packets in the network. It is the variation in time between each of the packets arriving [19]. Jitter is an index for consistency and stability of the network. Jitter can be calculated by equation (3) [15].

$$PPD(n) = \sum_{n=1}^{N} (PAT(n+1) - PST(n+1)) \qquad (3)$$

$$CPD(n) = \sum_{n=1}^{N} (PAT(n) - PST(n)) \qquad (4)$$

Where PPD is Previous Packet Difference, CPD is Current Packet Difference and AJ is Average Jitter. Average jitter can be calculated by using equation (3) and equation (4), as shown in equation (5).

$$AJ = \sum_{n=1}^{N} (\frac{PPD(n) - CPD(n)}{N-1}) \qquad (5)$$

Average jitter is a critical element in determining the performance of the network and QoS offered by the network.

2.2.5. Packet Loss

Packet loss or corruption of packets indicates the packets which have been sent by the sender but not received by the destination node. It affects the perceived quality of the application. Packet loss could be due to unstable wireless connection, overflowing of the queuing buffer or congestion in the network. The formula formulated for computing lost packets is equation (6) [15].

$$PL = \frac{\sum_{l=1}^{L} (LP(l) * Size(l))}{\sum_{n=1}^{N} (SP(n) * Size(n))} * 100 \qquad (6)$$

Where PL is Packet Loss, LP is Lost Packet, SP is Sent Packet, L is number of lost packets. All of the above mentioned performance metrics are evaluated by the developed testbed. The development process of the complete testbed is explained in the following section.

3. MANETs Testbed

The developed Mobile Adhoc Networks testbed is capable of providing a generic platform for the testing of Adhoc protocols in real world scenarios, equipped with a suitable software environment for the facilitation of user's utility. Real world testing is being carried out to cater for deficiencies of simulations, which are based on a significant level of abstractions.

All the protocols mentioned in section 2 were implemented in user space daemon. A daemon is a program that runs in the background, rather than under the direct control of a user. It communicates with the kernel module to discover and maintain the routes. User space daemon implementation is advantageous to kernel modification in many aspects. The user space communicates with the kernel space through sockets as shown if figure.5.

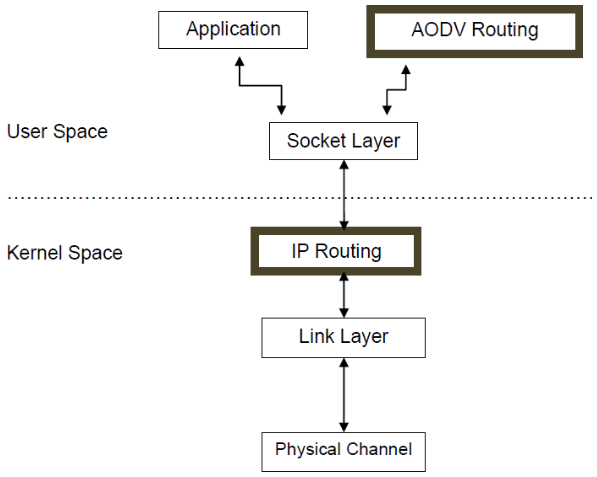


Fig. 5. *User Space Daemon*

The developed testbed is composed of a hardware implementation and a software application. The software application was developed in Java NetBeans IDE and testing nodes were developed by integrating several hardware components. Details about tested data, software application and hardware are in the following sections.

3.1. Tested Data

Different types of data have been used in the proposed testbed to analyze the robustness of the testing protocol. Test data includes file transfer through FTP and video streaming through VLC. All the nodes have been time synchronized with an NTP server. Precise time synchronization is required to ensure the viability of the test results resulting in a reliable analysis. NTP is used for synchronizing the clocks of the computer systems over packet-switched, variable-latency data networks. It utilizes UDP on port 123.

FTP is based on a client-server architecture which makes use of separate control and data connections between the client and server. The client is allowed to access to the server's database after entering the correct username and password.

VLC has been utilized to stream videos from one to multiple nodes over the adhoc network. Before streaming is initialized, various parameters such as transcoding rate and the video format together with the IP address of the destination are required to be specified.

