

Implementing Voice over Internet Protocol
On Mobile Ad-hoc Network and
Analyzing its Features Regarding Efficiency and Security



Submitted by:

Naveed Ahmed Sheikh

2010-NUST-MS PhD-Elec (Comm-N)-20

Supervisor:

Dr Athar Mahboob

Thesis

Submitted to

Department of Electronic and Power Engineering,
College of Marine Engineering (PNEC), Karachi
National University of Sciences and Technology, Islamabad

In partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE IN ELECTRICAL ENGINEERING
With Specialization in Communications

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

In the name of ALLAH,
the most Beneficent, the most Merciful

ACKNOWLEDGEMENTS

First and foremost I would like to offer my sincerest gratitude to Almighty ALLAH for his kindness and blessings that he has always bestowed upon me. I pray that, ALLAH shower his countless blessings and peace upon his most beloved and last Prophet Hazrat Muhammad Mustafa (S.A.W).

I would also like to express my special thank to my parents, brother and other beloved ones for supporting me throughout my study and showing great patience during this research work.

After that, I would like to express my sincere gratitude to my supervisor Dr. Athar Mahboob for his kind and continuous guidance with great patience and knowledge. I cannot express in words the support, encouragement and motivation he has provided throughout the work and without him it could be very difficult to complete the work and achieve desired results.

At last but not least, I would like to thank all of Guidance Committee members.

- Dr. Pervaiz Akhter
- Dr. Arshad Aziz
- Dr. Khawaja Bilal Mahmood

ABSTRACT

Ad hoc networks have immensely helped wireless communications to grow. The VoIP technology has been a hot issue in the Information Technology (IT) industry for more than ten years now. Voice over Internet Protocol (VoIP) is a technology that allows users to make telephone calls, i.e., interactive two-way voice communication using a broadband Internet connection instead of an analog phone line. Providing a secure real-time VoIP service on the MANET is the main design objective of this thesis. We have developed a prototype system that establishes secure VoIP communication and will provide an extremely flexible method for VoIP in mobile ad hoc networks. We have established a cooperative mesh based mobile Ad-hoc networks for rapidly deployable VoIP communication with survivable, efficient dynamic networking.

ACKNOWLEDGEMENTS	III
ABSTRACT	IV
LIST OF FIGURES.....	IX
LIST OF TABLES	XI
CHAPTER 1:.....	1
INTRODUCTION	1
1.1 Motivation:	1
1.2 Thesis Scope:	3
1.3 Related Work:	4
1.4 Thesis Organization:.....	4
CHAPTER 2:.....	6
MOBILE AD HOC NETWORK	6
2.1 Introduction:	6
2.2 MANET Features:	6
2.3 Result of Limited MANET Resources:	8
2.4 Performance of Routing Protocols in MANET:	9
CHAPTER 3:.....	11
ROUTING PROTOCOLS FOR MANETS	11
3.1 Routing protocol.....	11
3.2 Static routing	12
3.3 Adaptive routing	12
3.4 Distance-vector routing protocol	13

3.5	Link-state routing protocol	13
3.6	Link-State versus Distance Vector	13
3.7	Optimized Link State Routing Protocol.....	14
3.8	Characteristic explicit to OLSR.....	14
3.9	Messages.....	15
3.10	Benefits.....	16
CHAPTER 4:.....		17
VOICE OVER INTERNET PROTOCOL		17
4.1	Introduction:	17
4.2	Packet Switching:.....	18
4.3	Advantages of using VoIP:.....	19
4.4	Disadvantages of using VoIP:	19
4.5	Codecs & Soft Switches:.....	20
4.6	VoIP Call Monitoring:.....	21
4.7	Latest Trends in VoIP:	22
CHAPTER 5:.....		24
SECURITY ISSUES IN MANETS		24
5.1	MANET Security Services/ Vulnerability:	24
5.2	Attacks on MANET:	24
5.3	Passive Attacks	25
5.4	Active Attacks.....	25
5.5	Counter Measures:.....	26
CHAPTER 6:.....		27

ANDROID.....	27
6.1 Introduction:	27
6.2 Android Look and Feel:	27
6.3 Android Architecture:	28
6.4 Android Community:	29
CHAPTER 7:.....	31
IMPLEMENTATION.....	31
7.1 Implementation of System:	31
7.1.1 User Defined IP:.....	31
7.1.2 OLSR:	32
7.1.3 Neighbour recognition:.....	32
7.2 Configuration of Systems:.....	33
7.2.1.1 Setup on PC for Real Time Monitoring:.....	34
7.2.1.1 Installing OLSR on Ubuntu	34
7.2.1.2 Configuring OLSR.....	35
7.2.1.3 Configuring OLSR Plugins	37
7.2.1.4 Shifting Ubuntu to Ad-hoc Mode	38
7.2.1.5 Execute OLSR	39
7.2.1.6 Execute Mapping of OLSR Routes.....	39
7.2.2.1 Setup on Android:.....	41
7.2.2.2 Rooting android phone:.....	41
7.2.2.3 Rooting Samsung Galaxy Y GT-S5360	42
7.2.2.4 Rooting Samsung Galaxy S II GT-I9100	42
7.2.2.5 Rooting Ainol Tablet Aurora 2.....	43

7.2.2.6	Install Android Terminal Emulator	43
7.2.2.7	Install BusyBox.....	43
7.2.2.8	Build OLSR from PC for Android	44
7.2.2.9	Configure OLSR for Android.....	45
7.2.2.10	Install Custom Wifi Tether	45
7.2.2.11	Installing PTTDroid.....	46
7.2.2.12	Execute Wifi Tether	46
7.2.2.13	Execute PTTDroid	46
7.3	Securing the Systems:	46
7.3.1	CSipSimple	47
7.3.2	Stream Cipher	47
7.3.3	IPSec Tunneling	48
7.3.4	Securing the Network	49
7.3.4.1	Functioning and explanation.....	49
CHAPTER 8:	50
RESULTS:	50
CHAPTER 9:	70
CONCLUSION:	70
CHAPTER 10:	71
FUTURE WORK	71
REFERENCES:	72

List of Figures

Figure 1: Conceptual View	4
Figure 2: Hello	15
Figure 3: Topology control	15
Figure 4: Android Architecture	29
Figure 5: Implementation from Existing System	33
Figure 5: Neighbor Recognition	35
Figure 7: Implementation Setup 1 - on PC	36
Figure 8: Example 1 Topology View	42
Figure 9: Example 2 Topology View	42
Figure 10: Implementation Setup 2 - on Android	43
Figure 11: CSipSimple Issue	49
Figure 12: AES Encryption Using Speex Codec	50
Figure 13: Test Case Scenario 1	53
Figure 14: Ping Distance Graph	54
Figure 15: ICMP Packet Detail	54
Figure 16: Throughput Distance Graph	55
Figure 17: Throughput Distance Reading	55
Figure 18: Bandwidth Distance Graph	56
Figure 19: Bandwidth Distance Reading	56
Figure 20: Jitter Distance Graph	57
Figure 21: Jitter Value Reading	57
Figure 22: Signal Strength Graph	58
Figure 23: Signal Strength Reading	58
Figure 24: Packet Loss Distance Graph	59
Figure 25: Packet Loss Reading	59

Figure 26: Sound Quality at 3.95 kbps.....	60
Figure 27: VoIP Packet at 3.95 kbps.....	60
Figure 28: Sound Quality at 24.6 kbps.....	61
Figure 29: VoIP Packet at 24.6 kbps.....	61
Figure 30: Sound Quality without CODEC.....	62
Figure 31: VoIP Packet without CODEC.....	62
Figure 32: Test Case Scenario	63
Figure 33: Ping Distance Graph	64
Figure 34: ICMP Packet Detail.....	64
Figure 35: Throughput Distance Graph	65
Figure 36: Throughput Distance Reading	65
Figure 37: Bandwidth Distance Graph.....	66
Figure 38: Bandwidth Distance Reading.....	66
Figure 39: Jitter Distance Graph.....	67
Figure 40: Jitter Value Reading	67
Figure 41: Signal to Noise Ratio Graph	68
Figure 42: Signal to Noise Ratio Reading	68
Figure 43: Packet Loss Distance Graph	69
Figure 44: Packet Loss Reading	69
Figure 45: Sound Quality at 3.95kbps	70
Figure 46: VoIP Packet at 3.95kbps	70
Figure 47: Sound Quality at 24.6 kbps.....	71
Figure 48: VoIP Packet at 24.6kbps	71
Figure 49: Sound Quality without CODEC	72
Figure 50: VoIP Packet without CODEC	72

List of Tables

Table 1: Jitter Value	57
Table 2: Packet Loss	59
Table 3: MOS.....	60
Table 4: Jitter Value	67
Table 5: Packet Loss	69
Table 6: MOS.....	70

CHAPTER 1: INTRODUCTION

1.1 Motivation:

Mobile Ad hoc Networks (MANET) are self-organizing networks and do not require a fixed infrastructure. The nodes of a mobile ad hoc network have to self-configure in some arbitrary manner in order to perform their function using the network or to provide user with application services utilizing the network. The areas with depleted fixed infrastructures can be a place where a natural disaster has occurred such as an earth quake, flood or a chaotic situation has been created by humans as in a battle field. These networks are becoming wide-spread between mobile entities such as persons or vehicles. These entities can directly communicate with each other because either they are in the required radio range or otherwise multi-hop routing is used for communication purposes. Since the nodes may be in continuous motion and may move into and out of the radio range, frequent breakage of data links can occur resulting in high vulnerability of wireless link between nodes. Routing information also changes so the topology of the network is highly dynamic. There is also a bandwidth constraint in such a wireless network because it lacks access points and high-speed wired links. All the nodes operate on battery power which is usually very limited resulting in a need for energy for efficient operation. Furthermore, since the current routing protocols do not focus much on the security aspects, mobile ad hoc networks are also more vulnerable to security threats as compared to traditional wired networks [1].

Voice over Internet Protocol (VoIP) is a technology that allows users to make telephone calls, i.e., interactive two-way voice communication using a broadband Internet connection instead of an analog phone line. VoIP holds great promise for lowering the cost of telecommunications and increasing the flexibility for both businesses and individuals. VoIP leverages existing IP-based packet-switched networks to replace the circuit-switched networks used for voice communications ever since the invention and deployment of the telephone system. The VoIP infrastructure consists of endpoints (VoIP telephones), control nodes, gateway nodes, and the IP-based network. The IP network can utilize various media including Ethernet, optical fiber and

wireless links. The VoIP system interacts with both local and remote VoIP phones using the intranet and Internet as well as interacting with phones connected to the public-switched telephone network (PSTN) through gateways [2].

The VoIP technology has been a hot issue in the Information Technology (IT) industry for more than ten years now. Nowadays people often use analog voice communication devices (radios) in a local area. But these voice radios have an inherent disadvantage: the frequency band used for communication is open to everyone and shared with different users. So if there is more than one communication group existing, they might interfere with each other by using the same frequency occasionally or purposely—people would hear other person's talk. Because VoIP technology converts voice signals to data packets and transfers them on the network as IP packets where packet has its destination address, so other nodes cannot ignore this packet. By this way different communication groups can avoid interfering with each other. Furthermore, we can make use of the network security procedures such as encryption to improve the safety of communication further. Hence it is highly desirable and necessary to implement VoIP in portable MANET equipment such as PDAs, Notebook PCs, etc and use these equipments as communication tools in the MANET [3].

Providing a secure real-time VoIP service on the MANET is the main design objective of this thesis. It is perceived to be a difficult task due to restrictions in device resources, adverse properties of the wireless channel, dynamic topology and the lack of central administration. Also the flexibility of the VoIP system and the convergence of voice and data networks bring in additional security risks. All devices in an IP network become potential active or passive adversaries. It can be seen that the underlying IP data network facility that a VoIP system relies on complicates the security assurance requirement. Hence security issues are still at large in MANETs. Because of the highlighted limitations, it is a challenge to make a secure VoIP feasible over MANET.

The motivation to pursue this thesis is due to the following open items in existing techniques employed for security. The open-dynamic MANET routing protocols employed till date have their own tradeoffs: ubiquitous communication is provided at the expense of security. Proactive RSA and Distributed Key Sharing may provide some levels of security but require a trusted dealer to authenticate the initial group and establish the authenticated bulletin board. Secure

Routing protocol using AODV built on IPV6 (SecAODV) cannot prevent attacks that may take advantage of Message Authentication Codes (MAC) vulnerabilities. Secure Route Discovery Process (SRDP) protocol assumes bidirectional communication and relies on the assumption that source and destination are honest which is not realistic. Proactive Distributed Signature (PDS) does not provide any specific protocol and requires secure recovery protocol. Authenticated Routing Protocol (ARAN) requires trusted certification server to issue the initial certificates. Emulated Centralized Certification Authority (CA) assumes that all nodes are honest and not selfish or malicious to ensure correct and secure distribution of key shares. Distributed Authentication Scheme (DAS) restricts MANET to employ a centralized mechanism for authentication. Unified Network Layer Security Solution implements intrusion reaction without specifying who issues the tokens to make them secure enough. Secure AODV and Secure Routing Information cannot prevent tunneling attack and provides authority to provide nodes with valid certificates.

1.2 Thesis Scope:

Our objective in this work is to develop a prototype system which will establish secure VoIP communication and will provide an extremely flexible method for VoIP in mobile ad hoc networks. We will attempt to establish cooperative mesh based mobile adhoc networks for rapidly deployable VoIP communication with survivable, efficient dynamic networking [4].

Applications of our proposed work include industrial or commercial applications including existing and future military networking requirements for robust, IP-compliant data services for mobile wireless communication networks [5], public safety rescue operations, vehicular ad hoc networks, intelligent transportation system, intelligent home environment and monitoring wildlife, aquatic environment and health systems [6]. The figure given below shows a conceptual view of our desired objective.

This figure shows few MANET nodes, the establishment of mesh and relay based internetwork using MANET routing protocols, establishment of secure communication channel, and use of the MANET for VoIP by human users.

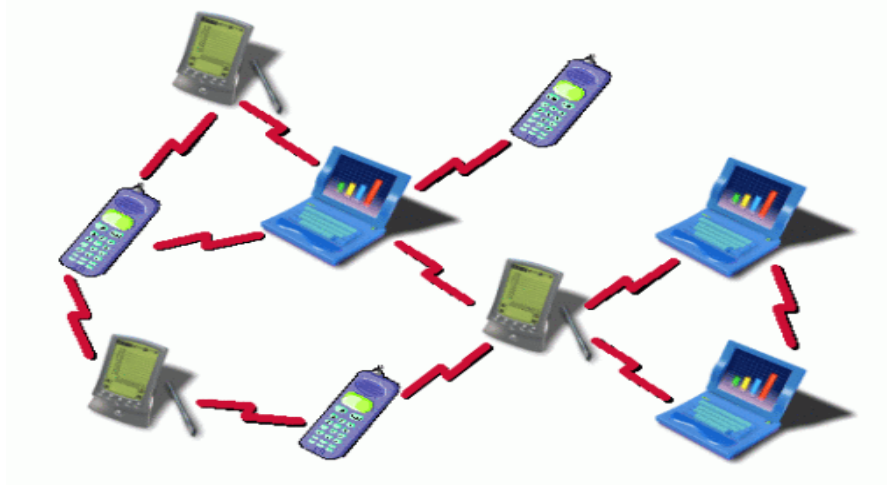


Figure 1: Conceptual View of Secure VoIP in MANETs

1.3 Related Work:

Commotion android is a team that is working on similar lines. The characteristics of their system include tethering and executing OLSR protocol. It has a user friendly GUI. However, the shortcoming of their system hinders them from being famous. These include failure of proper execution of tethering application. Consequently OLSR protocol fails to execute as well.

There is another application developed that is freely available on Google Play named MANET Manager. Its functions include tethering and executing OLSR again. Its GUI is very user friendly. However it only supports WEXT Kernel resulting in appliance over a limited number of android sets.

1.4 Thesis Organization:

This thesis is organized as follows. We start off with the introductory chapter that reveals the motivation and scope of work for the thesis. In chapter 2 we discuss the peculiarities of MANETs which result in challenges for VoIP implementation. Chapter 3 covers the discussion of the state of the art routing protocols for the MANETs. This discussion will allow us to choose the best routing protocol for our end objectives. In Chapter 4 we review the fundamentals of VoIP. Our objective is to identify the important elements of the VoIP system which we will use in our implementation. In chapter 5 we discuss the security issues in MANETs and possible

solutions to these. In chapter 6 we discuss the Linux-based Android operating system platform which we have used on MANET nodes to establish our required system. In chapter 7 we discuss the details of our implementation. In chapter 8 we present the performance results and security analysis of our implemented system. Finally we provide conclusions in chapter 9 and provide possible directions for future work in Chapter 10.

CHAPTER 2: MOBILE AD HOC NETWORK

2.1 Introduction:

Ad hoc networks have immensely helped wireless communications to grow to an extent that they are widely used in a lot of areas including battlefields, conference or disaster relief environments. Ad hoc networks share the same issues as mobile communications and a wireless network, as ad hoc networks, has them at its roots. These issues comprise of power control, optimization of bandwidth, quality of transmission etc. Since Ad hoc networks lack fixed infrastructure and rely on multiple hops, it introduces challenges in form of device recovery, topology maintenance and network configuration. When wireless mobile nodes are added together that have routing abilities, an arbitrary graph is formed resulting in a Mobile Ad Hoc Network. Each node is made intelligent enough to create its own autonomous network and topology regardless of infrastructure in a standalone way. They can also become a part of another infrastructure. These networks do not rely on base stations or access points like traditional mobile networks, so they prove to be useful for emergency and disaster prone situations. Multiple classes of Mobile Adhoc Networks have evolved lately e.g. VANET's (Vehicular Ad Hoc Networks), InVANET (Intelligent Vehicular Ad Hoc Networks), iMANET (Internet based Mobile Ad Hoc Networks) etc.

2.2 MANET Features:

There are seven basic characteristics of a MANET each of which is defined below [7]:

1) Autonomous Terminal

In Mobile Ad Hoc networks, nodes deliver a dual purpose. They may serve as a host and as a router. This feature makes it indistinguishable to see what end points are and what switches are. So nodes in Mobile Ad Hoc Networks have a unique ability to perform switching functions of a router and hence it makes the node autonomous to decide when to perform what function.

2) Distributed Operation

Nodes in a network do not rely on central management as the concept does not apply there for MANET's. Each node is responsible and control is distributed among all nodes to collaborate among them and act as switches whenever needed. Such a distribution of control and management allows these nodes to perform routing and implement security aspects.

3) Multi Hop Routing

MANET's can implement either of the transferring strategy i.e. single hop or multi hop. The structure is simple for single hop and it is simple in its function too. Single hop transferring strategy offers limited functionality and hence has limited applications. Multi hops are required when data packets need to be delivered outside the network range as more than one intermediate nodes are required to reach the destination.

4) Dynamic network Topology

It is a basic characteristic of a Mobile Ad hoc Network to keep changing its network topology since the nodes are mobile and so the connectivity among terminal nodes vary from time to time. The autonomous nature of MANET helps it to adapt to the traffic and propagation conditions and the patterns of mobility of the mobile network nodes. The nodes in a MANET develop their own network dynamically by establishing routing with other nodes as they move. There is a challenge that occurs here when a user wishes to access a public fixed network from a Mobile Ad hoc Network.

5) Fluctuating Link Capacity

A Mobile Ad hoc Network channel is more prone to interference, fading and noise A single end to end path is shared among several sessions so it has a limited bandwidth as compared to its counterpart of a fixed network. The data bit errors are also high in them. The paths between two nodes can be multiple so it results in fluctuating capacity that links can maintain.

6) Energy Constrained Operation

The computing and communication functions in a MANET have to be very efficient as the power is constrained in devices that build up a MANET. It is a huge challenge to devise optimized algorithms and techniques that consume less power to allow the user to sustain in the network for a prolonged period of time.

7) Limited Physical Security

Nodes in a MANET are dynamically linked and hence security breach is a common issue in Mobile Ad hoc networks. The possibility of spoofing, eavesdropping and denial of service attacks need to be properly addressed to make it reliable for use.

2.3 Result of Limited MANET Resources:

The resources in a MANET are limited. The structure allows addition of infinite nodes and thus the resources get constrained. There are three major issues that users face in such a scenario [7]:

1) Selfish Nodes

MANET is powerful only when nodes cooperate among themselves. Since nodes are autonomous entities, they can deny services to other users and use resources of other nodes to meet their own goal. This behavior leads to performance degradation as the resources in MANET are limited. Nodes need to collaborate and cooperate to avail resources and save time and energy. To forward packets and to detect routes, a lot of network bandwidth, CPU time and memory are consumed. So it results to be a cost intensive activity for each mobile node. To reduce this cost, collaboration among nodes is essential.

2) Denial of Service

This issue is also due to limitation of resources in Mobile Ad hoc networks. There are algorithms that are used to avoid hogging of resources by a single node. Such algorithms are applicable where there is a fixed infrastructure to implement denial of service attacks. Mobile Ad hoc network does not have a fixed infrastructure and this requires a degree of cooperation from all nodes. Now if a node is taking more time to execute its processes or is taking up

resources for a longer period of time, techniques need to be employed that can ensure starvation of resources does not occur for other nodes. Cryptographic techniques are employed but they are very costly where no infrastructure is present. It allows resolution by identifying neighbors first.

3) Malicious Nodes

The network layer of a Mobile Ad hoc network is attacked frequently by malicious nodes. They try dropping data packets or route them to different places that are not the correct destinations. They also give denial of service by not accepting to forward packets to other neighboring nodes. In the worst case the malicious nodes also modify data and the intended information is not passed to the destination nodes. Malicious nodes also have the tendency to damage other nodes in a network.

2.4 Performance of Routing Protocols in MANET:

There are 4 major routing protocols that are used in MANET. Each of them is discussed below:

1) DSR (Dynamic Source Routing)

DSR maintains and discovers routes between nodes. It is a reactive protocol. In the route discovery process DSR sends the route request packet to the network. It is a process for every node that gets this packet to save its address to the route request packet and then forward it to the next node. This is done for the purpose of keeping a track between sender and receiver. When the packet makes it to its destination, an acknowledgement is sent to the sender. Similarly if a failure is detected, error packet is sent to the sender indicating an error in network node or data packet. DSR relies on 3 steps at different levels [8]:

- Link Layer Acknowledgement
- Passive Acknowledgement
- Network Layer Acknowledgement

Results of researches show that when Media Access Delay, Throughput, Delay, Load and Network Load are considered, DSR is the worst choice [8].

2) OLSR (Optimized Link State Routing)

OLSR relies on the methodology of maintaining a table to store routes. It keeps the table updated and hence delay to fetch a route is minimal. OLSR optimizes flooding mechanism by introducing multi point relays to broadcast packets. This technique makes OLSR a proactive protocol as it maintains information in advance.

Results of researches show that when Media Access Delay, Throughput, Delay, Load, and Network Load are considered, OLSR returns a balanced value [8].

3) AODV (Ad Hoc on Demand Distance Vector Routing)

AODV is a reactive protocol. It maintains a routing table as well but in case the route information is missing in the table, it uses a complex procedure to broadcast a message to get information on the new route.

Results of researches show that when Media Access Delay, Throughput, Delay, Load and Network Load are considered, AODV returns a balanced value [8].

4) GRP (Gathering based Routing Protocol)

GRP is a hybrid protocol as it uses strengths of all above mentioned protocols. It selects the best possible route for packet transfer first and then routes the data packet.

Results of researches show that when Media Access Delay, Throughput, Delay, Load and Network Load are considered, GRP gives the best results [8].

CHAPTER 3: ROUTING PROTOCOLS FOR MANETs

3.1 Routing protocol

A routing protocol defines in what way or manner routers interconnect with one another. It defines path that allows them to choose routes linking any 2 nodes on a network. The algorithms determine the particular option of a path. Every router has preexisting information of networks connected to it. A routing protocol imparts this knowledge initially among adjoining associates (neighbors), and after that all over the network. That is how routers acquire information of the topography of the network. Even though there are several kinds of routing protocols, 3 categories operate prevalently on IP networks, namely:

- Interior gateway routing - link state routing protocols, e.g. IS-IS & OSPF
- Interior gateway routing - path vector / distance vector protocols, e.g. EIGRP & IGRP
- Exterior gateway routing - The Border Gateway Protocol (BGP) - routing protocol employed on the internet for swapping transportation connecting independent schemes.

Most of the accepted routing protocols have description in papers labeled RFC's. Certain editions of the Open System Interconnection (OSI) networking representation discriminate routing protocols in a distinctive level of the Layer 3 (Network Layer).

The particular characteristic of routing protocols is the behavior in which they skip routing loopholes, the way in which they opt for preferred routes, by means of knowledge regarding hop costs & additional features.

Several network curriculums discriminate amongst routing protocols & routed protocols. A routed protocol is employed to convey application transportation. It imparts suitable dispatch knowledge in its Internet Layer (Network Layer) to expedite a packet to be passed on from one network to another. Examples of such routed protocols are the Internet Protocol (IP) & Inter-network Packet Exchange (IPX).

3.2 Static routing

In this scheme, paths all the way through a data network are represented by means of permanent course (statically). These paths are generally inserted into the router by the scheme supervisor. A full network can be put together by means of static routes, although this sort of design is not error tolerant. As soon as there is a variation in the network or a malfunction happens amid two statically distinct nodes, transfer will not be re-routed. This indicate that something that desires to undergo a disturbed pathway will either have to linger for the malfunction to be refurbished or the static route to be restructured by the overseer before regenerating its trip. The majority request will expire (finally failing) before any of these maintenance can be completed. There are, nevertheless, instants when static routes can enrich the operation of a network.. The contradictory of stationary course-plotting is dynamic routing, occasionally attributed as adaptive routing.

3.3 Adaptive routing

Adaptive routing describes the capability of a system, through which routes are characterized by their destination, to alter the path that the route takes through the system in response to a change in conditions [9]. The alteration is anticipated to permit as various paths as possible to stay applicable which explicitly include target that can be attained) in reaction to the alteration.

Public utilizing a transportation scheme can exhibit adaptive routing. For instance, if a neighboring railway station is blocked, general population can dismount it at a dissimilar place & make use of a different technique, for example a bus, to make it to their goal. The expression is generally employed in data system to portray the competence of a system to sustain a malfunction, the general example that employ this technique are as follows:

- RIP
- OSPF
- IS-IS
- IGRP/EIGRP
-

3.4 Distance-vector routing protocol

In packet switched system, a distance-vector routing protocol is one of the two major module of routing protocols, the other known as the link-state protocol. Distance-vector routing protocols utilize the Bellman-Ford algorithm, Ford–Fulkerson algorithm to estimate routes.

A distance-vector routing protocol impose that a router update its neighbors of topography variation periodically. Distance-vector routing protocols has fewer data processing difficulty & communication operating cost.

Exemplar of distance-vector routing protocols incorporate IGRP, RIPv2 & RIPv2.

3.5 Link-state routing protocol

A link-state routing protocol is the other of the two major module of routing protocols utilized in packet switching networks for workstation interactions . Exemplar of link-state routing protocols comprise of intermediate system to intermediate system (IS-IS) & open shortest path first (OSPF).

The link-state protocol is executed by each switching node in the network (i.e. nodes that are equipped to forward packets; in the Inter-net, known as routers). The fundamental perception of link-state routing is that all nodes compose a chart of the association to the network, in the shape of a diagram, presenting which nodes are linked to which other nodes. Every node then separately determines the optimum rational route from itself to all likely targets in the network. The flock of top routes will therefore outline the node's course-plotting chart.

3.6 Link-State versus Distance Vector

Link-state algorithms (additionally recognized as shortest path first algorithms) deluge routing knowledge to every node in the inter-network. Every router fabricates a representation of the complete network in its routing charts. Distance vector algorithms (in addition acknowledged as Bellman-Ford algorithms) entitle every router to launch several section of its course-plotting chart, to its neighbors. On the other hand link-state algorithms throw small information all over

the place, whereas distance vector algorithms convey bigger information solely to adjacent routers. Distance vector algorithms are solely aware of their neighbors.

Since they flock more rapidly, link-state algorithms are not much inclined to course-plotting spirals than distance vector algorithms. Link-state algorithms consume more central processing unit energy & memory than distance vector algorithms. Link-state algorithms, for that reason, can be additionally costly to put into operation & sustain. Link-state protocols are normally more updateable on demand than distance vector protocols.

3.7 Optimized Link State Routing Protocol

The Optimized Link State Routing Protocol (OLSR) is a link-state routing protocol enhanced for mobile ad-hoc networks (which is also operable on different wireless ad-hoc networks). OLSR is pro-active, it utilizes Hello & Topology Control (TC) messages to ascertain & distribute link state knowledge into the mobile ad-hoc network, Via Hello messages all nodes detects 2-hop neighbor knowledge & establishes an array of multipoint relays (MPRs).

3.8 Characteristic explicit to OLSR

Link-state routing protocols such as IS-IS & OSPF opt for a selected router on each connection to achieve flooding of topography knowledge. In wireless ad-hoc networks, there is a distinctive conception of a connection, packets are able to go in and out using the identical interface; for this reason, a special method is considered necessary in turn to enhance the flooding procedure. By means of Hello messages the OLSR protocol at every node determines 2-hop neighbor knowledge & executes a dispersed selection of an array of multipoint relays (MPRs). Nodes choose MPRs that there exist a lane to every one of its 2-hop neighbors by means of a node chosen as an MPR. These MPR nodes then supply & promote TC messages that include the MPR endorsers. This operation of MPRs makes OLSR exclusive from other link state routing protocols

Given that link-state routing necessitates the topography record to be harmonized throughout the system, IS-IS & OSPF execute topography flooding by means of a dependable algorithm. Such an algorithm is exceptionally complex to devise for ad-hoc wireless networks, so OLSR does not

hassle through consistency; it plainly deluges topography data frequently as much as necessary to ensure that the record does not hang on disorganized for extensive phase of time.

3.9 Messages

OLSR employ of "Hello" communication to discover its single hop neighbors & its two hop neighbors owing to their answers. The dispatcher can after that decides its multipoint relays (MPR) based upon the one hop node that suggests the most excellent route to the two hop nodes. Every node has as well an MPR endorser array, which specifies nodes that have elected that node as an MPR node. OLSR utilize topology control (TC) communication next to MPR to propagate neighbor knowledge all the way through the system. Host & network association (HNA) communication are employed by OLSR to broadcast system route announcements like TC communication publicize host paths. The Diagram of Hello and TC messages are illustrated below:

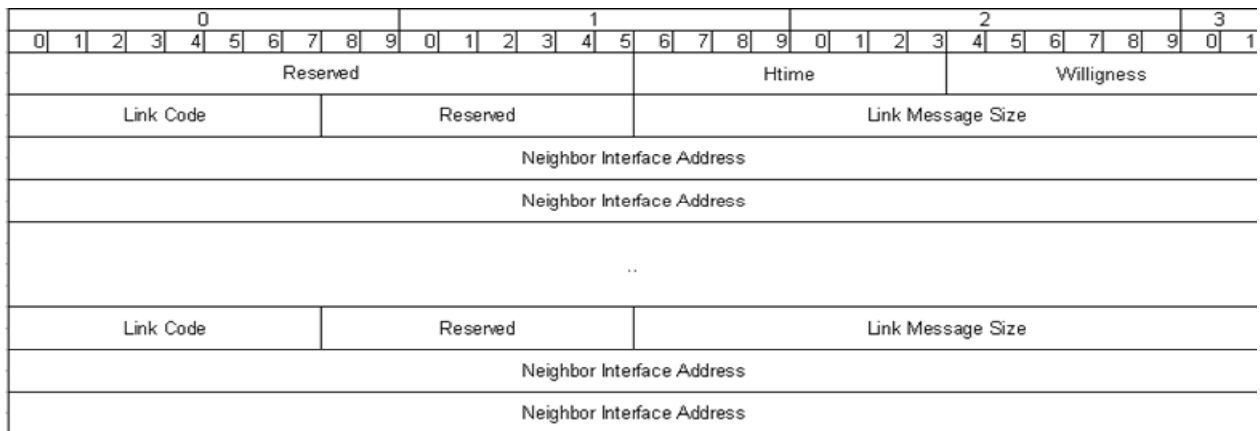


Figure 2: Hello

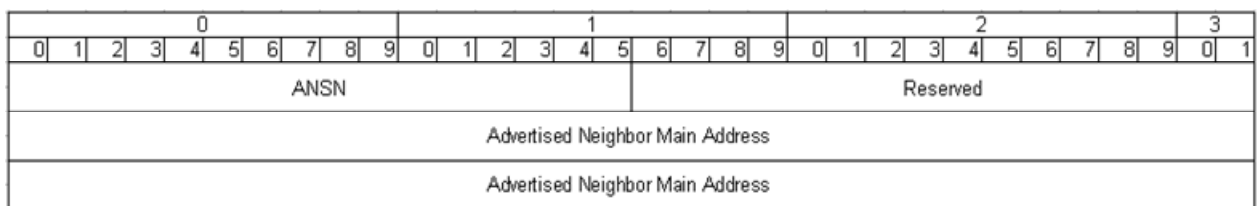


Figure 3: Topology control (TC)

3.10 Benefits

As being a pro-active protocol, paths to each and every target inside the system are recognized & sustained prior to utilization.

The cost of course-plotting operation, while normally larger than that of a reactive protocol, doesn't increase with the amount of paths being formed.

De-fault & system paths can be infused into the scheme by HNA communication allocating for link to the inter-net or additional system inside the OLSR MANET cloud. A system path is the thing that reactive protocols don't implement very remarkably.

Time-out assessment & legitimacy knowledge is also enclosed inside messages being shared inside the system.

CHAPTER 4: VOICE OVER INTERNET PROTOCOL

4.1 Introduction:

Voice over IP is the term used to name the phenomenon of transferring voice or multimedia over the internet. The difference between VoIP and PSTN (Public Switched Transmission Network) is in the transmission methodology. VoIP relies on generating packets of the digital information and using packet switched networks to transfer data over the internet protocol. PSTN relies on circuit switched networks. The remaining processes of sending a signal, setting up channel, digitizing analog voice signals, and using encoding are similar in both methodologies. There are three generations of VoIP structures:

- 1) The first generation targeted business models and technical solutions that mimic the same infrastructure of traditional telephone system.
- 2) The second generation targeted a closed network for a private group and charge only for services requested for traditional communications systems
- 3) The third generation covers the barrier the second generation systems imposed and allows connection of any two domains whenever a user wishes.

VOIP uses TCP/ IP protocol on its base. Protocols are used to define a set of rules for communication. TCP/ IP is the oldest one so far. TCP/ IP have five layers each of which is discussed below:

- 1) Application:

This layer is responsible for implementing soft switches that take care of the quality of the packets and delivers the packets at the target. Special protocols are put in place at this layer for the quality to be maintained and avoid packet loss and jitters.

2) Transport:

A special protocol called User Datagram Protocol is employed at this layer. This UDP protocol is responsible for transferring the VoIP packets from originating end to the target and vice versa.

3) Internetwork:

Network devices use Internet Protocol addresses for identification over a network. Each device on the network gets a unique internet protocol address. IP addresses are bonded into the packets on this layer and it remains intact till the life of the call between any two ends in a network.

4) Network Interface:

To determine where the packets are routed to, a MAC address is required. This MAC addresses is unique and is given by the NIC that is a requirement for all network devices. This layer assigns the MAC address to packets to find their way through the network to target machine.

5) Physical:

The digitized signals are transferred to electrical signals to travel on the network through this layer.

4.2 Packet Switching:

The packets of data that are transferred via packet switching follow any path and reach in any order to the destination over the network. This distinguishes circuit switching in a way that instead of keeping a constant and open connection, connections are built for a small interval to transfer the packet. The process of packet switching follows the steps mentioned below:

- The originating computer creates chunks of data. An address is attached with each chunk. This address is an indicator of destination node for the particular packet
- Each packet encapsulates a payload. This keeps the message that has to be transferred.

- The sending computer's task is to make sure the packet is sent to a nearby router. Once this is done. The sending computer does not track the packet. This creates data integrity risk in packet switching.
- The router searches for a nearby router that is closest to the packet destination as well. This way packet hops from router to router till they make it to their intended destination.
- Once the packet makes it to its target system, the target system decodes the message and assembles the message back to its original state.

Packet switching is an effective technique in a way that it makes the communicating computers free from the hassle of locating and transferring packets. This responsibility lays with the routers. Therefore routers use efficient algorithms to maintain a quick but cheap path to the destination node.

4.3 Advantages of using VoIP:

One of the greatest benefits VoIP provides is its capability to reduce the data transfer in the same amount of time as compared to a packet switching network. The data reduction is attributed to the fact that small packets are made that are transferred and no packet carries disruption or silence. So if a data compression technique is added to it as well, the cost of using VoIP turns out to be a lot cheaper than traditional telephony.

VoIP has an infrastructure that is best suited to network design and hence adds capability for the networks to be utilized at their maximum.

4.4 Disadvantages of using VoIP:

Reliability has been the major concern in VoIP. It requires a stable power due to the fact that it depends on wall power unlike current phone system that runs on phantom power. So if a power breakdown occurs, communications halt. Another issue is the fact that people are switching their home networks on VoIP such as TV channels etc. Unless a stable power supply is guaranteed, people will feel insecure adopting such services. Emergency phone calls become risky in a way that location cannot be traced by IP straight away. Latency, jitter and packet loss are inherent to

all VoIP systems. VoIP is also vulnerable for hacking and virus attacks. Quality of service is also a concern with VoIP as the system specifications strongly matter when VoIP systems are used.

4.5 Codecs & Soft Switches:

Coders refer to the coding and decoding phenomena of an audio signal to a digital one and then back to audio. Compressing and decompression of signals takes place in this technique and sampling of data is done. Sampling is done at such regular intervals that the end user cannot guess the drops in the signal. Sampling allows an audio signal to be digitized and then converting it back to an audio signal at the receiving end. There are multiple types of Codecs that are used. Each has its own sampling frequency and an algorithm. Algorithms are made robust enough to ignore silent periods in a conversation.

Once the sampling is done, the system needs to know where to send the data. The information of destination address is managed by a soft switch. The addresses in VoIP are IP addresses. IP addresses are dynamically assigned by a DHCP server so the switching taking place is not hard wired. A central call processor uses soft switch to map users and their addresses hence connecting end points in VoIP systems. Mapping tables are maintained within the soft switch so it knows where the end points are, the phone number they are associated with and the internet protocol they carry.

While finding an internet protocol address if the soft switch encounters an issue of unavailability of an address, it sends the request to downstream systems. The procedure carries on till a perfect hit is found. As soon as the address is located, the message is sent to the end user. VoIP technology relies on a lot of network devices to work together and understand the needs of each other. In order to facilitate smooth communication, several protocols are suggested and are in use. The major protocols designed are for video, audio, transport and data. There is a huge technical gap of compatibility between all VoIP protocols. Some offer excessive features while others rely on traditional settings. Refinement of this technology is a hot topic these days to replace the traditional phone system.

4.6 VoIP Call Monitoring:

Since quality is a serious drawback of VoIP, VoIP call monitoring is used to fix the quality to an acceptable level. The quality of service is a measure through which call quality is assessed through a series of tests on both hardware and software levels. The tests are based on mathematical algorithms so the assessment is quantitative. A measure called mean opinion score is used that has a scale from 1 to 5 where 1 is the lowest and 5 is the highest quality. The parameters used for mean opinion score are as follows:

1) Latency:

The delay between two end points in a VoIP system is called latency. There are two ways to measure it. It is either taken to be a one way trip or a round trip. Poor latency results in people ending up talking without waiting for the other to finish first as they feel that the other person is already silent. This is termed as talk over effect.

2) Jitter:

When packets arrive late at the end point or are not in the order they should be, jitter is caused. So jitter refers to the latency of packets in making to the end point and hence is a random variable. In order to reduce jitters, a technique called jitter buffer is devised in such a way that it collects packets at the end point in form of groups. It then arranges the packets in the correct order. Once the order is maintained, the packet is delivered in one go to the end point.

3) Packet Loss:

The drawback of jitter shows up when it gets overloaded and it starts delaying the chunk of packets or late arriving packets are not picked up in the chunk. Packets may also get lost in the network due to other reasons that result in entire sentences to be skipped. This is called bursty loss. Packet loss is taken as a ratio of percentage of packets lost to the percentage of packets received.

Call monitoring is done in Active mode before a deployment of VoIP and passive testing is done once deployment of VoIP is done.

4.7 Latest Trends in VoIP:

Consumers have a demand to keep on VoIP with mobility. This mobility is ensured by the use of VoIP on cellular network. So a demand in cell phones having VoIP has increased. A recent research shows that by 2013 there will be 288 million VoIP users across the globe [10].