3.2. Software Application Development

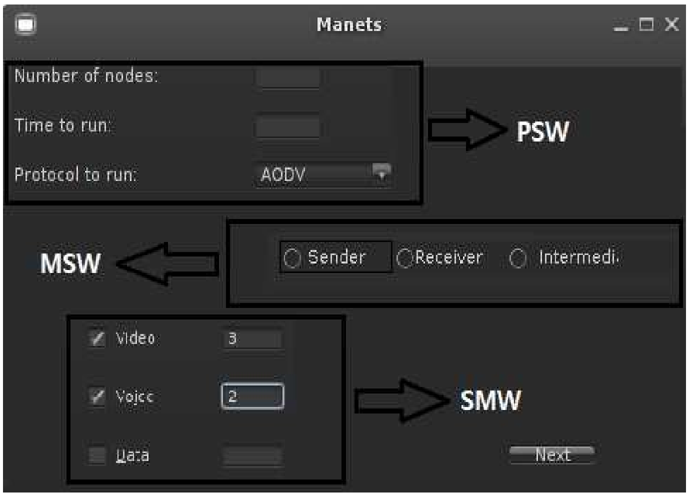
In order to facilitate user interaction, our MANET's testbed is equipped with an intelligent application. The software application was developed in Java NetBeans IDE and several features for testing Adhoc network protocols were incorporated in it. This utility enables the configuration of a multitude of mobility and data models. In order to ensure repeatability, this interactive utility allows the arrangement of various scenarios with specific parameters.

The developed application is broadly categorized into a profile manager and a client application. Profile manager is a centralized configuration tool which allows the user to manipulate every test node in terms of mobility and data model. The purpose of client application is to implement the mode as defined by the profile manager.

3.2.1. Profile Manager

Profile manager is composed of different modules, each having a particular function. The profile manager's function is to build the MANETs test for all the participating nodes rather than being a part of the experiment itself.

On startup, the Parameter Selection Window (PSW) pops up and it requires information regarding the number of nodes, experiment time and testing protocol.



Number of nodes:

Time to run:

Protocol to run: AODV

○ Sender ○Receiver ○ Intermedi

☑ Video 3

☑ Voice 2

□ Data

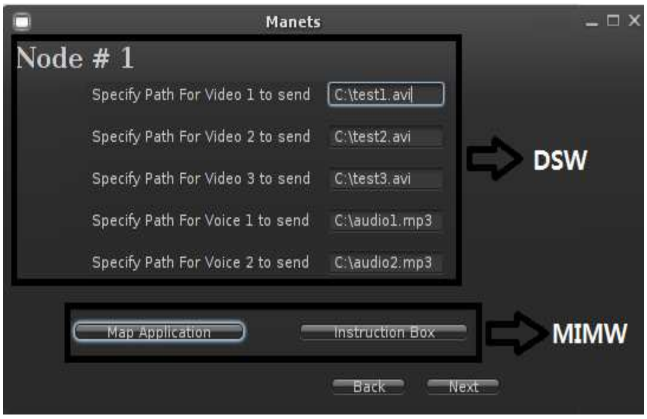
Fig. 6. Parameter, Mode and Model Window in Profile Manager.

The top of figure 6 shows the parameter section window. After filling the selection window fields, the next step is to select which node will act as a sender, receiver or an intermediate node. The next window is named Mode Selection Window (MSW) and is shown below the parameter selection window in figure 6. Multiple senders and multiple receivers can also be selected in the mode selection window.

If a node is selected as a sender node, a new window named Sender's Model Window (SMW) will appear. The user is restricted to select at least one type of data in the model window. The model window for sender nodes is shown at the bottom of figure 6. Multiple data types can be selected at a time. The number of data files for each data type should also be entered in each data field. Parameter Selection Window, Mode Selection Window and Model Window are three separate windows; however, for simplicity, all have been combined in figure 6.

Once a specific data model is selected along with specifying the number of files to be transmitted, the path for each file has to be specified. This feature is implemented by the Data Source Window (DSW). The data source window is shown at the top of figure 7.

After specifying the paths of all the data files, which were specified in the previous steps, the next step is to load the Map and Instruction menu. The bottom of figure 7 shows the Map and Instruction Menu Window (MIMW). Data Source Window and Map & Instruction Window are two separate windows in the actual application. For simplicity, both of these windows have been combined in figure 7.



Node # 1

Specify Path For Video 1 to send C:\test1.avi

Specify Path For Video 2 to send C:\test2.avi

Specify Path For Video 3 to send C:\test3.avi

Specify Path For Voice 1 to send C:\audio1.mp3

Specify Path For Voice 2 to send C:\audio2.mp3

Fig. 7. Data Source and Map and Instruction Menu Window..

To specify a particular mobility model for the participating nodes, the map allocation utility is employed. The purpose of this utility is to create a route to be followed by the mobile nodes. A particular testing region is selected, in our case, the Military College of Signals was chosen as the test area. Different control points were marked on the map and users were given the liberty to join specific points to define the path to be traversed by a particular test node.

Users can select an Offline or an Online Map application. The instruction box is a manual guiding for the nodes about the actions which have to be executed together with the paths they have to traverse in a sequential order. It is in the form of text file and is made available to each and every test node. The instruction box comprises of two components: actions to be performed and movements

Online Map application has been used for testing in the developed testbed and marked points are shown in figure 8.

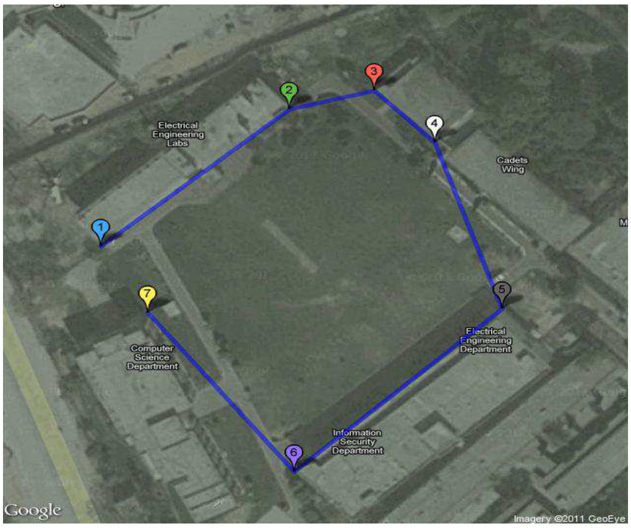
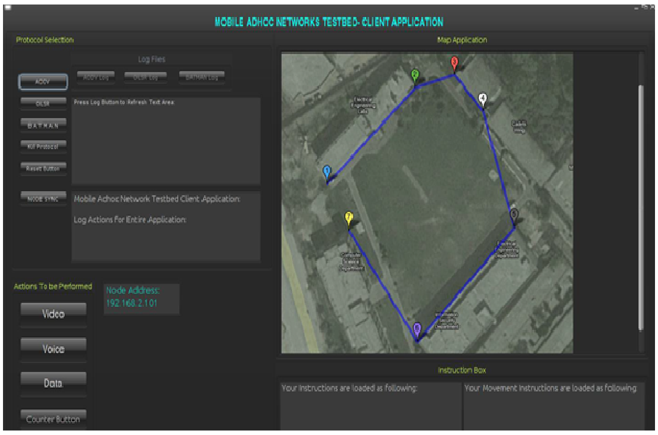

1
2
3
4
5
6
7
Electrical Engineering Labs
Cadets Wing
Computer Science Department
Electrical Engineering Department
Information Security Department
Google
Imagery ©2011 GeoEye

Fig. 8. Online Map Application in Profile Manager.

3.2.2. Client Application

The client Application is running on the nodes, which are part of the Mobile Adhoc Network Testbed. This application provides a generic platform for the clients to execute the profile created by the profile manager. It demonstrates the mobility model in the form of a map, which represents the physical path to be traversed by the node, displays the instructions from the configuration file and provides the necessary options to execute the actions listed. The major components of this application are protocol selection box, actions to be performed, map application and instruction box. Figure 9 shows the layout of the client application developed for the testbed.



Press Log Button to Refresh Text Area:

Mobile Adhoc Network Testbed Client Application:

Log Actions For Entire Application:

Node Address:
192.168.2.101

Your Instructions are loaded as following:

Your Movement Instructions are loaded as following:

Fig. 9. Client Application in MANETs User Application.

3.3. Hardware Implementation

Several customized portable nodes were developed by integrating different hardware components at the hardware implementation phase of the testbed. It comprises of an Intel Atom Development board, DDR3 2GB RAM, Wifi card, Wifi Extender Antenna, SATA 160GB Hard Disk, NiCad battery Pack, Pico PSU-160 ITX power supply, Miniature keyboard, VGA enabled LCD and power supply design. As opposed to standard laptops our device has been manually configured specially for a Mobile Adhoc Networks environment. This configuration consists of the implementation of adhoc routing protocols namely AODV, OLSR and B.A.T.M.A.N. Some of the major hardware components are shown in figure 10.