Competition has grown abundantly of VoIP service providers. Rebtel alone has generated a revenue of 0 million US dollars with 20 million active users across the globe [10]. Petitions have been filed with the federal communications commission to make a plan to retire traditional telephone systems. The only harbinger of bad news in this regard is the gap due to the drawbacks of VoIP systems. However, efforts are being placed for developing a reliable system that protects privacy as well in VoIP. Another issue is fraud detection and protection in the system. Even the best of VoIP current services are not spared from being hacked. Current reports suggest that phone fraud is going on at the rate of 29% per year [10]. For availability and better performance, VoIP has to be moved in the cloud and a lot of research is underway for implementing an effective cloud strategy for VoIP systems. Currently used servers are becoming obsolete and will soon vanish from practices of current enterprise applications and processes.

Implementing a VoIP and then adding security measures to protect it becomes an expensive approach to implement VoIP. In order to make VoIP less expensive overtime, a comprehensive plan is put into place to add a VoIP system. This plan has an architectural framework which digs out the points where VoIP has risks and how to address them. The five key points that can help in this regard are as follows [11]:

1- Security responsibility assignment:

A centralized approach for security responsibility should be put in place to avoid cracks in security.

2- Review of Network wide security:

There is a need to make sure that when network security measures are taken, they not only consider data, but also consider VoIP security. There should be a mapping activity that defines what security measure is taken for what type of data. Probably extra layer of security

will be required for VoIP depending on the measures already taken by the organization or its existing security policies.

3- Development of security policies:

When security policies are defined, it should be kept in mind that they should not be a hassle for the employees and should be actionable. The policies should also be understandable and their goal should be clear to motivate employees to use the policies as system of operation.

4- Vendor engagement

It is of great necessity to engage with vendors to see overtime f new vulnerabilities are created in the system, what is latest n the market and how to protect the organization from specific security issues. Such engagements promote knowledge transfer at both ends. The vendors come to see new requirements and make their system flexible overtime. As far as the organization is concerned, it will save itself from latest security attacks by adopting latest security measures.

5- Training and Education of employees:

Employees use software that help them connect with the outside world very often. They need to be trained to adopt what policies while doing such an activity to save the organization from damage. If employees are not made a part of the solution for VoIP security measures, due to unawareness they will turn out to be part of the problem in the long run unconsciously.

CSipSimple and Jitsi softwares are used for secure VOIP using TLS/ SRTP mechanism. A unified Communications summit is also in place these days that gathers data of latest trends in communications and how to help secure communication process and define global patterns or policies for security. VoIP has a long way to go and is a hot subject in research these days.

CHAPTER 5: SECURITY ISSUES IN MANETS

5.1 MANET Security Services/ Vulnerability:

The security services in MANET are usually limited to non-repudiation, confidentiality, Integrity, availability and authentication. Knowing the correct identity of the communicating partner is authentication. Confidentiality describes the level to which data can be disclosed and to which communicating node. The sensitivity of data decays over time in a MANET e.g. location of a secret service today is more important than the past days. In order to hold integrity, a MANET is supposed to transfer data as a whole by not allowing corruption of it. Availability is the measure of connection established against time when the network is attacked. Non repudiation is a check to track the originator of the message being shared in the network. This way the originator cannot deny sending it on the network. This also keeps track of commitment of actions to be performed by nodes. The concept of security in a MANET is attributed to the authentication feature. Communication makes sense if you know with whom you are communicating and you can trust the source. Then you can decide if the data needs protection or not. Because of the characteristics of MANET that are discussed in the previous chapter, a MANET is more susceptible to security breaches than a wired network. During the past couple of years, end users have shifted to mobile banking etc, security aspects in android devices and networks has become a huge concern.

5.2 Attacks on MANET:

Every layer in a network protocol has its own sets of attack that are introduced due to the routing protocol strategy. Application layer has data corruption and virus or worms attack issue. Transport layer has TCP/ UDP or SYN Flood types of issues. Black hole is typical to network layer and traffic analysis monitoring is an attack on the data link layer. Eavesdropping and direct attacks on physical layer are the main issues that kill privacy in MANETs. There are two major types of attacks in MANETS: Active and Passive [13].

5.3 Passive Attacks

Any unauthorized node can try to find out information of a network by monitoring it. The attacker can silently work on the network and does not have to disrupt communications on it. They also do not cause any direct damage to the network itself. Eavesdropping and traffic analysis are examples of passive attacks. Data gathered this way can be used in future attacks to cause harm. Eavesdropping can be done by external nodes of a network or internal nodes as well. To handle such an issue, the following solutions are proposed in literature:

- Frequency Hopping
- Spread Spectrum Communication

Traffic analysis can detect the roles of each nodes in a network, if two packets are linked or not, the network topology communications, current location of specific individuals or functions that are being performed on nodes, the location and existence of nodes etc.

5.4 Active Attacks

State changes in a network that are not authorized are forms of active attacks. These include modification of data packets and denial of service etc. These are usually initiated by nodes on a network so they are internal attacks. Nodes may try to drop data packets deliberately that are not meant for them or try to change data within a packet to preserve their resources. MANET works in a collaborative fashion and cannot tolerate a selfish node. It also affects the network performance and communication between nodes is halted. Nodes might also perform selective dropping to avoid being detected. Many routing protocols do not have the ability to detect if packets are delivered securely. DSR protocol does verify it though. Other strategies that are used to detect such a behavior include:

- Hop by hop acknowledgement
- Passive acknowledgement

These are used at the data link layer to verify packet transmission. Data packets can also be attracted to a single node disrupting the flow of network traffic. If they are attracted towards a compromised node in the network, they are termed as sinkholes. If they are attracted outside the

network by some other node, they are called blackholes. This can also assist the attacker to modify packets once they are attracted. Network nodes may also try to send fake data packets to a node that does not exist in the network. This may be done for the purpose of draining nodes of their resources and causing sleep deprivation of making them update their routing tables erroneously.

5.5 Counter Measures:

The counter measures range from prevention techniques such as creating a secure routing algorithm to intrusion detection. However they are mostly conventional procedures and do not apply readily on MANETS as due to the basic nature of MANETS, the conventional procedures may not detect possible attacks. Research is still underway for this area [13].

CHAPTER 6: ANDROID

6.1 Introduction:

Android uses Linux as its operating system. This operating system is developed to cater the need of touch screen devices e.g. smart phones and tablets. Android was initially introduced as Android Inc. Google stepped forward in 2005 and bought the company. A consortium for the support of open standards for mobile devices called Open Handset Alliance was held in 2007 which revealed the android operating system. The technology is not old and has been in market since October 2008. It has been news of great speculation when Google stepped into the android market. Experts inferred that Google is trying to be very persuasive in providing its search optimization and navigation features on mobile handsets. The consortium helped in revealing the intentions and gathered telecommunication industries; hardware and software industries to gather at one place and create something in conjunction to utilize the operating system to its maximum. The interesting part of Android is that it is open source and its code is released by Google under the Apache license. This fact has made it possible to create code that suits the user needs and has opened doors to an endless possibility of mobile phone applications. Records indicate that more than 700,000 applications of mobile phone applications are available and people eagerly download and use them. The usage of mobile applications is not limited to end users alone, but the developers re use the code to create enhancements or customize applications to suit their needs.

6.2 Android Look and Feel:

The android interface has been designed in a way to provide maximum output to users with minimum effort. The interface is made sophisticated as it uses some sensitive hardware that facilitates proximity. Android interface is developed to allow users to customize the endless possibilities of icons that they wish to see on their interface. Being an open source has allowed users to select from a wide range of selections the themes that they wish to use. Users get prompt indications of updates, their emails and other connectivity.

6.3 Android Architecture:

Android has Linux at its core and uses Java and C programming languages majorly for development. The security aspect of the architecture has always been a concern. Linux has its name as it does not follow a single convention for developing its modules. So virus attack and malware are hard to impact Linux. The free available code of android makes it more susceptible to damage because attackers are well known of the fact that people use android frequently for their personal use. Following is an architectural diagram of android operating system:



Figure 4: Android Architecture [12]

Since android depends on limited battery power, memory management is a serious issue in android phones. A number of techniques have been devised to make android phones more responsive. These involve suspending tasks that do not utilize memory on demand only. Background processing task for memory management is also handled. User does not have to open applications from scratch but they open it by revoking a suspended task from background. User is relieved of the headache to kill tasks to conserve energy and memory. This helps to save battery power. Processing time is also reduced. Since memory management is a serious issue in android phones because efficient response time is required, a lot of third party applications have

been developed for managing memory for android phones. It is speculated that such applications take more memory consumption rather than relieving the memory activities.

6.4 Android Community:

Updates are provided on android software by Google in the form of updated versions. These versions are released over a period of time and almost all devices can download them and install them the moment they are made available. Android updates are relatively slow to reach android users if the brand they are using is not NEXUS and the reason behind this is the fact that Google source code is tailored to support NEXUS officially. It often takes months to test the same source code for a device with other hardware designs and make it available for end users. One other reason is that source codes are usually made available to run on latest devices. It requires extra resources and testing effort for backward compatibility in respect of devices. So manufacturers may assess if they have the required resources available for updating old devices and then provide updated codes for them. The manufacturer may decide otherwise as well even if the device is capable of supporting the new code but resources may constrain the manufacturer or they deem the update to be worthless for investing time and energy in it. This attitude results in users being forced to purchase newer devices and get access to latest versions of software when users could be given the same thing on the devices they are already using. This lack of after sale support is widely criticized by people and causes mistrust as they already know at the backend of their mind that they will have to buy the recent version soon.

Google and android communities are planning to form an alliance to address this issue and make releases available soon for former as well as latest devices available in markets. A lot has to be done yet to make the initiative working and devise processes so they are respected and followed. Unfortunately it has been 2 years since it was mentioned in 2011 and so far no one has taken a step forward to even bring it under discussion again.

Android developers are rapidly producing versions of android operating systems. They often deliver updated versions of software even quicker than the manufacturers. The compromise they make is on the quality of the software. They do not do proper testing and do not follow proper quality assurance processes to provide a stable and bug free release. They also provide continuous support for software that were released for old devices but updates for them are no

longer provided by their manufacturers. If we peek into the past, it becomes evident that manufacturers have not been very supportive in appreciating third party development. By making jail-breaking of mobile devices permissible across the world, it has led to vigorous advancement in the android development technology. Developers take root permission and access the boot loader to get a better control of the hardware itself. This results in a tension between manufacturers and the android community. The android community keeps the stance that the industry has failed to provide updates in time where as the manufacturers are concerned over the proprietorship of the software that are being produced as a result of privileged access.

Android applications are not granted access to system resources unless a user allows it to happen. Prior to application installation, user is notified which system resources the application might be using. They range from access to memory cards to components in the android. Access to personal data is also notified in advance. So security of android devices is made a user's decision. They can discontinue giving the third party applications access that they do not wish to share. Similarly they can also discontinue access of applications to mobile resources any time by uninstalling the applications. There are some giant companies that have also made anti-virus software for android devices. Due to limited documentation and awareness of android architecture, these antivirus software are not as effective as they were expected to be. Firstly they take unnecessary permissions to run and secondly they are unable to run a full system scan at deeper levels. Google maintains the fact that security companies are aggravating the security negative news for business purposes and the actual facts are different. The true facts remain that android is a Linux based system so obviously security issues are not very common but also malwares are being created to call or send messages from one number to another without the consent of users and in the worse case, the users do not even know about the activity that has taken place. Security companies report an exponential growth of security issues in android devices where as Google reports only 0.5% of issues that are application security related. Google has employed its own malware detector which scans applications and notifies users of the security level of each application. Open source applications are also created and are in use that inform users when personal data is being sent to servers from the system.

CHAPTER 7: IMPLEMENTATION

7.1 Implementation of System:

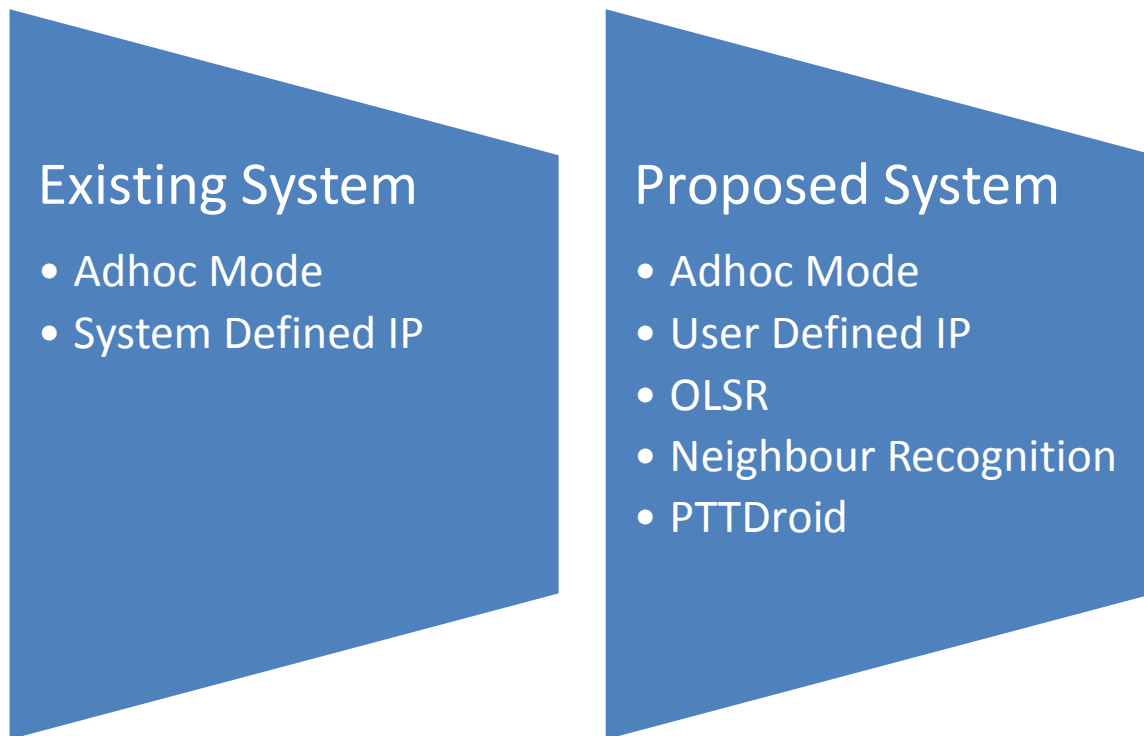


Figure 5: Implementation from Existing System

7.1.1 User Defined IP:

The existing system is modified to accept user defined IP. On the user interface, a section is provided for this purpose. A text box is added that takes an IP added by the user as input.

Following is the change that is done for it:

```
String ssid1 = this.settings.getString("ssidpref1", "AndroidTether");
```

The above line sets the SSID according to the SSID preferences given to the program.

```
String lannetwork = this.settings.getString("ssidpref1", DEFAULT_LANNETWORK);
```

Initially default LAN network settings are passed to the program unless user specifies their own.

```
String lanconfig = lannetwork;
```

Finally the network address is passed to the lanconfig parameter which is passed through a parser. It tokenizes the IP address and recognizes the address to assign to itself.

7.1.2 OLSR:

When the start button is pressed, we need a file to be picked from our memory stick and be placed in the internal memory of the android so it is accessible for the application to execute flawlessly. This file comprises of Olsr configuration. To access this file, we need to first mount the memory card in read/ write mode through our program and then write move command to move it from the memory stick to the internal memory. These sets of actions are performed when the Start button is clicked for the Wifi Tether application. A new process thread is created which commences execution of Olsr protocol at the backend. Following is the code snippet for this purpose:

```
public void run(){  
  
    String[] command = {  
  
        "busybox mount -o remount,rw /system",  
  
        "busybox mv /mnt/sdcard/olsr.conf /system/bin/olsr.conf",  
  
        "/system/bin/olsrd" } };
```

These commands open a terminal at the backend and execute in a sequential manner. The command /system/bin/olsrd executes the olsr protocol with its setup configurations. Busy box commands are used in place of normal linux/ android commands because busy box is known to be efficient in embedded systems that have limited resources.

7.1.3 Neighbour recognition:

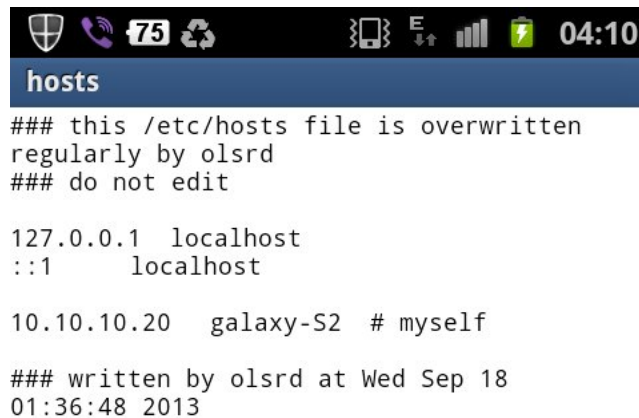
It is essential to know about neighbors in order to communicate with them. This feature helps in recognizing the name and IP of neighbors. This can be achieved by writing the following command on the terminal emulator:

```
$ su
```

```
# cat /etc/hosts
```

It displays the IP and device name/user name associated with it

The screen shot is as follows:



```
hosts
### this /etc/hosts file is overwritten
regularly by olsrd
### do not edit

127.0.0.1 localhost
::1 localhost

10.10.10.20 galaxy-S2 # myself

### written by olsrd at Wed Sep 18
01:36:48 2013
```

Figure 6: Neighbor Recognition

7.2 Configuration of Systems:

The configuration steps are different depending on whether the device uses a Linux based system or an embedded Linux system. Each step of the configuration needs to be executed in a methodical way to achieve a reliable setup. Configurations are done to build a connection between the devices to form a MANET and hence be able to use VoIP communication. If both devices are same, they follow the same setup procedure. Each of the process is explained below with its step wise procedural diagram.

7.2.1.1 Setup on PC for Real Time Monitoring:

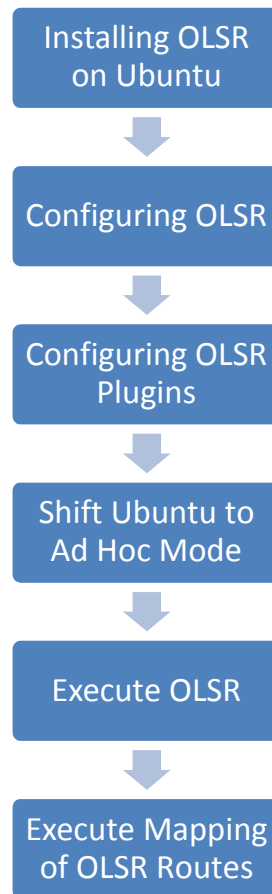


Figure 7: Implementation Setup 1 – On PC

7.2.1.1 Installing OLSR on Ubuntu

To install OLSR, a set of pre requisites are required. These comprise of:

1. flex
2. bison
3. libc6
4. git-core

The following commands are written in the terminal to install the plugins:

```
$ sudo su

# apt-get install flex bison libc6 git-core

# git clone git://olsr.org/olsrd.git

# cd olsrd

# make

# make install

# make clean

# make uberclean

# make libs

# make install_libs
```

The above commands install Olsrd on Ubuntu along with the plugins.

7.2.1.2 Configuring OLSR

To configure the Olsr.conf the default file is discarded right away and the new one is created having following parameters:

DebugLevel 2

IpVersion 4

LockFile "/tmp/olsrd.lock"

Hna4

{

}

Hna6

{

}

MprCoverage 1

LinkQualityAlgorithm "etx_ffeth"

LinkQualityFishEye 1

IpcConnect

{

}

InterfaceDefaults

{

}

Interface "wlan0"

{

HelloInterval 2.00

HelloValidityTime 20.00

TcInterval 5.00

TcValidityTime 30.00

MidInterval 5.00

MidValidityTime 50.00

HnaInterval 5.00

HnaValidityTime 15.00

}

7.2.1.3 Configuring OLSR Plugins

There are 5 plugins that needs to be inserted, their details are discussed below:

- i. TXT INFO PLUGIN - This format of code that is shown below is inserted in the config file. This Plugin helps in watching the Olsr status inside the CLI and it keeps updating itself up via a command

```
#watch -d -n1 "echo '/all' | nc localhost 2006"
```

passed inside the terminal

```
LoadPlugin "olsrd_txtinfo.so.0.1"
```

```
{ }
```

- ii. HTTP INFO PLUGIN - This format of code that is shown below is inserted in the config file. This plugins normally helps in viewing the status of the Olsr daemon inside the default web explorer making it easy to understand.

```
LoadPlugin "olsrd_httpinfo.so.0.1"
```

```
{ }
```


- iii. DOT DRAW PLUGIN - This format of code that is shown below is inserted in the config file. This plugin helps in making an image of the routes being made or a visual display of the topology of OLSR currently not supported by Android only to be deployed over a Ubuntu Machine.

```
LoadPlugin "olsrd_dot_draw.so.0.3"
```

```
{ }
```

- iv. NAME SERVICE PLUGIN - This format of code that is shown below is inserted in the config file. This plugins helps in the neighbor recognition where the name is defined before the olsr is set to run - the IP and names associated are then listed in hosts file.

```
LoadPlugin "olsrd_nameservice.so.0.3"
```

```
{
```

```
PIParam "name" "Laptop"
```

```
PIParam "hosts-file" "/etc/hosts"
```

```
PIParam "resolv-file" "/etc/resolv.conf"
```

```
}
```

- v. OLSR SECURE PLUGIN - This format of code that is shown below is inserted in the config file. This plugins helps in securing the network the detail of how it works is explained in this thesis at section 7.3.4.1.

```
LoadPlugin "olsrd_secure.so.0.6"
```

```
{
```

```
PIParam "Keyfile" "/path/to/keyfile.enc"
```

```
}
```

7.2.1.4 Shifting Ubuntu to Ad-hoc Mode

Shifting Ubuntu 12.04 into ad-hoc mode is extremely easy using the local default network manager. Right click on the network manager and click on the option Edit Connections. Then go

to the wireless Tab click on Add button In the SSID a particular name is entered by which we want our network to be recognized with. Change the mode to ad-hoc. Select the Channel you want and click on the save button. Connect any other device to it and the pc will be in ad-hoc mode. The interface is the one used for wireless network which can be cross checked via the command in terminal i.e.

```
# iwconfig
```

7.2.1.5 Execute OLSR

In the terminal, pass the command `# olsrd` and the daemon will be up and running.

7.2.1.6 Execute Mapping of OLSR Routes

This is the trickiest part as there are multiple packages that are required to make this plugin work. These commands are passed in the terminal in order to make the graphical view of the topology. They are listed as follows:

```
$ sudo su

# telnet localhost 2004 ( to check if the plugin is working )

# apt-get install imagemagick graphviz wmcctl

# chmod 755 /olsrd/lib/dot_draw/olsr-topology-view.pl

# while true

# pkill /olsrd/lib/dot_draw/olsr-topology-view.pl

# wmcctl -c "topology.png"

# sleep 10

# do /olsrd/lib/dot_draw/olsr-topology-view.pl

# done
```

This will run a small program that will display the Olsr topology and it updates in real time. Below are the examples of the topology view that is experienced by the user.

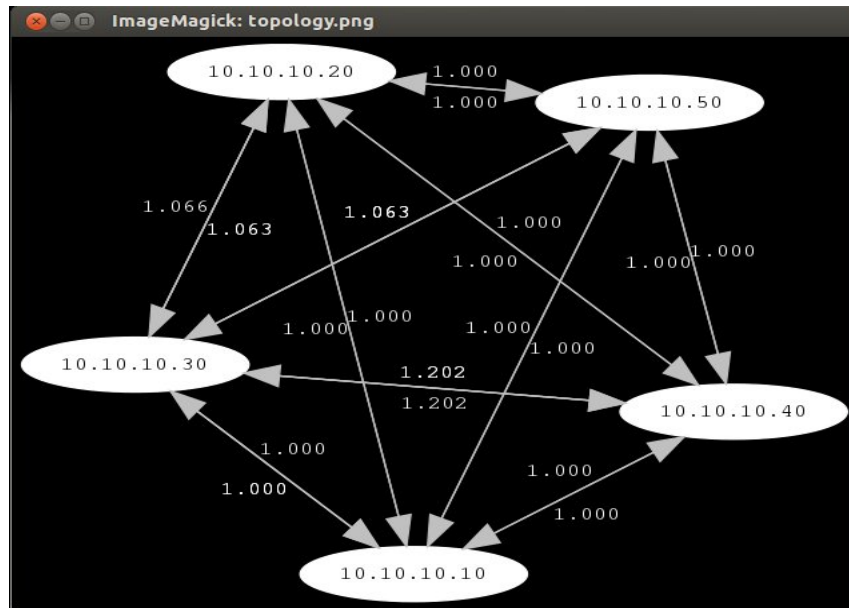


Figure 8: Example 1 Topology View

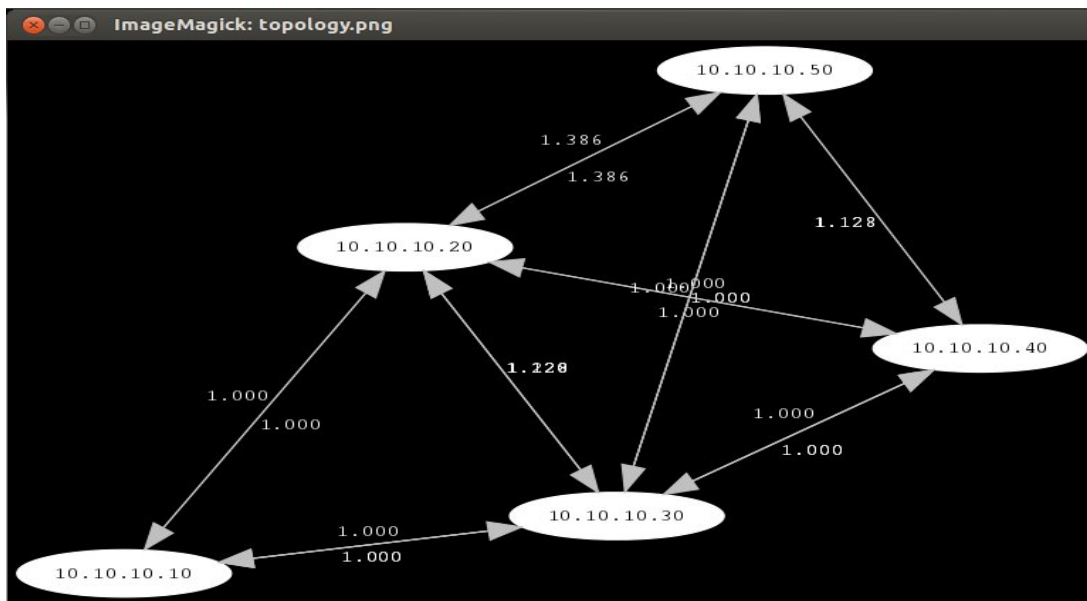


Figure 9: Example 2 Topology View

7.2.2.1 Setup on Android:

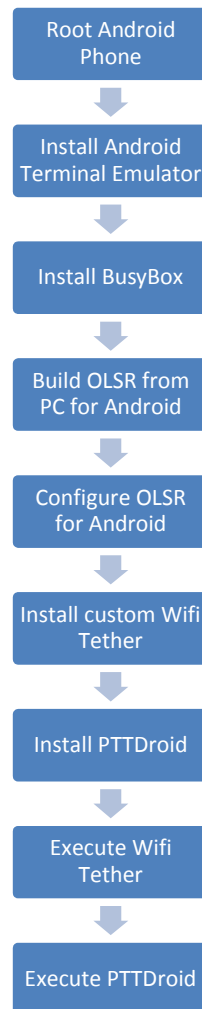


Figure 10: Implementation Setup 2 – On Android

7.2.2.2 Rooting android phone:

We used three different devices using three different versions of android to verify that the proposed system works on every level of android and every type of android device. The devices used are as follows:

1. Samsung Galaxy Y GT-S5360
2. Samsung Galaxy S II GT-I9100
3. Ainol Tablet Aurora 2

As every device is different so rooting of each of these devices has a different method. For every device the rooting method is listed as follows:

7.2.2.3 Rooting Samsung Galaxy Y GT-S5360

Samsung Galaxy Y is an android phone. It has 800 MHz CPU and possess 290 MB RAM. It runs on android 2.3.6 Gingerbread.

1. Download the Update.zip file from this link -

https://docs.google.com/uc?export=download&id=0B_DvhUU21L2-MzINVThja1Q1TW8

2. Now using PC move this file to your Phone's SD card.

3. Now turn off your device.

4. Boot into recovery Mode- [Press Home+Volume Up+Power Button]

5. From the Menu Select "Update from SD card"

6. Next choose the copied Update.zip file from SD card.

7. Reboot using the reboot option in Recovery options menu.

Rooting Samsung Galaxy S II GT-I9100

7.2.2.4 Rooting Samsung Galaxy S II GT-I9100

It is an android phone. It has a Dual-core 1.2GHz ARM Cortex-A9 CPU and possesses 1024 MB RAM. It runs on android 2.3.6 GB.

1. Download the "CWM-SuperSU-v0.97.zip" file from this link -

<http://forum.xda-developers.com/attachment.php?attachmentid=1674014&d=1359080668>

2. Now using PC move this file to your Phone's SD card.

3. Now turn off your device.

4. Boot into recovery Mode- [Press Home+Volume Up+Power Button]
5. From the Menu Select "Update from SD card"
6. Next choose the copied "CWM-SuperSU-v0.97.zip" file from SD card.
7. Reboot using the reboot option in Recovery options menu.

7.2.2.5 Rooting Ainol Tablet Aurora 2

Ainol Tablet Aurora 2 is an android Tablet. It has a Dual-core 1.5GHz ARM Cortex-A9 CPU and possesses 1024 MB RAM. It runs on android 4.0.3 IceCreamSandwich.

1. Download the "Root Aurora II.zip" file from this link -

<http://www.mediafire.com/download/gc76rpbt8g2yb75/Root+Aurora+II.zip>

2. Now using PC move this file to your Tab's SD card.
3. Now turn off your device.
4. Boot into recovery Mode - [Press and Hold Power + Volume - buttons]
5. From the Menu Select "Update from SD card"
6. Next choose the copied "Root Aurora II.zip" file from SD card.
7. Reboot using the reboot option in Recovery options menu.

7.2.2.6 Install Android Terminal Emulator

This is a free application available on Google Play Store which can be easy installed on any android device by downloading it on the device or getting an apk and installing from the sdcard.

This application helps in giving a CLI on android like Linux.

7.2.2.7 Install BusyBox

This is a free application available on Google Play Store which can be easily installed on any android device by downloading it on the device or getting an apk and installing from the sdcard. The installation is done over /system/xbin. It helps in the execution of the basic commands of Linux on android device like ls cat etc.

7.2.2.8 Build OLSR from PC for Android

There are a set of plugins that are required for OLSRD in order to run on an android phone. The list is as follows:

1. bison
2. flex
3. libc6
4. Android SDK
5. Android NDK
6. git-core

An SDK and NDK are also required and are extracted in /opt folder to avoid system issues. NDK base file location is updated in makefile to reflect the actual location of extracted development toolkits. The following commands are executed to generate relevant files that are required in android:

```
#make OS=android DEBUG=0 NDK_BASE=/opt/android-ndk
#sudo make OS=android DEBUG=0 install_all
```

The following files are generated:

```
/bin/sgw_policy_routing_setup.sh
/bin/olsrd
/etc/olsrd.conf
/lib/../../
/usr/share/doc/olsrd
/usr/share/man/man5
/usr/share/man/man8
```

These are copied on the android phone, to their respective folder in the android root. A copy of olsrd.conf is also kept /mnt/sdcard in case we need to modify our Olsr configuration again. The wifi tether application is configured to copy the latest olsrd.conf to the system /etc folder every time it starts, making it easy to change configuration at any stage even if root exploration is not possible.

7.2.2.9 Configure OLSR for Android

The files generated in step 2 are copied on the android phone. Interfaces are then verified which can be done easily by the Android Terminal Emulator by executing the following command inside the prompt which is as follows,

```
$ su
```

```
# netcfg
```

This command lists down interface that is responsible for wifi connectivity

It is used on three devices separately. Their names and the interface found are listed below. They can be modified in the olsrd.conf to suit the requirement.

1. Samsung Galaxy S2 - wlan0 (running JellyBean 4.1.2)
2. Samsung Galaxy Y - eth0 (running Gingerbread 2.3.6)
3. Ainol Tab Aurora II - wlan0 (running IceCreamSandwich 4.0.3)

7.2.2.10 Install Custom Wifi Tether

The custom Wifi Tether that is prepared earlier for the operation is installed by first moving the apk file of the application onto the sdcard of the device and then installed simply by tapping over it. The system recognizes the format of the file and installs it over the device.

7.2.2.11 Installing PTTDroid

The PTTDroid that is kept prepared earlier for the operation is installed by first moving the apk file of the application onto the sdcard of the device and then installed simply by tapping over it. The system recognizes the format of the file and installs it over the device.

7.2.2.12 Execute Wifi Tether

After installation, the icon for the Wifi Tether is made available on the Application Drawer Panel and is executed by first tapping over the app icon. When the application is open there is a large icon of wifi which needs to be tapped and our application is then executed and the device moves into Adhoc mode and Olsr starts running in the background. The application can be exited as it will continue to run in background and a symbol appears in the upper left panel of the android screen confirming that the application is running.

7.2.2.13 Execute PTTDroid

After successful implementation of Ad-hoc network and olsr we again move to the application drawer and look for the PTTDroid application icon and execute by tapping over it. To establish a voice connection the IP is supplied in the option of the application and communication is commenced by tapping over the microphone icon and holding it for the time the user needs his voice to be sent over to the target. As soon as the user lifts his finger from the microphone icon on the screen the application switches out of the sending voice data mode. Though the device remains to be in the listening mode i.e. receive data mode at all times.

7.3 Securing the Systems:

Security of the working system was the last part of this entire thesis, we used multiple options for this and explored them to secure the system. Each of which is listed as follows:

7.3.1 CSipSimple

This was the first attempt at making the system secure but the issue was that the software had a feature in which it used to call on the network manager of the device to check the wireless interface state. As the system was functional over ad-hoc mode and the network manager used to indicate to the software that wifi default state is off, the software failed to execute indicating the wifi state . Hence this option could not be considered. The security feature of securing the call using TLS/SRTP/ZRTP would have ben a good option for making the system secure.

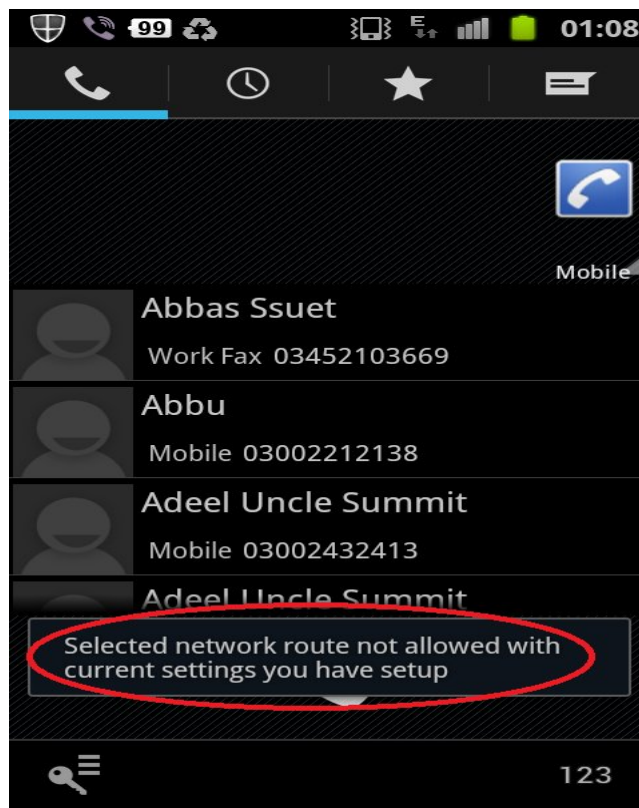


Figure 11: CSipSimple Issue

The red circle indicates the error for which there was no work around and the application failed to start.

7.3.2 Stream Cipher

The next method that was adopted to secure the system was to do stream ciphering. In android Google has already developed a class in its API by the name of JAVA.CIPHER which can be

called for ciphering while building the application in the eclipse framework. The code snippet is as follows:

OnPlayer Side:

```
while(keep_on_running) {
    socket.receive(packet);
    encodedFrame = AESEncrytion.decrypt(encodedFrame);
    Speex.decode(encodedFrame, encodedFrame.length, pcmFrame);
}
}
```

On Recorder Side:

```
while(keep_on_running) {
    Speex.encode(pcmFrame, encodedFrame);
    AESEncrytion.encrypt(encodedFrame);
    socket.send(packet)
}
}
```

Figure 12: AES Encryption Using Speex Codec

Although Encryption was achieved but due to heavy load and processing power this operation required, PTTDroid Application halted leaving it render-less and unstable and hence became non usable for the system.

7.3.3 IPsec Tunneling

This mode of communication in android is still in alpha stage. The application named Advanced IPsec Tools is available for free download on the Google Play. It requires root access to the cell phone. It claims to support two modes of operation namely:

- i. PSK mode
- ii. CERT mode

In the PSK mode it claims to establish connectivity with the help of 8 characters ASCII password that is user defined. In the cert mode it requires a private key file and a certificate generated pem file from that key. These two should be supplied when the application is put in the certificate mode to establish an IPsec tunnel between two wirelessly connected cell phones.

After cross-checking when the application it establishes a connection with the other nodes. The packets were analyzed with the help of the wirehshark in promiscuous mode. It was discovered that the channel was indeed being established with the help of ISAKMP (Internet Security

Association and Key Management Protocol). But when it was later known that the tunnels were not active and data was not transferred as the nodes were still stuck in negotiating the parameters for connectivity.

The two conf files namely set-key.conf and racoon.conf were reconfigured numerous times but to no avail. And hence this idea was dropped as well.

7.3.4 Securing the Network

Although the methods listed above were all good approaches towards securing the system however every method had its deficiency therefore this final and fourth method indeed worked out.

This was done by first creating a simple key file with the following command:

```
# echo "password" > keyfile
```

This generates a simple file containing the word "password". To add one more layer of security to this we encrypted the file by using the following command:

```
# openssl enc -aes-128-cbc -salt -in keyfile -out keyfile.enc
```

This encrypts the file using aes encryption algorithm with 128 bit block in CBC mode. This file is then placed in the folder anywhere desired.

Using the olsr secure plugin in the olsr.conf, the file is directed as the key to be used for securing the network.

7.3.4.1 Functioning and explanation

The key that is selected an md5sum is generated from it which is used then to digitally sign each and every packet of the Olsr network. A same key file is placed in other nodes as well which generates an md5sum of it. When the other node receives this digitally signed file by the md5sum hash it is first separated from the packet and matched with the current md5sum hash residing on the node locally. If it matches i.e. verified the packets are accepted and Olsr sending node joins the network. If the signature cannot be verified, the message is discarded and the node sending the packet is devoid from joining the system. The same method is used as well for the intrusion detection, if someone is trying to join the network without a key, as it doesn't contain the digital signature and therefore is rejected by the Olsr Daemon. The sender IP is also displayed upon such action in the running Olsr Daemon. This can be viewed in the terminal emulator of the Android GUI.

CHAPTER 8: RESULTS

These values are calculated for the node to node scenario using line of sight approach i.e. no barriers. A case scenario illustration is as follows giving the idea of the way these tests were conducted in real world environment.



Figure 13: Test Case Scenario 1

The results are calculated on the following parameters and the tools used for its measurement are also explained as well. They are as follows:

i. Ping

This command was used to measure the round trip time taken by the packet from source to destination and way back. Total time taken was measured in ms (milli-seconds) this command was inserted into Android system with the help of BusyBox. The command was used as follows:

```
# ping -c60 10.10.10.40
```

ii. Throughput and Bandwidth

This was measured with the help of IPERF a simple application installed in android which calculates the amount of data transferred and the optimum bandwidth achievable. The command used was as follows:

```
# iperf -s -i 1 ( on server or side )
```

```
# iperf -c 10.10.10.40 -i 1 -t 60 ( on client side )
```

iii. Jitter and Packet Loss

This was also measured using IPERF for Android. The following commands were used:

```
# iperf -s -u -i 1 ( on server or side )
```

```
# iperf -c 10.10.10.40 -u -i 1 -t 60 ( on client side )
```

iv. Signal Strength

This was measured with the help of a simple app named Wifi Analyzer. The values given are in dBm i.e. it is the measure of the absolute power received by the node at that given point.