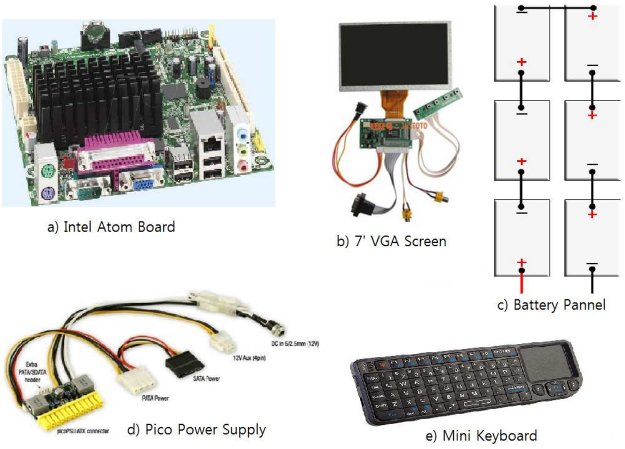


Fig. 10. Major Hardware Components of the Testbed.

The Intel Desktop Board D525MW is designed to support internet-centric computing, delivering incredible capabilities in the new flexible Mini-ITX form factor, featuring the integrated 45nm Intel Atom processor 330 and the Intel 945GC Express Chipset. This board is energy efficient. In order to design a miniature, portable and customized device, DDR3 RAM has been used. D-Link wireless adapter has been used for the purpose of communication between mobile computing nodes. Wifi extender antenna acts as a range extender. It effectively increases the operating distance of the wireless network and conventionally avoids the additional need for power cables or the device clutter.

The Nickel-Cadmium battery is a type of rechargeable battery. The major advantages of NiCad battery are high charge density, durability and enhanced charging discharging cycles. PicoPSU-160-XT High power, 24pin mini-ITX power supply is small efficient package. The developed board operates at 12V; in order to provide this voltage PiscoPSU is being used. It is highly reliable, contains lesser cables and is durable. A miniature keyboard has also been incorporated, in order to make the node portable and so the users could easily carry it.

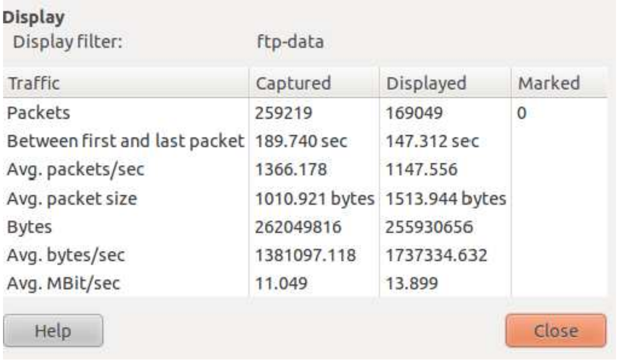
In order to run Intel the atom board and the accessories like a hard disk, Wifi card, VGA card, and control circuitry power supply has been designed to operate on a battery. A battery pack of 16V and 5600mAH is made to power up all these components.

A regulator circuit was used to get 12V, which acts as an input to PicoPSU160. Micrels are high current, high accuracy, low drop out voltage regulators. A Micrel regulator is used. The Mic29302 that is an adjustable regulator with a maximum current of 3A has been used. It is fully protected against high current faults, reversed input polarity, over temperature operations and negative and positive transient voltage spikes.

4. Results and Discussion

4.1. Experimental Setup

A software application was developed in Java NetBeans IDE and several features for testing Adhoc network protocols were incorporated in it. Several customized portable nodes were developed by integrating different hardware components at the hardware implementation phase of the testbed. Ubuntu was set as the operating system for the developed nodes. Multi-hop topologies have been configured by planting the developed nodes at the marked locations by the map allocation. All the nodes have been time synchronized by using the NTP. Different types of data have been transferred between the nodes by utilizing each of the three Adhoc network protocols explained in section 2. FTP has been used for transferring the files from the source to the destination. The VLC has been used for transferring video data from the source to the destination by using the application described in section 3. Wireshark has been used to analyze the data and to calculate the different performance parameters, explained in the following section. Information window of Wireshark is shown in figure 11.



Display

Display filter: ftp-data

Traffic	Captured	Displayed	Marked
Packets	259219	169049	0
Between first and last packet	189.740 sec	147.312 sec	
Avg. packets/sec	1366.178	1147.556	
Avg. packet size	1010.921 bytes	1513.944 bytes	
Bytes	262049816	255930656	
Avg. bytes/sec	1381097.118	1737334.632	
Avg. MBit/sec	11.049	13.899	

Fig. 11. *Wireshark Display Window.*

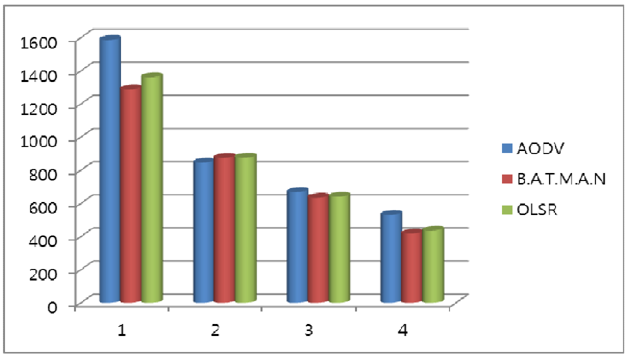


Fig. 12. *Throughput Vs Number of Hops*

4.2. Throughput

Throughput and Packet Loss has been computed by the changing the number of hops. Table 3 shows the performance of each of the three protocols in terms of throughput and percent packet loss as a function of the number of hops from the source to the destination. X-axis shows the number of hops while throughput and packet loss is on the Y-axis of figure 12 and figure 13 respectively.

Figure 12 shows the throughput versus the number of hops for AODV, OLSR and BATMAN protocol. It is evident from the graph that the performance of AODV degrades as the number of nodes increases, while OLSR and BATMAN show a gradual degradation. Being reactive protocol AODV consumes large part of the bandwidth for transferring control information, while OLSR and BATMAN only transfers the control messages once they are requested.

Table 3. *Throughput and Error Rate with variable Hops*

Number Of Hops	Protocol	Parameter	Values
1	AODV	Throughput	1586.6
		Error Rate	0.333
	OLSR	Throughput	1361.75
		Error Rate	0
	BATMAN	Throughput	1288.5
		Error Rate	0.333
2	AODV	Throughput	845.95
		Error Rate	4.33
	OLSR	Throughput	876.49
		Error Rate	1
	BATMAN	Throughput	875.2
		Error Rate	1.678
3	AODV	Throughput	668.65
		Error Rate	7
	OLSR	Throughput	640.35
		Error Rate	2.33
	BATMAN	Throughput	34.15
		Error Rate	2
4	AODV	Throughput	530
		Error Rate	8.677
	OLSR	Throughput	436.05
		Error Rate	2.678
	BATMAN	Throughput	417.9
		Error Rate	2.678

4.3. Packet Loss Rate

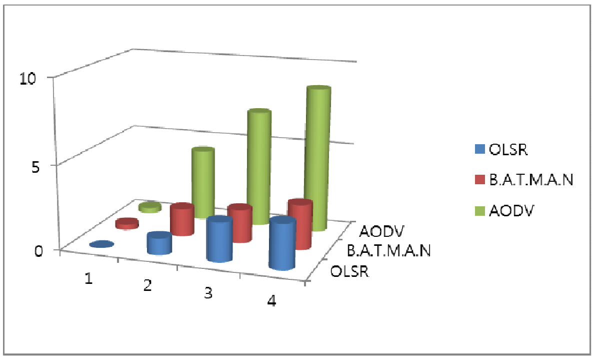


Fig. 13. *Packet Loss Rate Vs Number of Hops*

Figure 13 shows the packet loss rate, calculated by using equation (6) versus the number of nodes. It can be observed in figure 13 that the percent error rate propagates exponentially for AODV, while it increases on a linear fashion for OLSR and BATMAN with an increase in the number of nodes. Due to the reactive nature of AODV with the increase in the number of nodes, the quantity of control packets also increases and the path get congested. While in the case of OLSR and BATMAN, the control information only exchanges upon request.

4.4. Overhead Packets

Number of overhead packets and end to end delay has been compared against pause time. Pause time is equal to the time waited by a particular node after arriving at the destination provided as a mobility plan by the developed application. Figure 14 shows the number of overhead packets transferred verses the pause time in seconds. Pause time is on the X-axis and number of overhead packets is on the Y-axis.