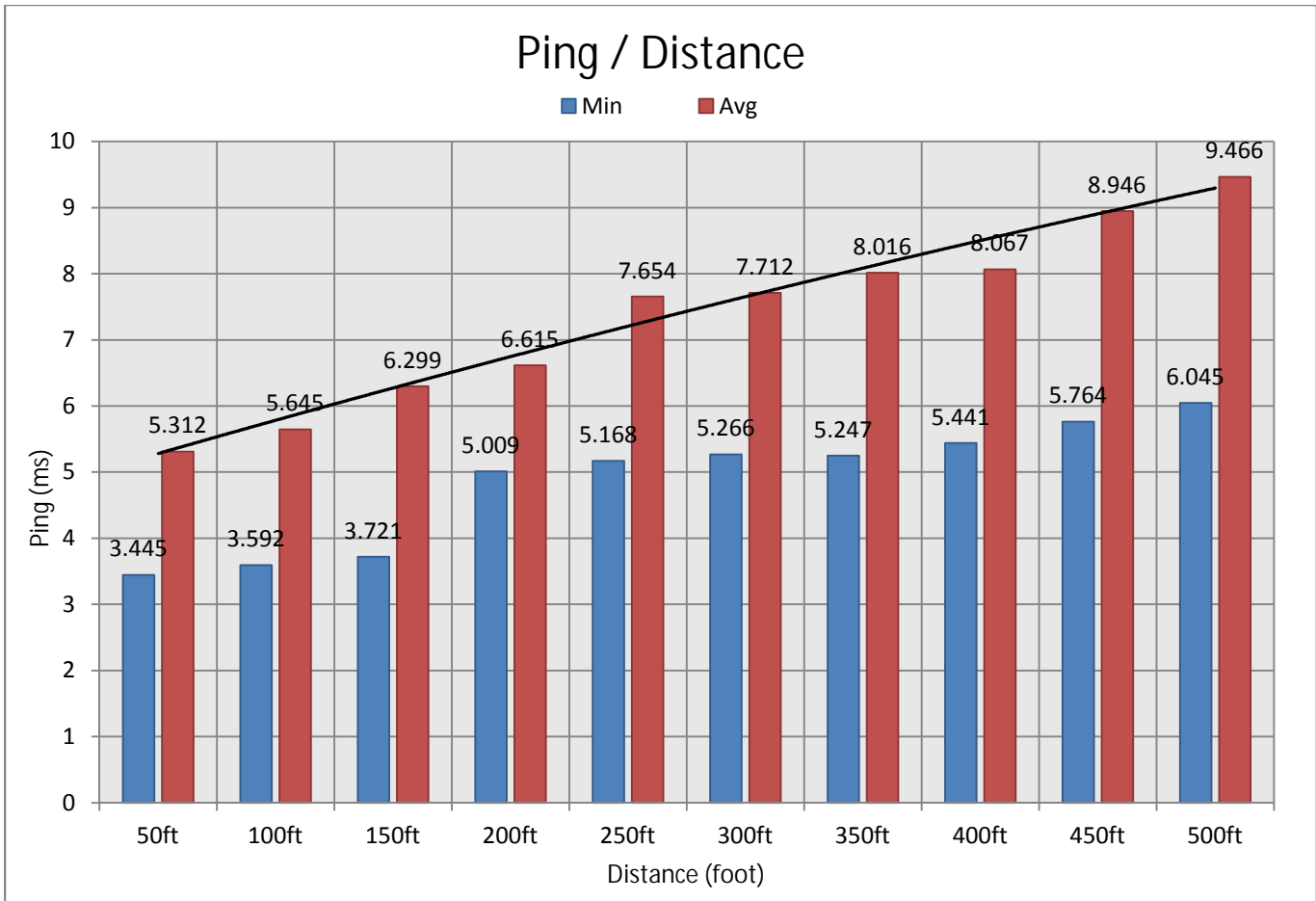


Figure 14: Ping Distance Graph

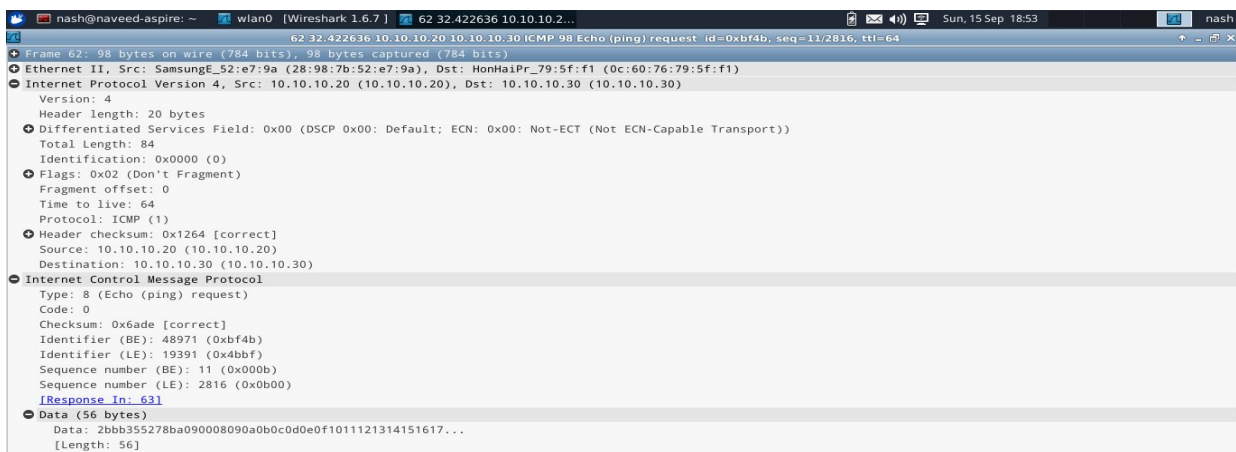


Figure 15: ICMP Packet Detail

Ping was calculated, and live packet was captured for deep packet inspection for checking out the payload of the packet on the network.

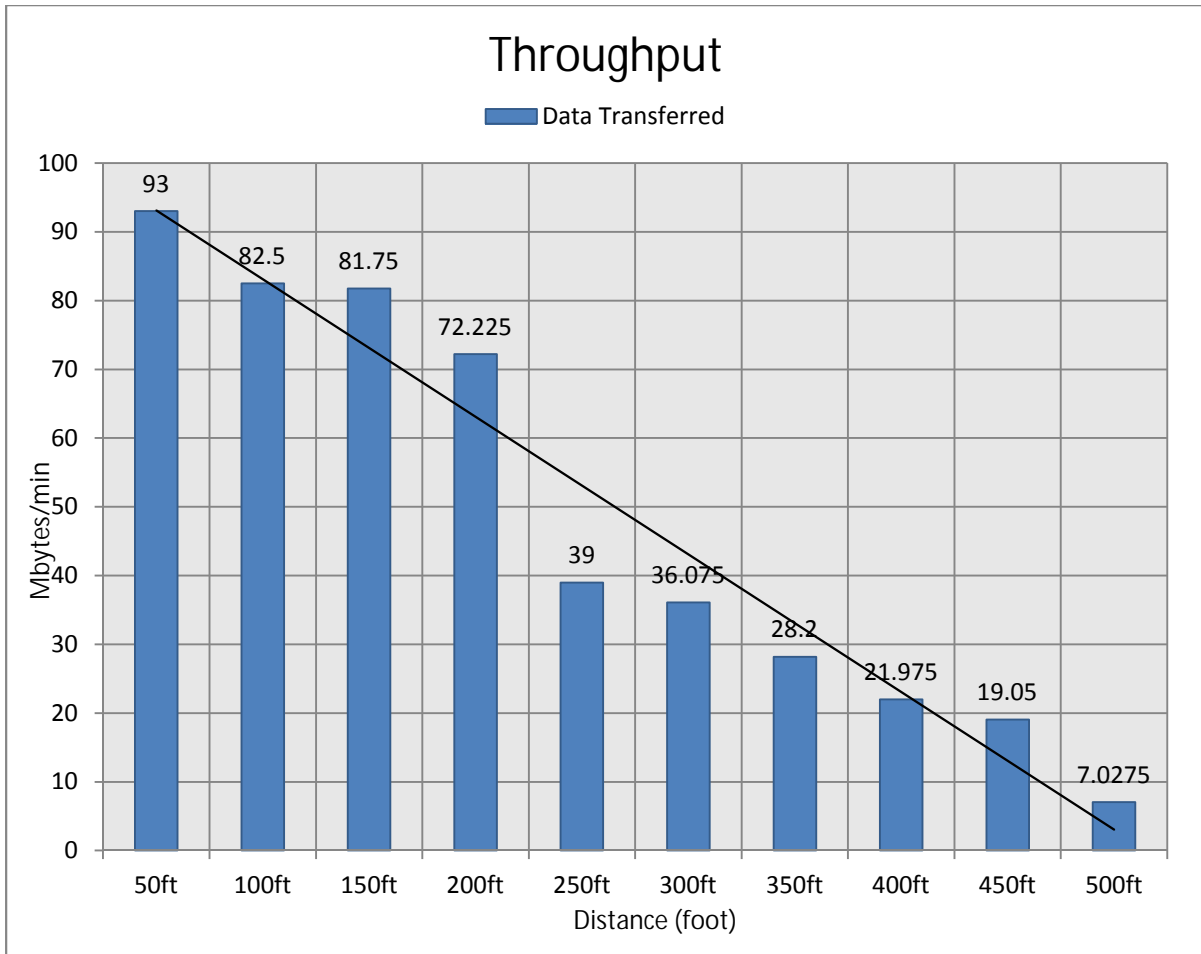


Figure 16: Throughput Distance Graph

```

-s -i 1
-----
Server listening on TCP port 5001
TCP window size: 25.7 KByte (default)
-----
[ 14] local 10.10.10.20 port 5001 connected with
10.10.10.10 port 34151
[ ID] Interval   Transfer   Bandwidth
[ 14] 0.0- 1.0 sec 1.59 MBytes 13.3 Mbits/sec
[ 14] 1.0- 2.0 sec 1.69 MBytes 14.2 Mbits/sec
[ 14] 2.0- 3.0 sec 1.63 MBytes 13.6 Mbits/sec
[ 14] 3.0- 4.0 sec 1.70 MBytes 14.3 Mbits/sec
[ 14] 4.0- 5.0 sec 1.69 MBytes 14.2 Mbits/sec
[ 14] 0.0- 5.0 sec 8.38 MBytes 13.9 Mbits/sec

```

Figure 17: Throughput Distance Reading

An indication of the amount of data transferred at different distances in the first graph, the second displays the readings taken with the help of IPERF.

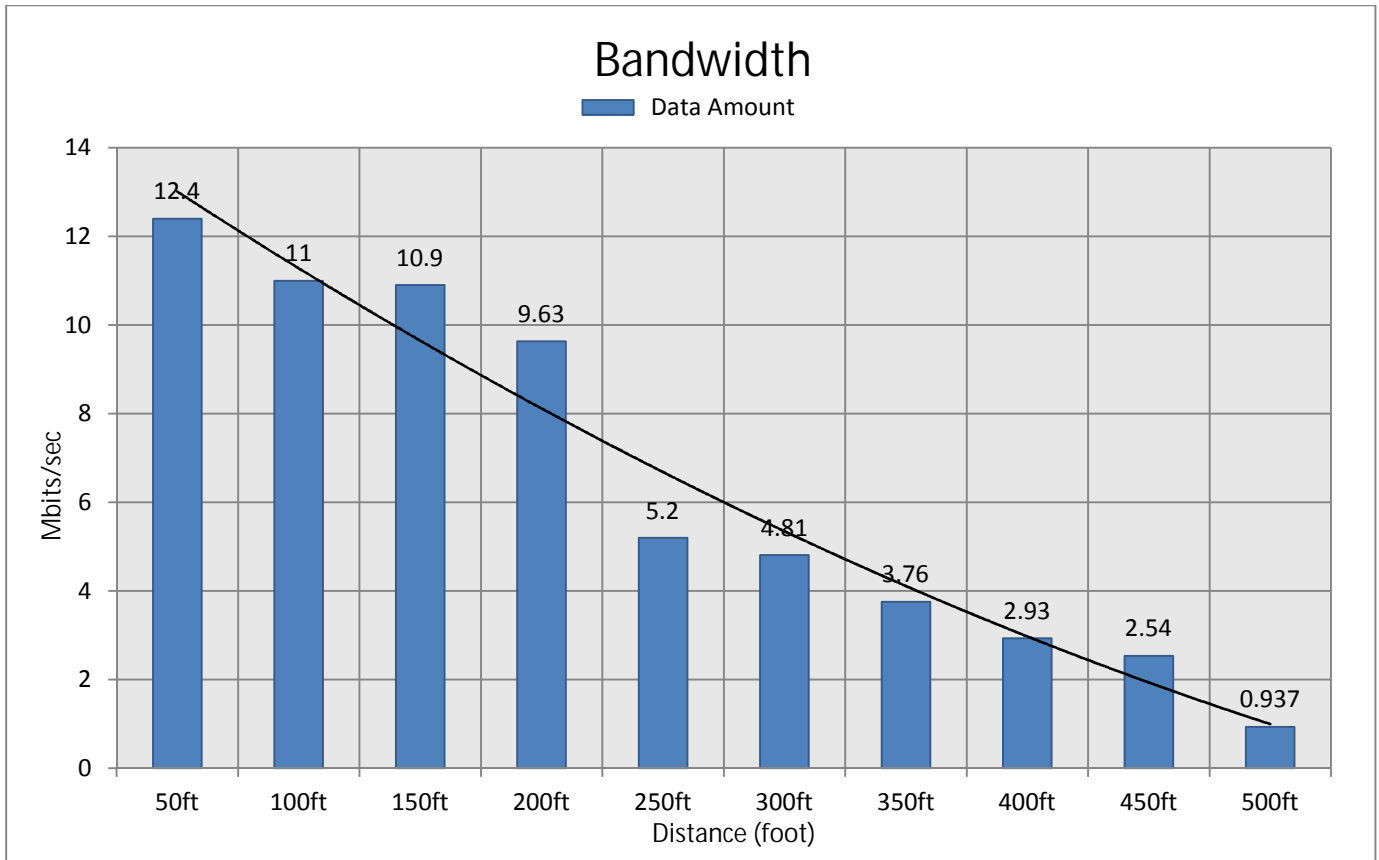


Figure 18: Bandwidth Distance Graph

```

-s -i 1
-----
Server listening on TCP port 5001
TCP window size: 25.7 KByte (default)
-----
[ 14] local 10.10.10.20 port 5001 connected with
10.10.10.10 port 34151
[ID] Interval  Transfer  Bandwidth
[ 14] 0.0- 1.0 sec 1.59 MBytes 13.3 Mbits/sec
[ 14] 1.0- 2.0 sec 1.69 MBytes 14.2 Mbits/sec
[ 14] 2.0- 3.0 sec 1.63 MBytes 13.6 Mbits/sec
[ 14] 3.0- 4.0 sec 1.70 MBytes 14.3 Mbits/sec
[ 14] 4.0- 5.0 sec 1.69 MBytes 14.2 Mbits/sec
[ 14] 0.0- 5.0 sec 8.38 MBytes 13.9 Mbits/sec

```

Figure 19: Bandwidth Distance Reading

This was the most important parameter as it defined the boundary limit of the network, staying under the limit resulted in no, or minimum packet loss.

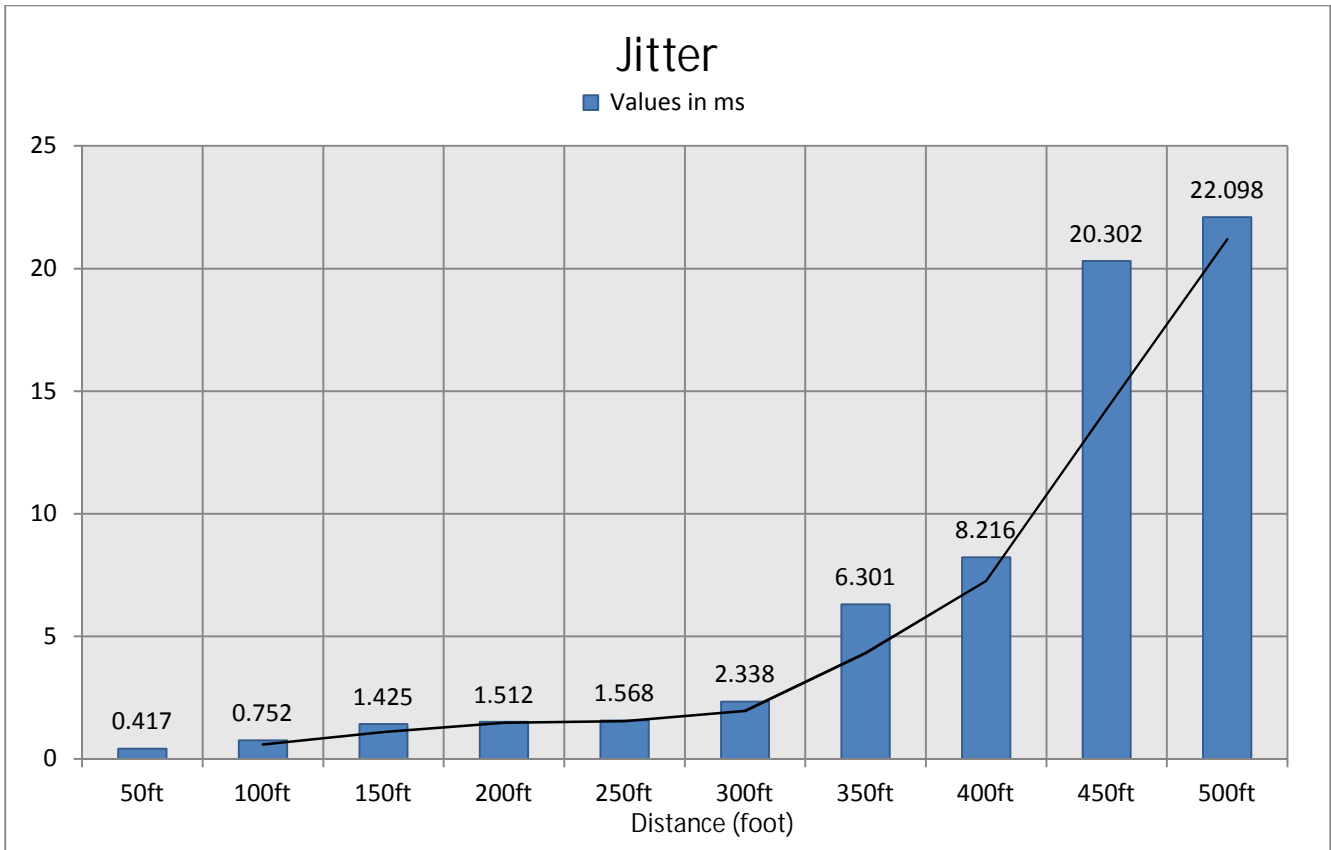


Figure 20: Jitter Distance Graph

Quality	Minimum value in ms	Maximum value in ms
Good	0	20
Average	20	50
Poor	Above 50	-

Table 1: Jitter Value

```
[ 11] local 10.10.10.20 port 5001 connected with 10.10.10.10 port 33126
[ID] Interval  Transfer  Bandwidth  Jitter  Lost/Total Datagrams
[ 11] 0.0- 1.0 sec 128 KBytes 1.05 Mbits/sec 1.052 ms 0/ 89 (0%)
[ 11] 0.0- 1.0 sec 131 KBytes 1.05 Mbits/sec 0.962 ms 0/ 91 (0%)
```

Figure 21: Jitter Value Reading

The first chart shows the amount of jitter experienced by the network, the second displays the ideal value for the VoIP system, the third shows how the readings were taken.