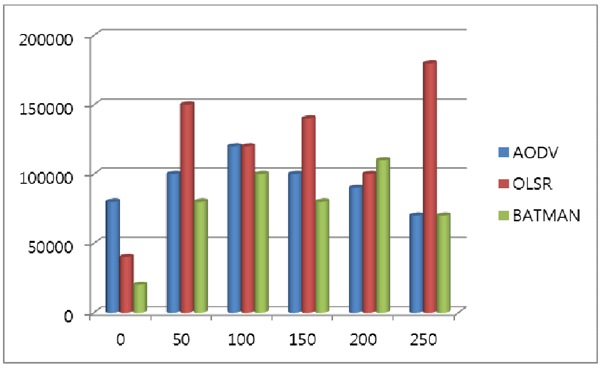


Fig. 14. *Number of Overhead Packets Vs Pause Time (sec)*

It can be deduced from the graph that BATMAN shows the minimum number of control packets. AODV shows a gradual increase initially and then a decrease. OLSR shows a higher number of overhead packets compared to other two protocols. Decrease in control packets after 50 seconds for AODV indicates that the AODV settles after 50 seconds. BATMAN has a built-in capability to control the number of overhead packets. In the case of OLSR, it can be concluded that overhead doesn't only depend on the pause time, but also depends on several factors like a change in the configuration of the network, the speed of moving nodes and many other factors. X-axis in Figure 15 shows the pause time while Y-axis represents the End to End delay. AODV shows a zigzag behavior while BATMAN and OLSR show a gradual decrease with increase in pause time.

4.5. End to End Delay

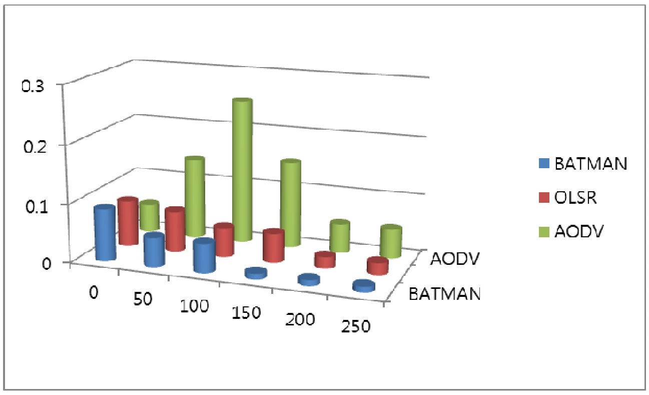


Fig. 15. *End to End Delay Vs Pause Time (sec)*

The zigzag behavior of AODV is because of the overhead load due to the proactive nature of the protocol. In the case of OLSR, when pause time is increased, the link between the source and destination becomes stable. Due to the stable connection, the end to end delay decreases. Same is the case for BATMAN. Due to the inherit overhead reduction capability of BATMAN, reduction in end to end delay is more rapid as compared to OLSR.

Table 4. *Overhead and End to End Delay with variable Pause Time*

Protocol	Parameter	Pause Time	Values
AODV	Over Head	0	80000
		50	100000
		100	120000
		150	100000
		200	90000
		250	70000
	End to End Delay	0	0.05
		50	0.14
		100	0.25
		150	0.15
		200	0.05
		250	0.05
OLSR	Over Head	0	40000
		50	150000
		100	120000
		150	140000
		200	100000
		250	180000
	End to End Delay	0	0.08
		50	0.07
		100	0.05
		150	0.05
		200	0.02
		250	0.02
BATMAN	Over Head	0	20000
		50	80000
		100	100000
		150	80000
		200	110000
		250	70000
	End to End Delay	0	0.09
		50	0.05
		100	0.05
		150	0.01
		200	0.01
		250	0.01

4.6. Average Jitter

Average Jitter is shown in the Y-axis while packet size is on the X-axis of figure 16. By analyzing figure 16, it can be concluded that AODV is more prone to jitter as compared to OLSR and BATMAN. Table 5 shows the performance of each protocol against the size of the packet.

OLSR shows little variation in performance in terms of jitter, while AODV and BATMAN show almost steady performance.

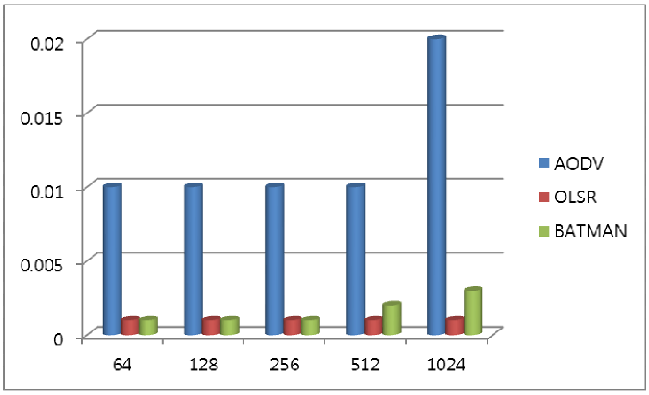


Fig. 16. *Average Jitter Vs Packet Size*

These types of deductions can be valid for all other protocols, which are tested by using the developed testbed in this paper.

Table 5. *Average Jitter with variable packet size*

Protocol	Packet Size	Average Jitter
AODV	64	0.01
	128	0.01
	256	0.01
	512	0.01
	1024	0.002
OLSR	64	0.001
	128	0.001
	256	0.001
	512	0.001
	1024	0.001
BATMAN	64	0.001
	128	0.001
	256	0.001
	512	0.002
	1024	0.003

By analyzing all the results of all the five performance metrics, it can be deduced that AODV performs much better in networks with static traffic. OLSR and BATMAN function appropriately in conditions where traffic is highly dense and sporadic, but scalability acts as a limiting constraint as the network size increases. Heavy flooding of control packets is observed in the case of AODV upon route discovery. In the case of OLSR, there is a non-linear increase in the size of the route table and control messages may block the actual data transmission.

5. Conclusion

The developed testbed provides a generic platform for the testing of Adhoc routing protocols in real world scenarios. The developed testbed is capable of implementing and testing Adhoc routing protocols in real life scenarios in order to test the viability of the newly developed protocols. A software tool has been developed in Java NetBeans IDE to facilitate the users carrying out different tests under same constraints, thus ensuring the repeatability and helping out in finding the exact pros and cons of the protocols being tested. The testbed is equipped with a customized, flexible and dedicated hardware, which is specially designed by integrating several hardware components for Adhoc Networks in contrast to the standard laptops or testing devices. All necessary parameters for evaluating the performance of a newly developed protocol are included in the testbed. The developed testbed is capable of testing newly developed Adhoc network protocols' viability, robustness and efficiency, to assess their key features, conquer their shortcomings and add more appealing features.

Test results have proven the developed testbed as a reliable, efficient and robust testbed for real time testing of newly developed protocols before using them in real life applications.

References

[1] Pan Li, Yuguang Fang, Jie Li "Throughput, Delay, and Mobility in Wireless Ad Hoc Networks" IEEE Communications Society, Technical Program 2010.

[2] Matthew Brand, Petar Maymounkov, and Andreas F. Molisch "Routing with probabilistic delay guarantees in wireless ad-hoc networks".

[3] Humayun Bakht "Survey of Routing Protocols for Mobile Ad-hoc Network" International Journal of Information and Communication Technology Research Volume 1 No. 6, October 2011.

[4] Erik Nordstrom, Per Gunningberg, Henrik Lundgren "A Testbed and Methodology for Experimental Evaluation of Wireless Mobile Ad hoc Networks".

[5] Yongguang Zhang, Wei Li "An Integrated Environment for Testing Mobile Ad-Hoc Networks" EPFL Lausanne, Switzerland 2002.

[6] Chandra Kanta Samal "TCP Performance through Simulation and Testbed in Multi-Hop Mobile Ad hoc Network" International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, July 2010.

[7] Ian D. Chakeres, Elizabeth M. Belding-Royer "AODV Routing Protocol Implementation Design".

[8] P. Jacquet, P Muhlethaler, T Clausen, A Laouiti, A Qayyum, A Viennot "Optimized Link State Routing Protocol for Adhoc networks".

[9] Laurent Delosi`eres and Simin Nadjm-Tehrani "BATMAN Store-and-Forward: the Best of the Two Worlds "Second International conference on Pervasive networks for emergency managements 2012.

[10] Sagar Sanghani, Timothy X Brown, Shweta Bhandare, Sheetalkumar Doshi "EWANT: The Emulated Wireless Ad Hoc Network Testbed" IEEE, 2003.