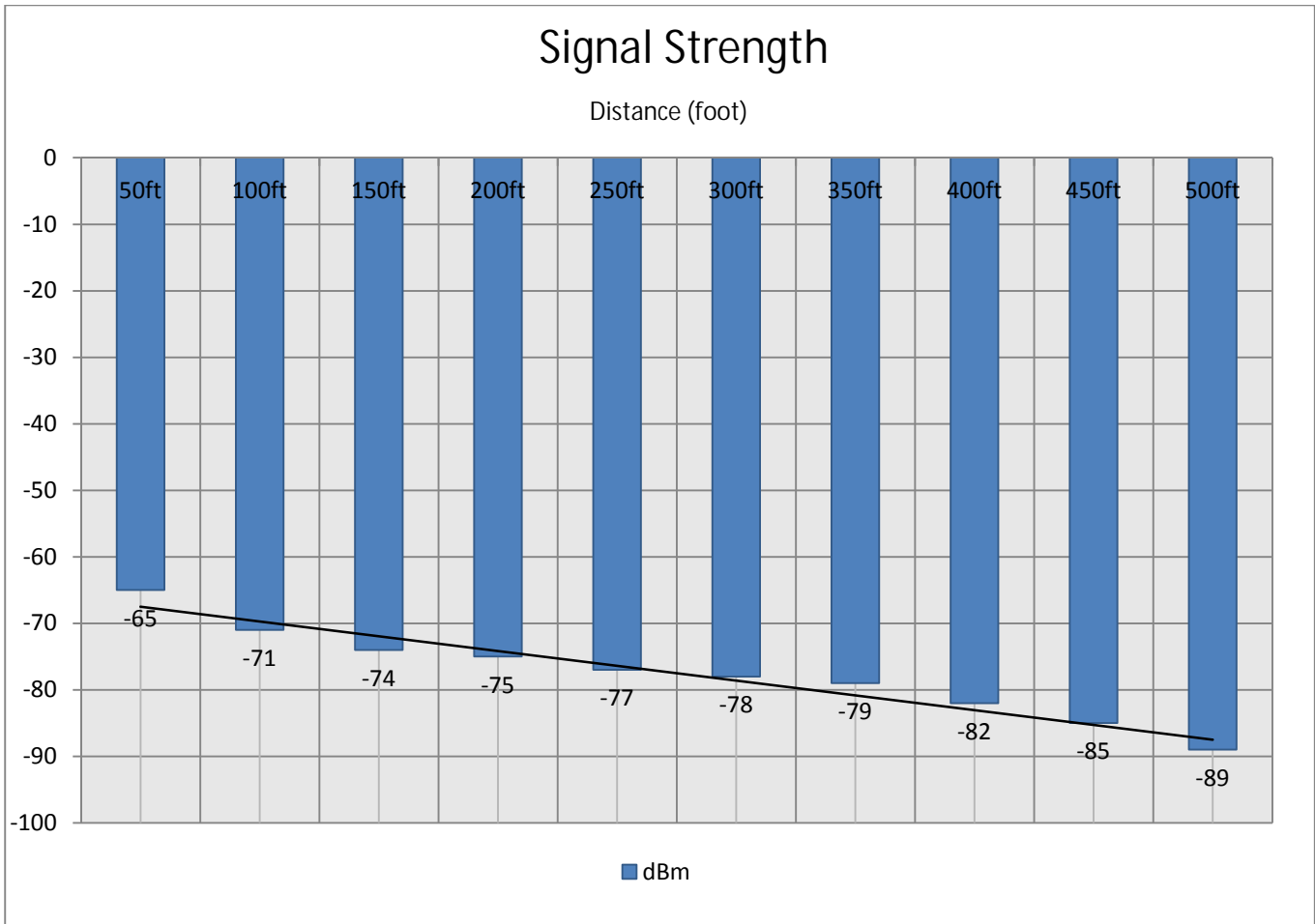


Figure 22: Signal Strength Graph



Figure 23: Signal Strength Readings

These values indicate the absolute power received by the node at different distances. The second shows the readings.

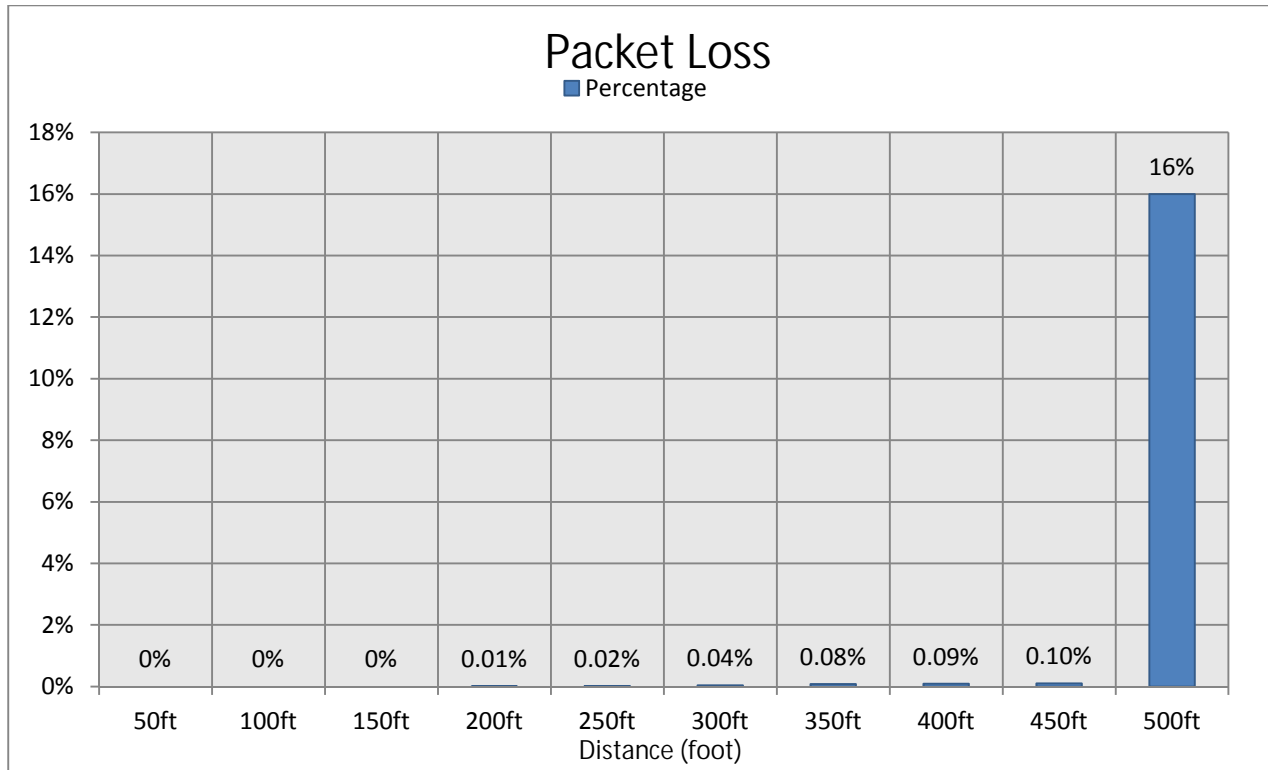


Figure 24: Packet Loss Distance Graph

Quality	Minimum value	Maximum value
Good	0%	0.5%
Average	0.5%	1.5%
Poor	Above 1.5%	-

Table 2: Packet Loss

```

$ ping -c 5 10.10.10.10
PING 10.10.10.10 (10.10.10.10) 56(84) bytes of data:
64 bytes from 10.10.10.10: icmp_seq=1 ttl=64 time=2.59 ms
64 bytes from 10.10.10.10: icmp_seq=2 ttl=64 time=2.92 ms
64 bytes from 10.10.10.10: icmp_seq=3 ttl=64 time=3.17 ms
64 bytes from 10.10.10.10: icmp_seq=4 ttl=64 time=2.83 ms
64 bytes from 10.10.10.10: icmp_seq=5 ttl=64 time=2.80 ms

--- 10.10.10.10 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 2.592/2.865/3.170/0.196 ms
$

```

Figure 25: Packet Loss Reading

This is after Bandwidth calculation the most integral part of readings as packet loss meant lagging in the communication of voice, resulting in bad user experience and the context of the message being lost. The first graph shows the reading taken at specific distances, the second shows the ideal values for packet loss in VoIP service and the third one shows the GUI of the Android when the readings were being noted down.

Quality	User Satisfaction	MOS
Good	Very Satisfied	5
	Satisfied	4
Average	Some users dissatisfied	3
	Many users dissatisfied	2
Poor	Nearly all users dissatisfied	1
	Not Recommended	0

Table 3: MOS



Figure 26: Sound Quality at 3.95kbps

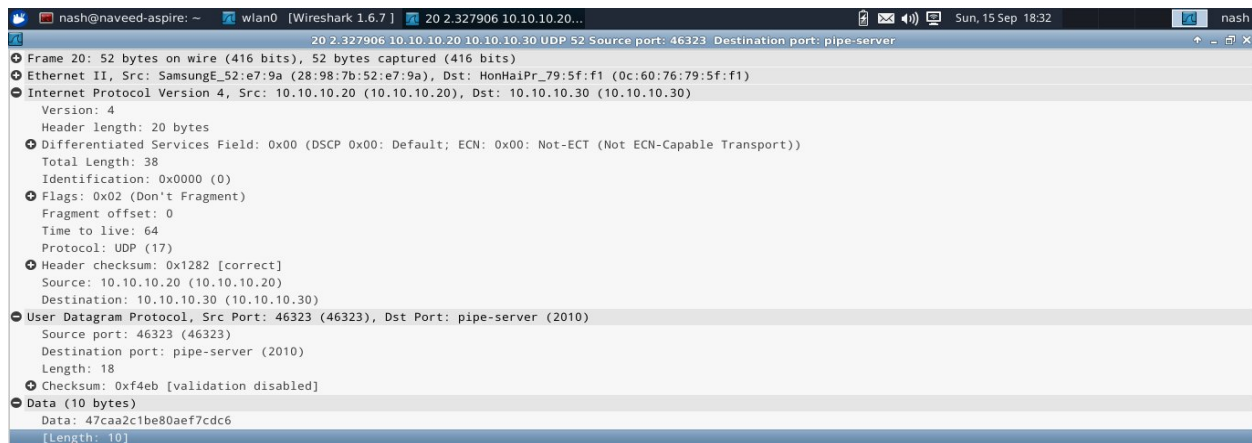


Figure 27: VoIP Packet at 3.95kbps

These tests were taken in real world. The codec used was Speex at 3.95Kbps bit rate setting. The first shows the MOS (user experience grade).The second graph shows the user experience (test conducted on multiple users) at different distances. The third snap shows the packet of the VoIP service.



Figure 28: Sound Quality at 24.6kbps

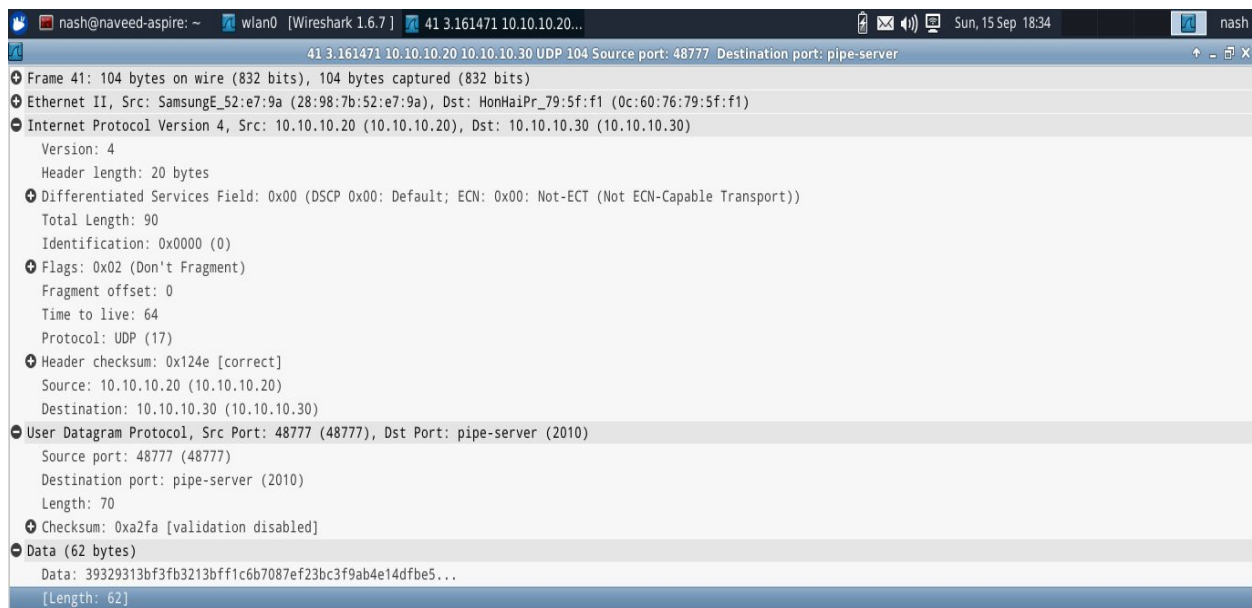


Figure 29: VoIP Packet at 24.6kbps

These tests were taken in real world. The codec used was Speex at 24.6Kbps bit rate setting. The first graph shows the user experience (test conducted on multiple users) at different distances. The second snap shows the packet of the VoIP service.

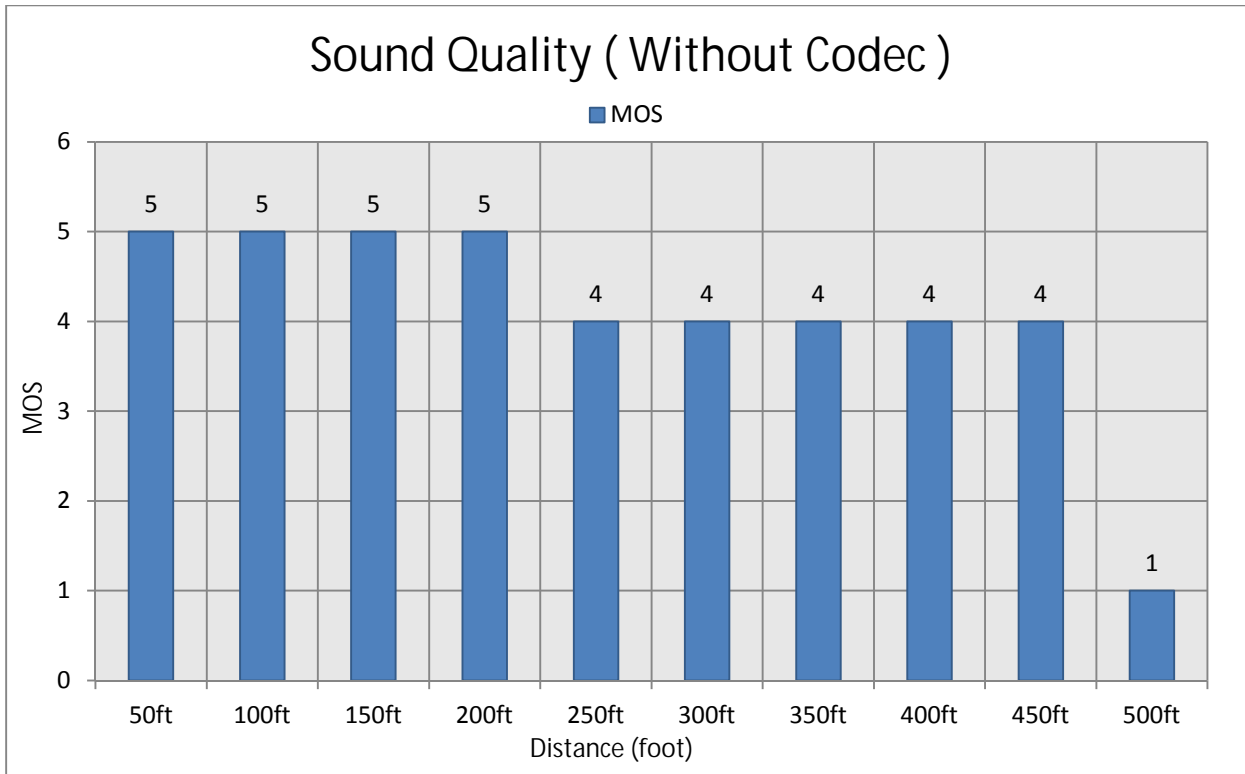


Figure 30: Sound Quality without CODEC

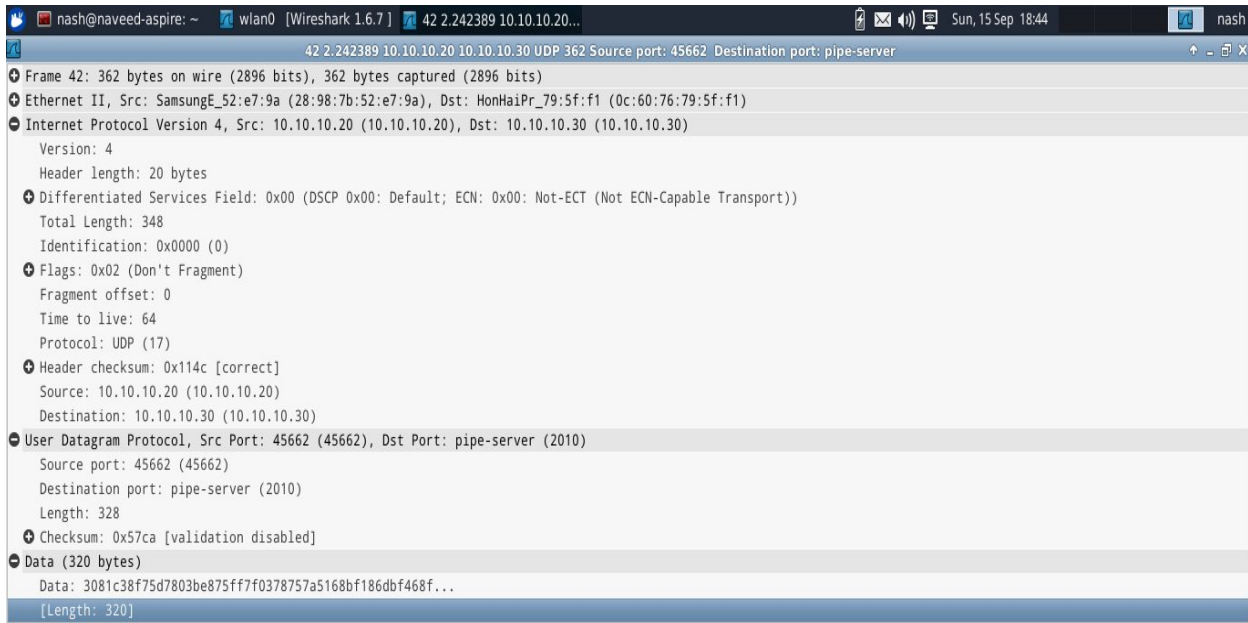


Figure 31: VoIP Packet without CODEC

These tests were taken in real world. In this particular case no codec was used. The first graph shows the user experience (test conducted on multiple users) at different distances. The second snap shows the packet of the VoIP service.

CHAPTER 8: RESULTS II (Contd.)

These values are calculated for the node to bridge to node scenario using line of sight approach i.e. no barriers between node-bridge but node to node are out of sight resulting in 2 hop neighbors. A case scenario illustration is as follows giving the idea of the way these tests were conducted in real world environment.

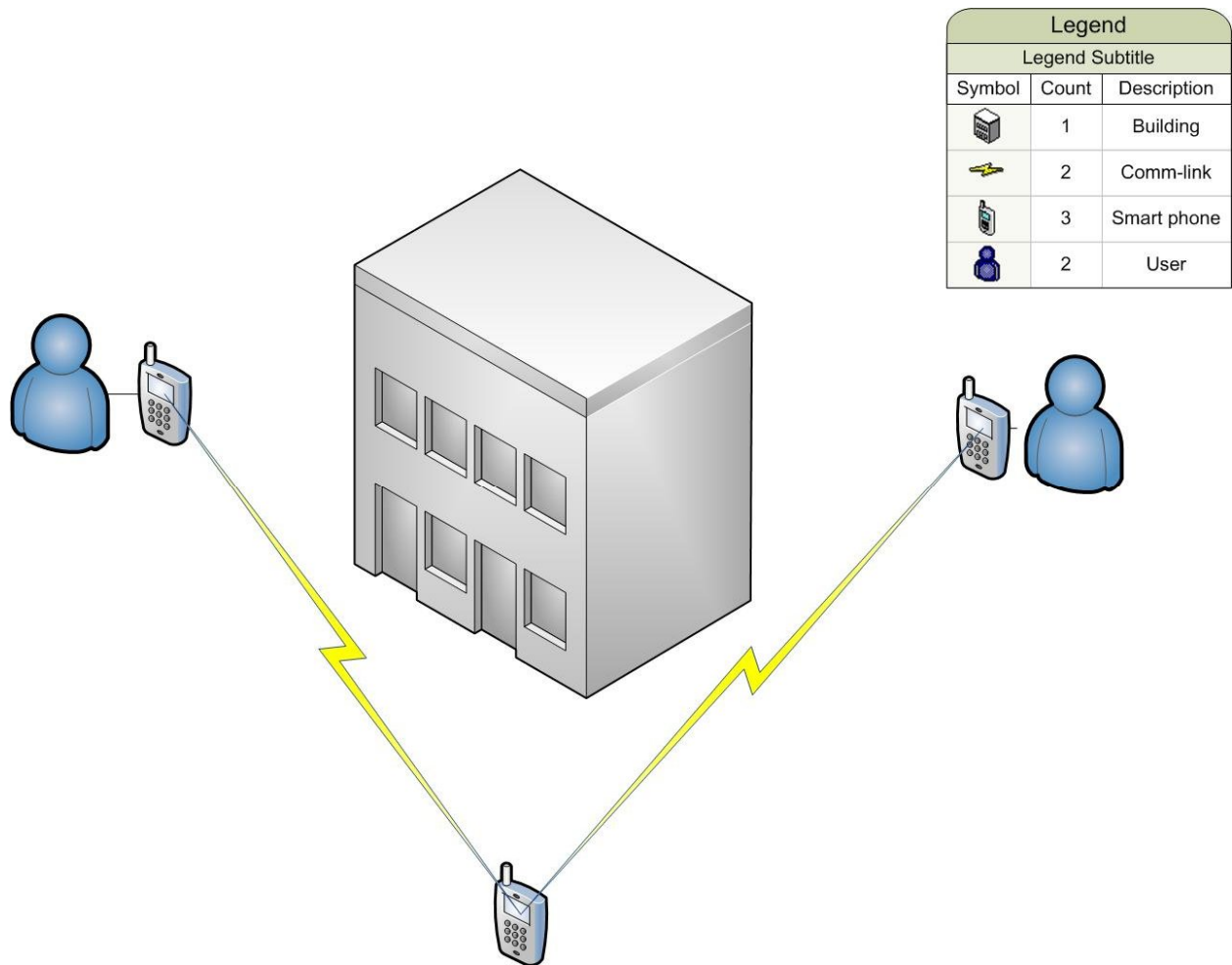


Figure 32: Test Case Scenario 2

The results are calculated on the same pattern as the first test case scenario. But in this case scenario a cell phone has been used as bridge between two nodes which is acting as relay for communication. The bridge is active but is not taking part in the communication nor can it hear any communication between the two end nodes. The maximum distance between the node-bridge is 400 feet while the distance between the two end nodes is collectively 800 feet. The entire tests have been taken starting from 200/200 feet setting and ended up at 400/400 feet.

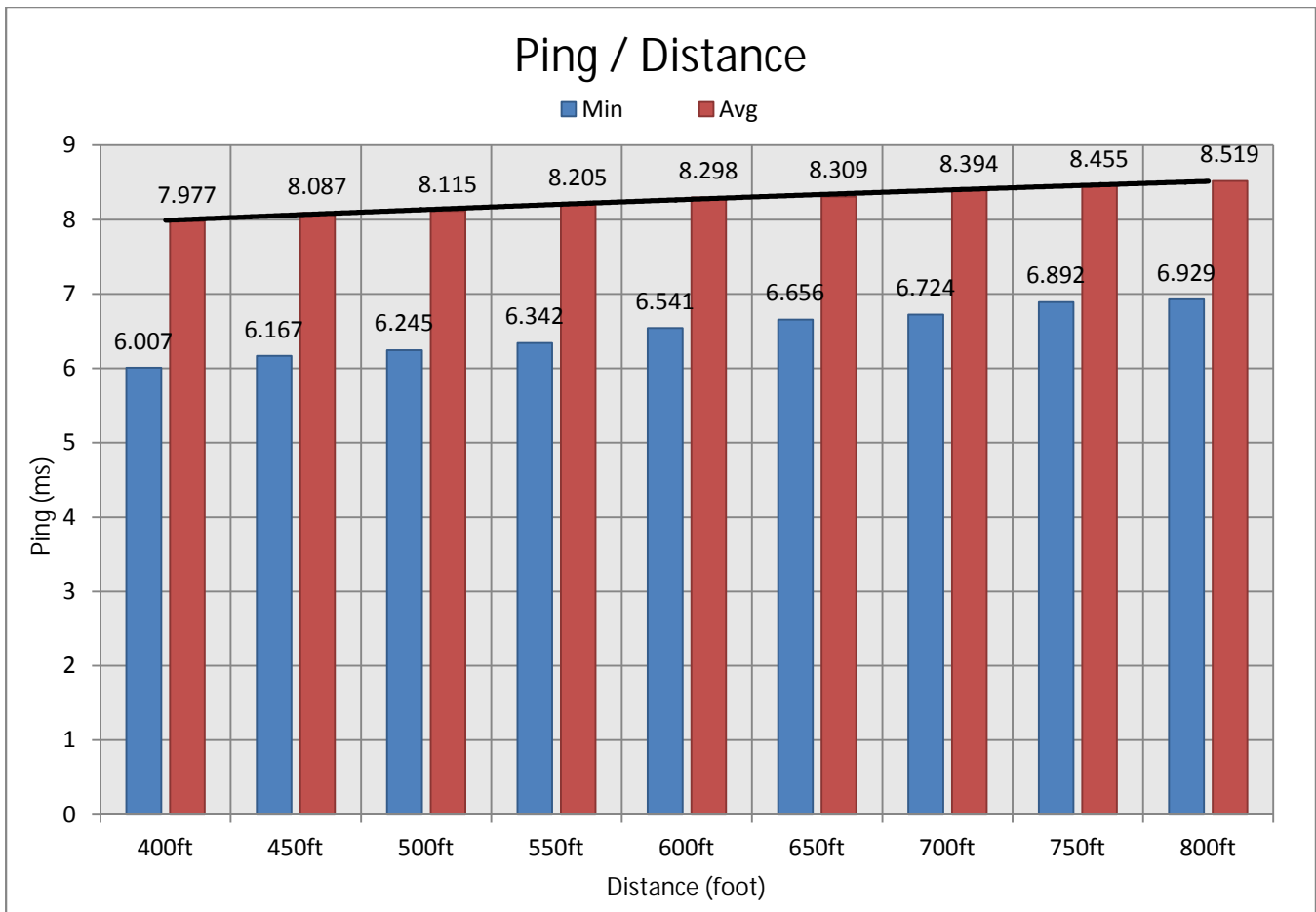


Figure33: Ping Distance Graph

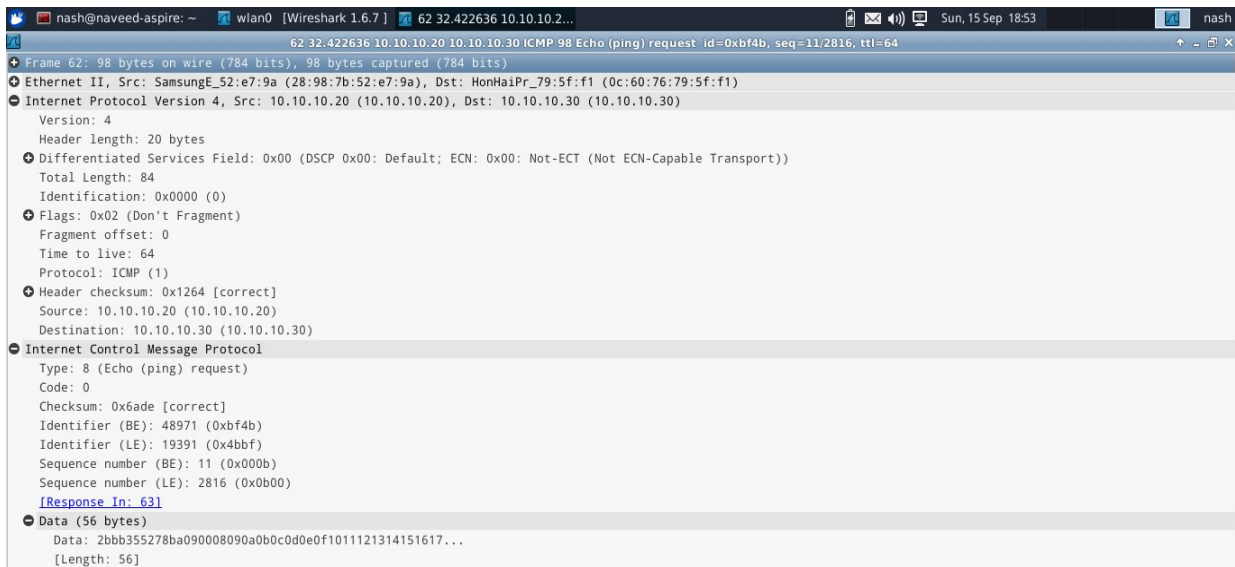


Figure 34: ICMP Packet Detail

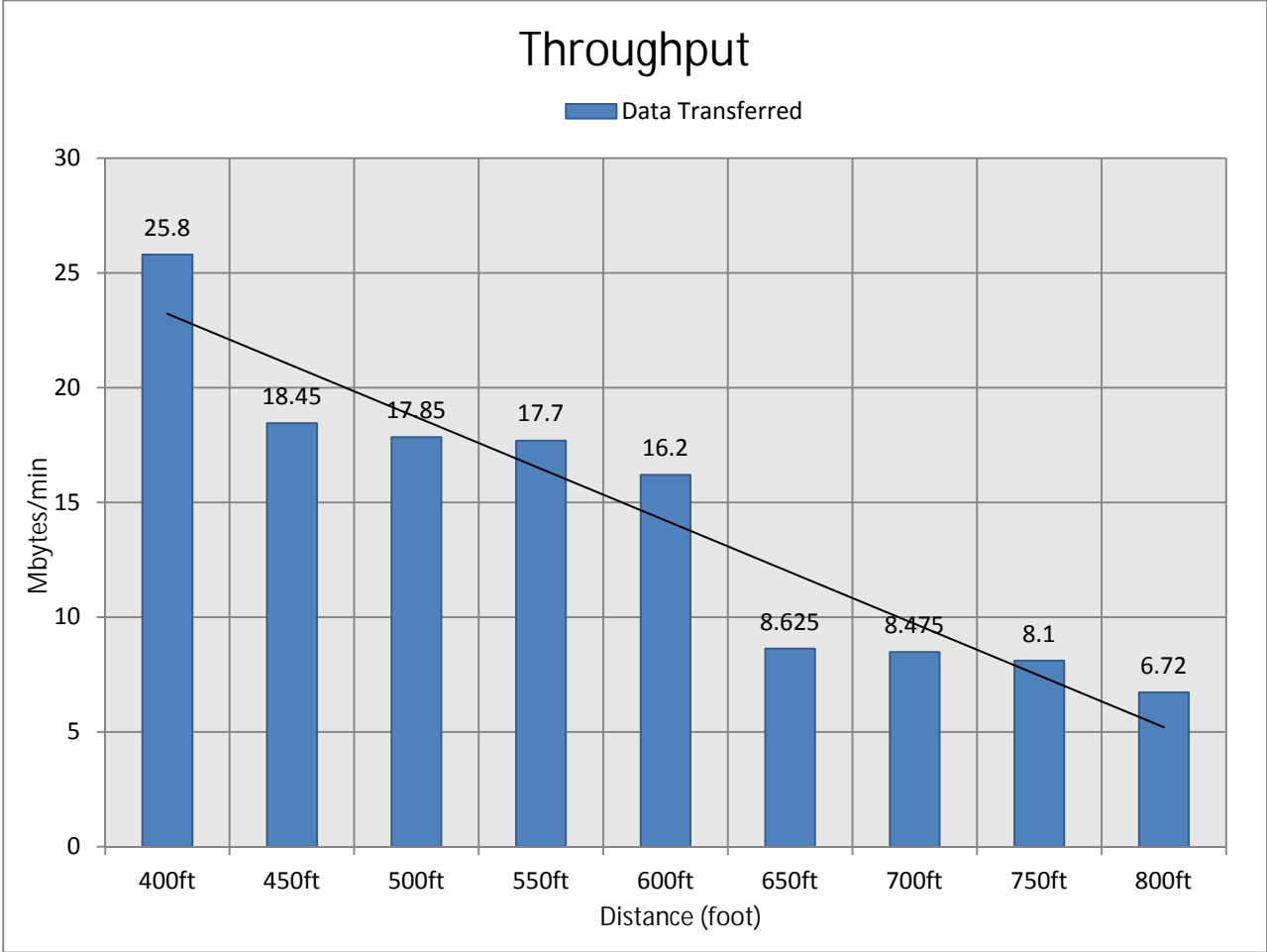


Figure 35: Throughput Distance Graph

```

-s -i 1
-----
Server listening on TCP port 5001
TCP window size: 25.7 KByte (default)
-----
[ 14] local 10.10.10.20 port 5001 connected with
10.10.10.10 port 34151
[ ID] Interval  Transfer  Bandwidth
[ 14] 0.0- 1.0 sec  1.59 MBytes 13.3 Mbits/sec
[ 14] 1.0- 2.0 sec  1.69 MBytes 14.2 Mbits/sec
[ 14] 2.0- 3.0 sec  1.63 MBytes 13.6 Mbits/sec
[ 14] 3.0- 4.0 sec  1.70 MBytes 14.3 Mbits/sec
[ 14] 4.0- 5.0 sec  1.69 MBytes 14.2 Mbits/sec
[ 14] 0.0- 5.0 sec  8.38 MBytes 13.9 Mbits/sec

```

Figure 36: Throughput Distance Reading

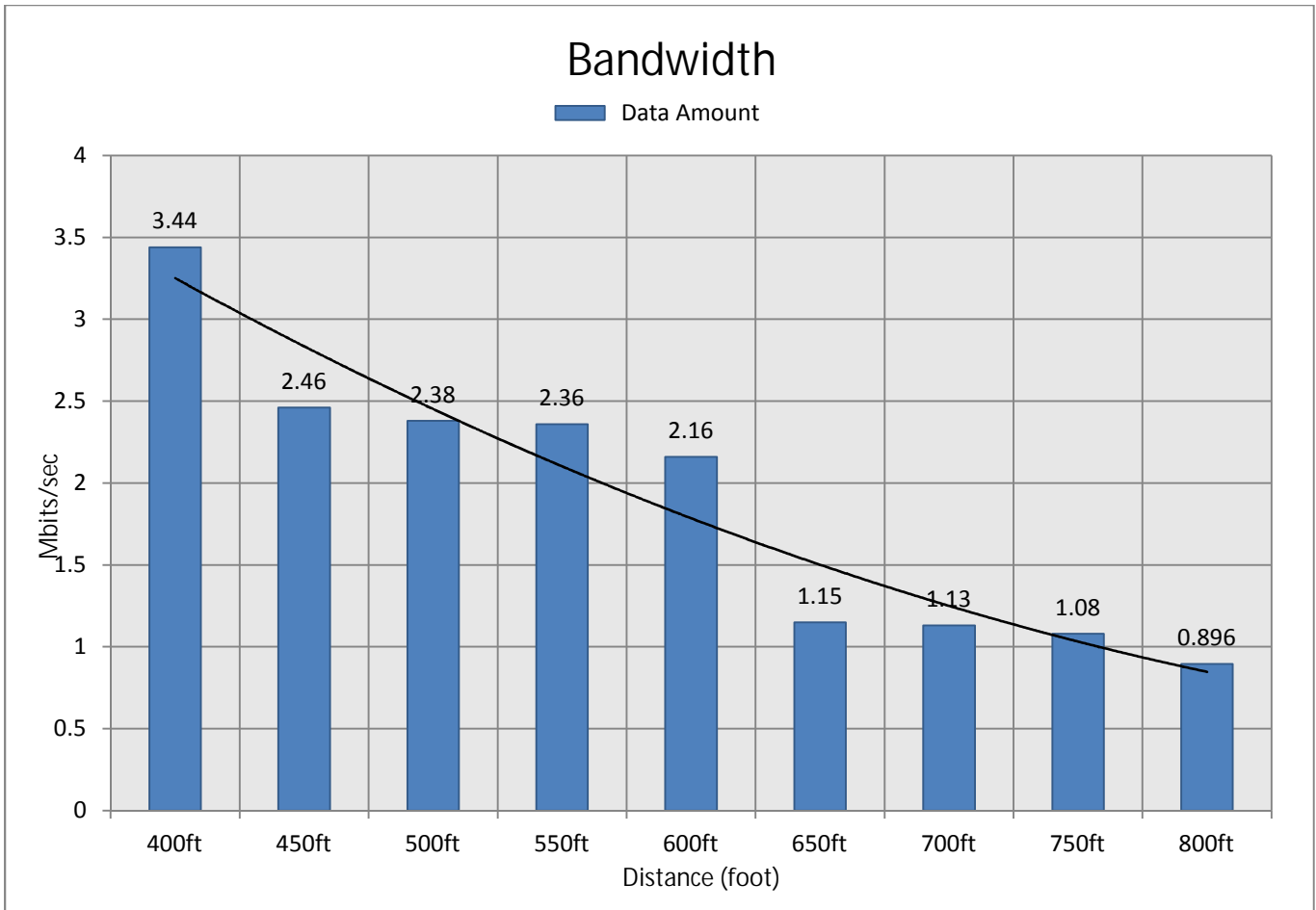


Figure 37: Bandwidth Distance Graph

```

-s -i 1
-----
Server listening on TCP port 5001
TCP window size: 25.7 KByte (default)
-----
[ 14] local 10.10.10.20 port 5001 connected with
10.10.10.10 port 34151
[ ID] Interval   Transfer   Bandwidth
[ 14] 0.0- 1.0 sec 1.59 MBytes 13.3 Mbits/sec
[ 14] 1.0- 2.0 sec 1.69 MBytes 14.2 Mbits/sec
[ 14] 2.0- 3.0 sec 1.63 MBytes 13.6 Mbits/sec
[ 14] 3.0- 4.0 sec 1.70 MBytes 14.3 Mbits/sec
[ 14] 4.0- 5.0 sec 1.69 MBytes 14.2 Mbits/sec
[ 14] 0.0- 5.0 sec 8.38 MBytes 13.9 Mbits/sec

```

Figure 38: Bandwidth Distance Reading

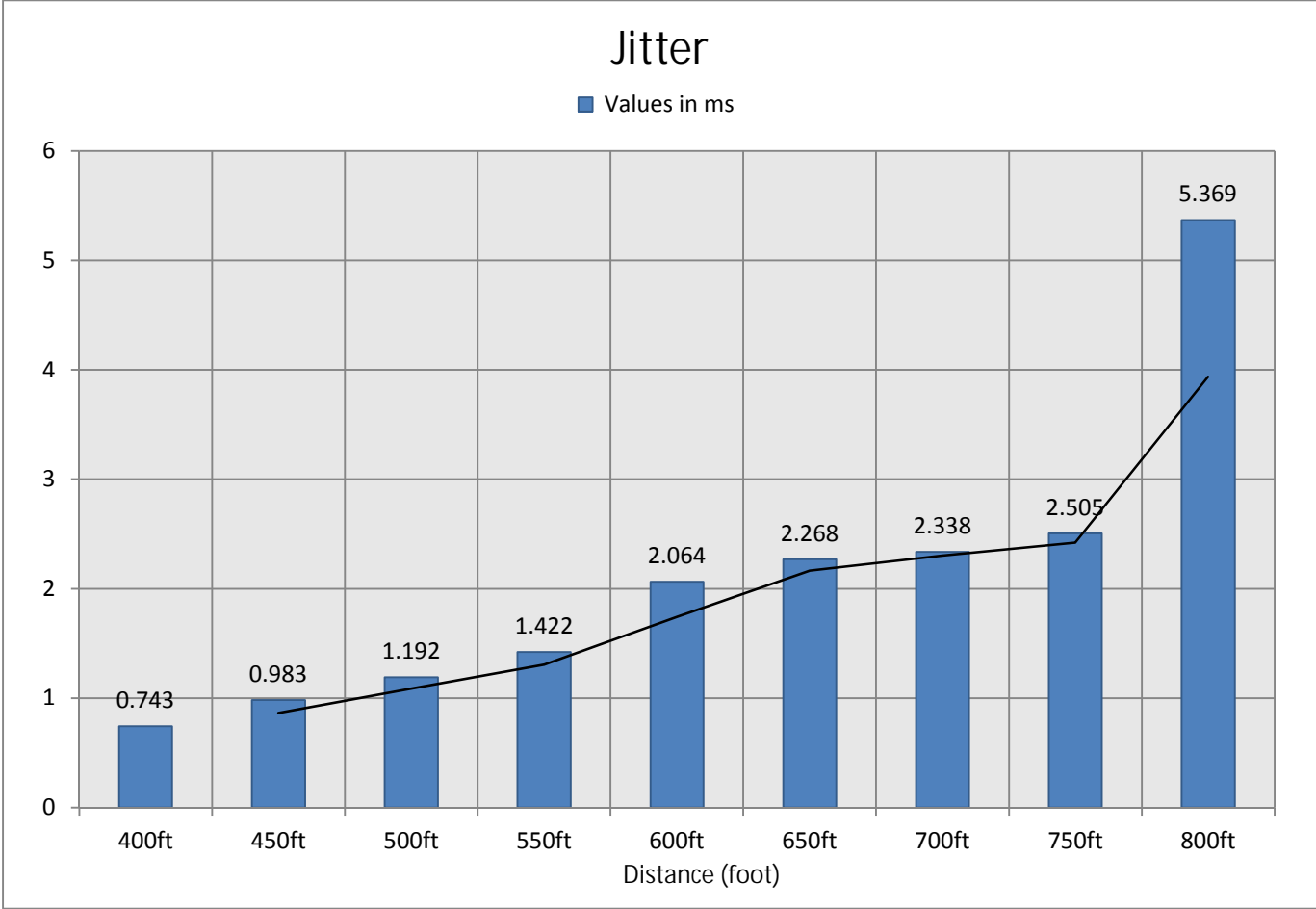


Figure 39: Jitter Distance Graph

Quality	Minimum value in ms	Maximum value in ms
Good	0	20
Average	20	50
Poor	Above 50	-