[11] Virendra Singh Kushwah1, Kamal Kumar Chauhan and Amit Kumar Singh Sanger "A Comparative Study of Mobile Ad Hoc Network Protocols for Throughput, Average End-to-End Delay and Jitter" International Journal of Computational Intelligence Research Volume 6, Number 3, 2010.

[12] David A. Maltz Josh Broch David B. Johnson”Quantitative Lessons From a Full-scale Multi-Hop Wireless Ad Hoc Network Testbed” IEEE, 2000.

[13] Joni Birla, Basant Sah, “Performance Metrics in Ad-hoc Network”. International Journal of Latest Trends in Engineering and Technology. Vol. 1 Issue 1 May 2012.

[14] Laurent Viennot, Philippe Jacquet and Thomas Heide Clausen”Analyzing Control Traffic Overhead versus Mobility and Data Traffic Activity in Mobile Ad-hoc Network Protocols”.

[15] Ahmed A. Radwan, Tarek M. Mahmoud and Essam H. Houssein “Performance Measurement of Some Mobile Ad Hoc Network Routing Protocols” IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 1, January 2011.

[16] Soumendra Nanda, Zhenhui Jiang, David Kotz “A Combined Routing Method for Wireless Ad Hoc Networks” Dartmouth College Technical Report TR2007-588, June 2007.

[17] Rakesh Kumar, Manoj Misra, Anil K. Sarje “A Simplified Analytical Model for End-To-End Delay Analysis in MANET” IJCA Special Issue on “Mobile Ad-hoc Networks” MANETs, 2010.

[18] Kamal Kumar Sharma, Hemant Sharma and A. K. Ramani “Modeling and Analysis of End-To-End Delay for Ad Hoc Pervasive Multimedia Network” Proceeding of the international multi conference of engineers on computer scientists, Hong Kong , 2010.

[19] Swati Bhasin, Ankur Gupta, Puneet Mehta “Comparison of AODV, OLSR and ZRP Protocols in Mobile Ad-hoc Network on the basis of Jitter” International Journal of Applied Engineering Research Vol.7 No.11 2012.

[20] R. Marutha Veni, R. Latha”Mobile Ad hoc Network” International Journal of Science and Research, Volume 2 Issue 4, April 2013.

Biography

Akhtar Hussain received his B.E degree in Telecommunications from National University of Sciences and Technology (NUST) Pakistan in 2011 and M.S in Power Systems from Myongji University, Korea, in 2014. Currently he is working as an associate engineer in SANION; IEDs development company, in Korea. His research interests are power system automation and protection, smart grids, communication in power systems and wireless communications.

Aimel Khan received his Bachelors in Telecommunication’s Engineering from National University of Sciences and Technology (NUST) Pakistan in 2011. Currently he is working as a Value Added Services professional in the Telenor Group, a global international provider of voice, data, and content and mobile communication services in 13 markets. His interests include Computer Networks, Mobile Communications and Network Security.

Abdul Rehman Qaiser received his Bachelors in Telecommunication’s Engineering from National University of Sciences and Technology (NUST) Pakistan in 2011. Currently he is working in GSM core networks for Mobilink Pakistan, a VimpleCom subsidiary which is a global provider of telecommunication services in 18 countries. His interests include Mobile communications, international roaming and computer networks.

Muhammad Mohsin Akhtar is a telecommunication professional currently engaged in the Value Added Services Operational Domain. He received his B.E degree from National University of Sciences and Technology (NUST), Pakistan in 2011. His interest areas include Mobile Communication Systems, Networks and Security.

Obaidullah Khalid is a PhD student at Queen Mary, University of London, UK and Alpen-Adria Universitat Klagenfurt, Austria working under the supervision of Prof. Andrea Cavallaro and Prof. B. Rinner. Mr. Khalid worked at the National University of science and Technology (NUST) as a lecturer/researcher from 2008-2012. He received his M.S degree in Network Engineering from Illinois Institute of Technology (IIT), Chicago, USA in 2008. His research interests include visual tracking, data fusion, performance evaluation and wireless networks.

Muhammad Faisal Khan received his PhD and MS degree in Electrical and Computer Engineering from Georgia Institute of Technology, USA in 2004 and 2008 respectively. He did his BS from University of Engineering and Technology, Lahore, Pakistan in 1994. He is working as an Assistant Professor at College of Telecommunication Engineering (MCS), a constituent college of National University of Sciences and Technology (NUST), since Feb 2009. His research interests include broadband access networks, wireless networks, NGN, video processing and image processing.