Table 4: Jitter Value

```
[ 11] local 10.10.10.20 port 5001 connected with 10.10.10.10 port 33126
[ID] Interval  Transfer  Bandwidth  Jitter  Lost/Total Datagrams
[ 11] 0.0- 1.0 sec 128 KBytes 1.05 Mbts/sec 1.052 ms 0/ 89 (0%)
[ 11] 0.0- 1.0 sec 131 KBytes 1.05 Mbts/sec 0.962 ms 0/ 91 (0%)
```

Figure 40: Jitter Value Reading

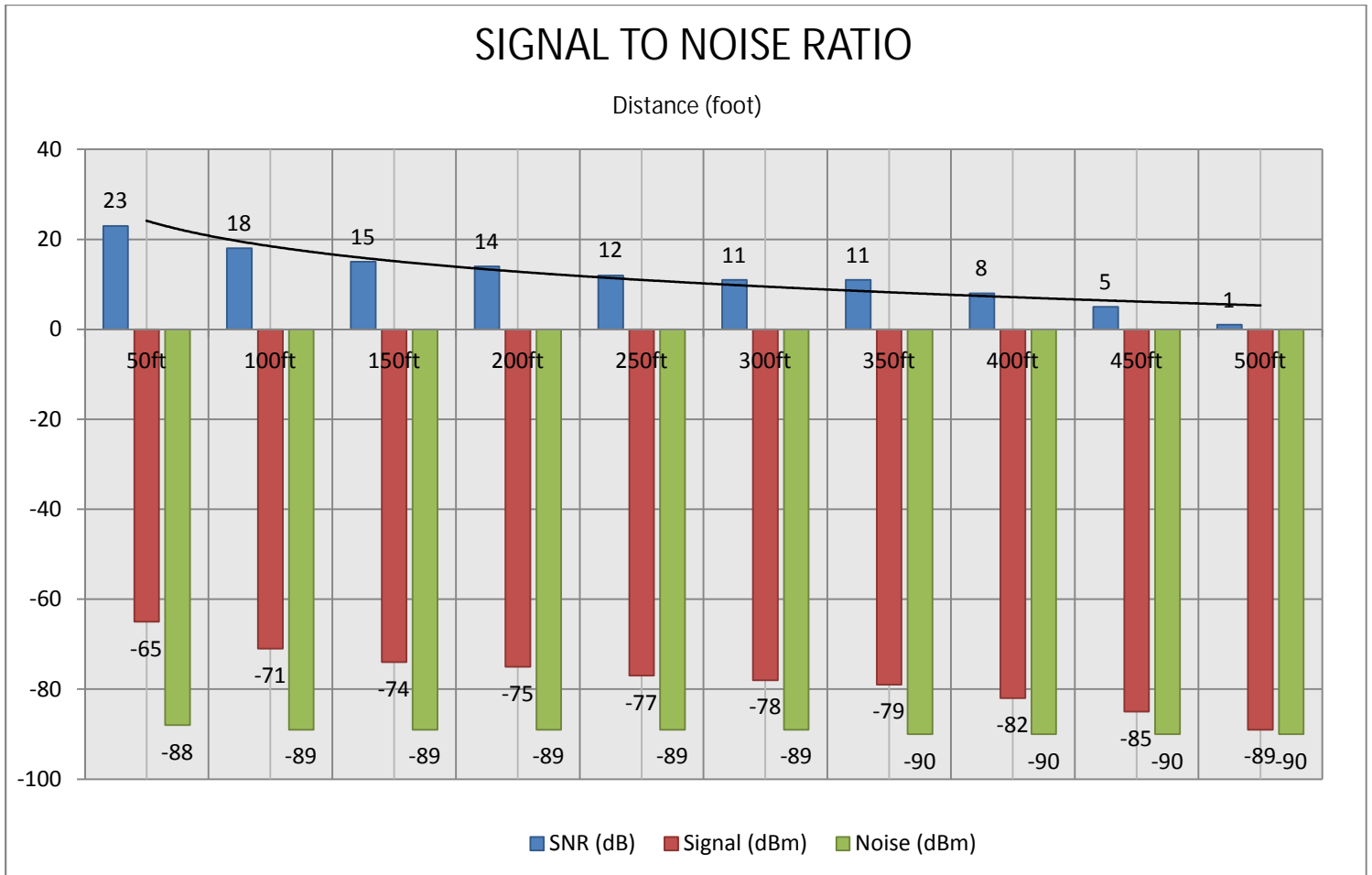


Figure 41: Signal to Noise Ratio Graph

```

sil0      no wireless extensions.

eth0      IEEE 802.11-DS  ESSID:"sheikh"  Nickname:""
          Mode:Ad-Hoc  Frequency:2.412 GHz  Cell: 16:
          21:72:1D:21:D0
          Bit Rate=65 Mb/s  Tx Power=32 dBm
          Retry min limit:7  RTS thr:off  Fragment
          thr:off
          Power Managementmode:All packets received
          Link Quality=5/5  Signal level=-37 dBm  Noi
          se level=-91 dBm
          Rx invalid mwid:0  Rx invalid crypt:0  Rx i
          nvalid frag:0
          Tx excessive retries:0  Invalid misc:0  Mi
          ssed beacon:0
#

```

Figure 42: Signal to Noise Ratio Reading

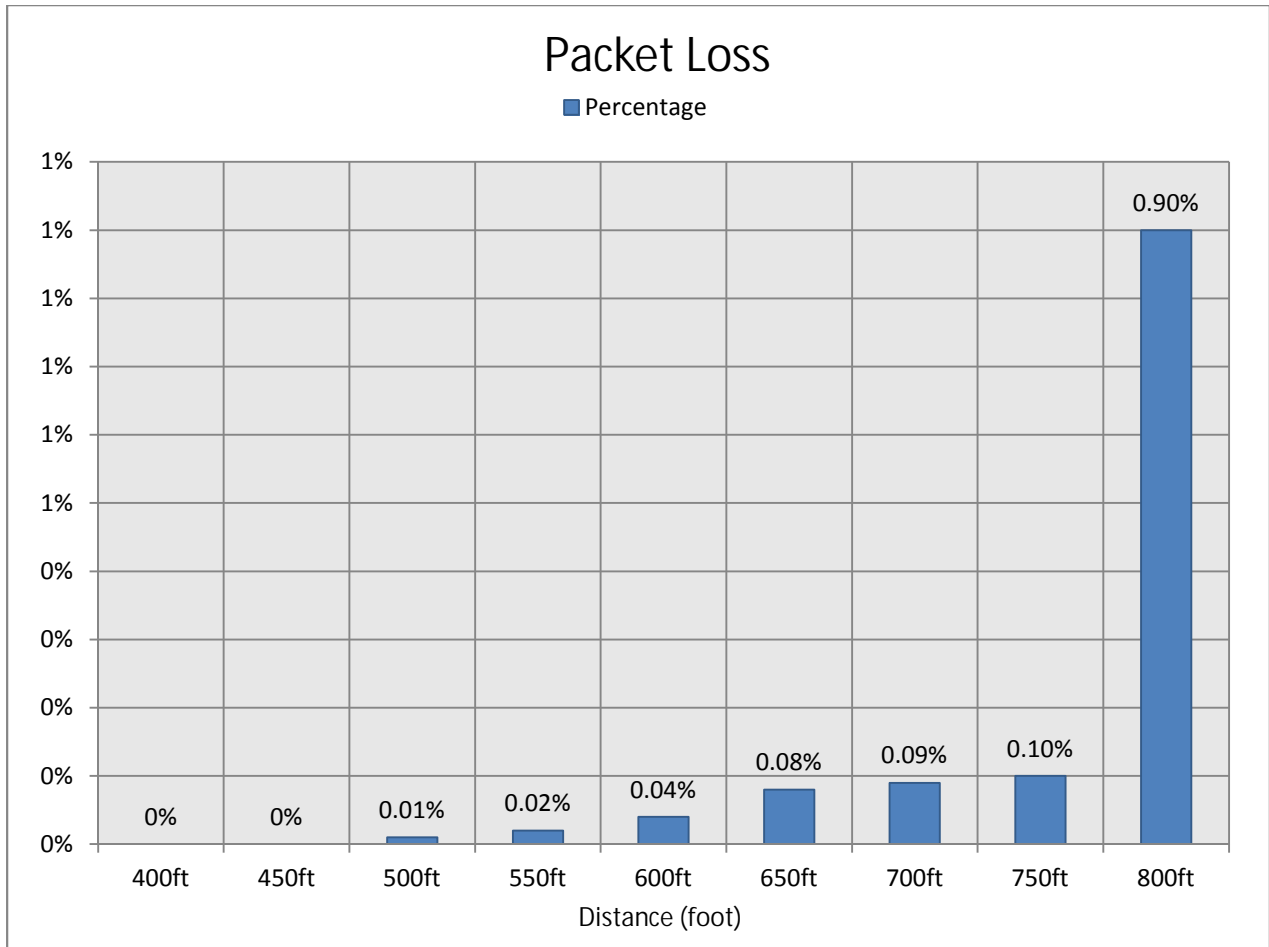


Figure 43: Packet Loss Distance Graph

Quality	Minimum value	Maximum value
Good	0%	0.5%
Average	0.5%	1.5%
Poor	Above 1.5%	-

Table 5: Packet Loss

```

$ ping -c 5 10.10.10.10
PING 10.10.10.10 (10.10.10.10) 56(84) bytes of data.
64 bytes from 10.10.10.10: icmp_seq=1 ttl=64 time=2.59 ms
64 bytes from 10.10.10.10: icmp_seq=2 ttl=64 time=2.92 ms
64 bytes from 10.10.10.10: icmp_seq=3 ttl=64 time=3.17 ms
64 bytes from 10.10.10.10: icmp_seq=4 ttl=64 time=2.83 ms
64 bytes from 10.10.10.10: icmp_seq=5 ttl=64 time=2.80 ms

--- 10.10.10.10 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 2.592/2.865/3.170/0.196 ms
$

```

Figure 44: Packet Loss Reading

Quality	User Satisfaction	MOS
Good	Very Satisfied	5
	Satisfied	4
Average	Some users dissatisfied	3
	Many users dissatisfied	2
Poor	Nearly all users dissatisfied	1
	Not Recommended	0

Table 6: MOS

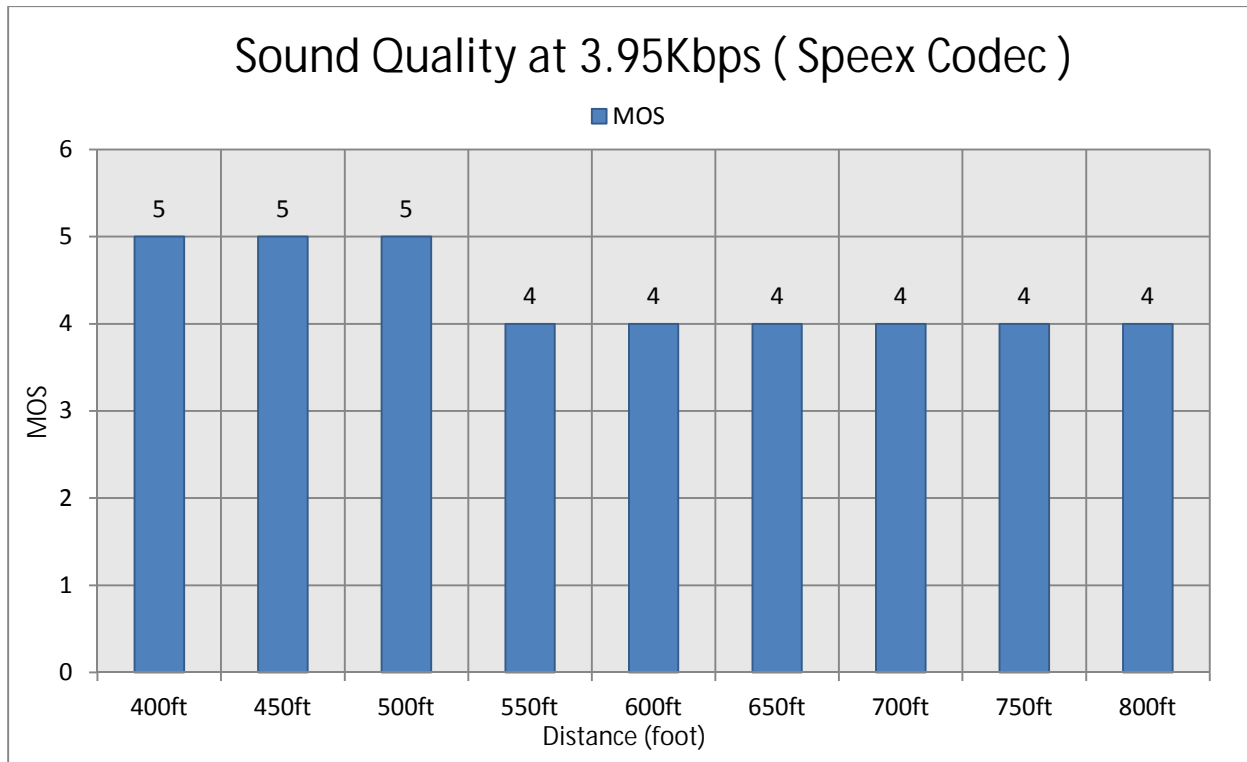


Figure 45: Sound Quality at 3.95kbps



Figure 46: VoIP Packet at 3.95kbps

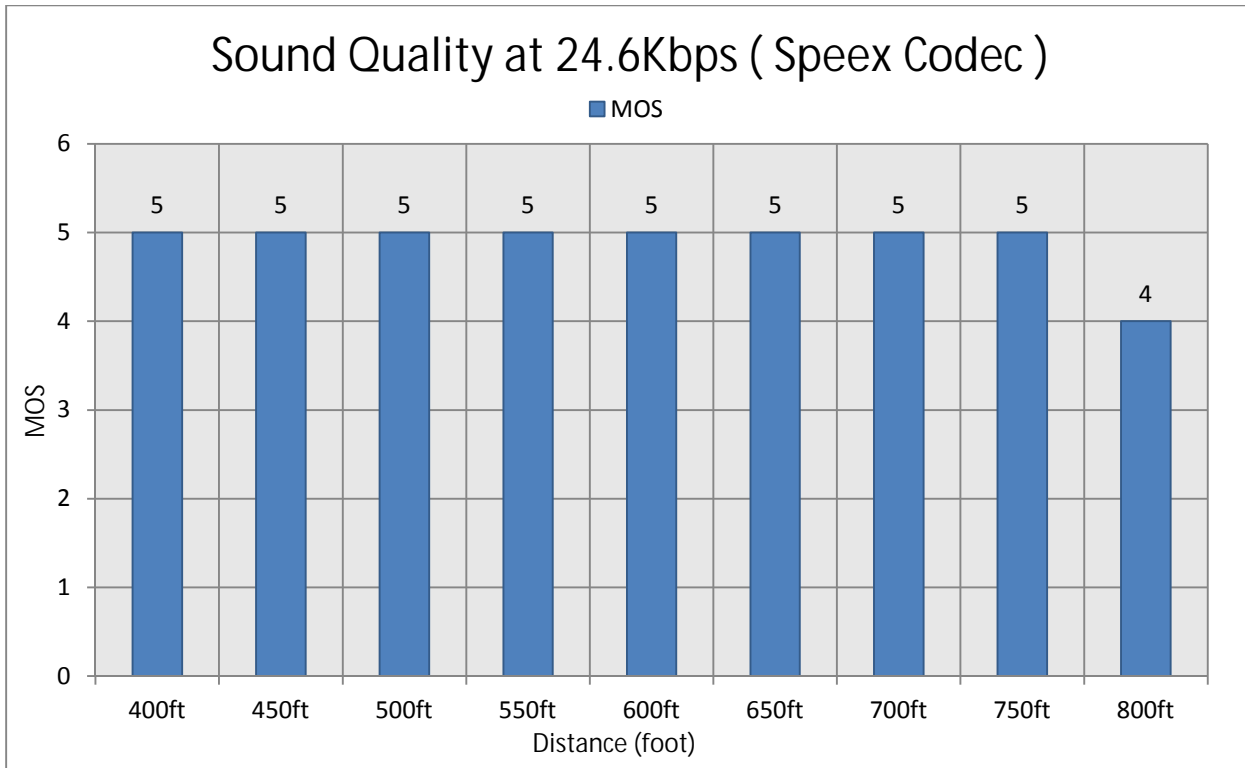


Figure 47: Sound Quality at 24.6kbps

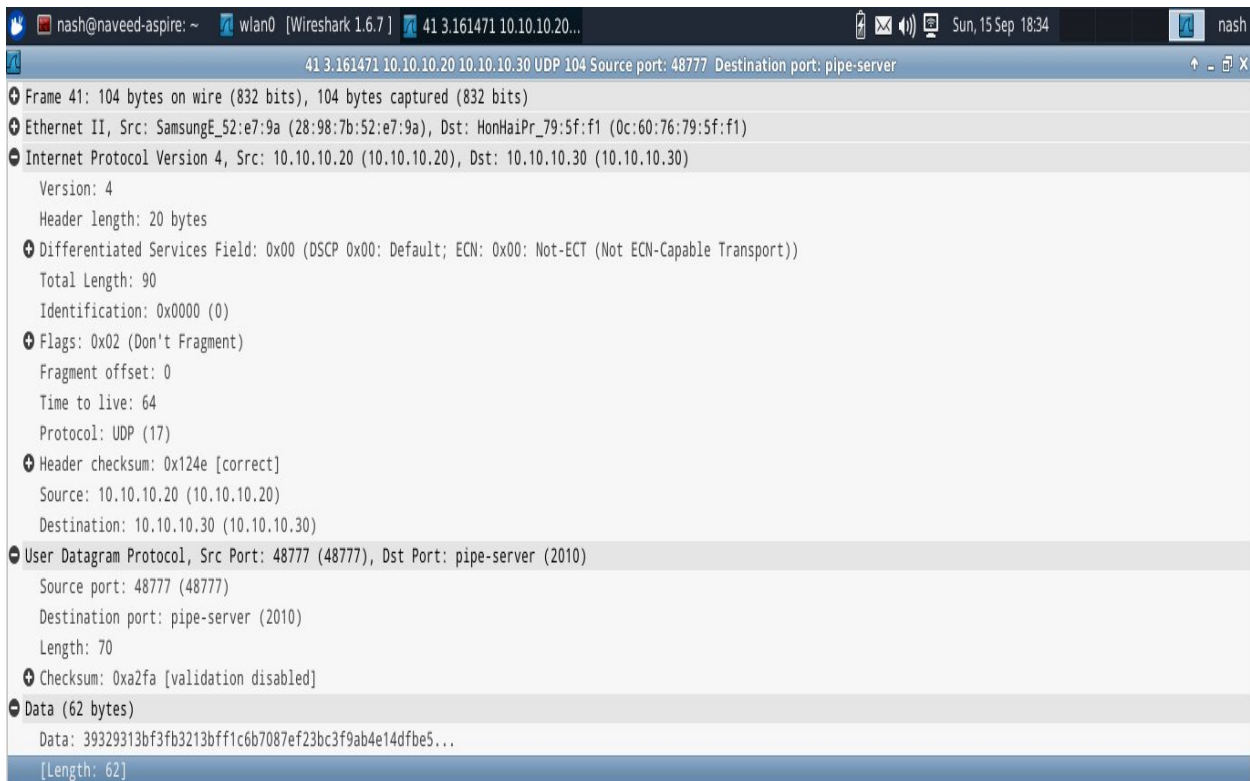


Figure 48: VoIP Packet at 24.6kbps

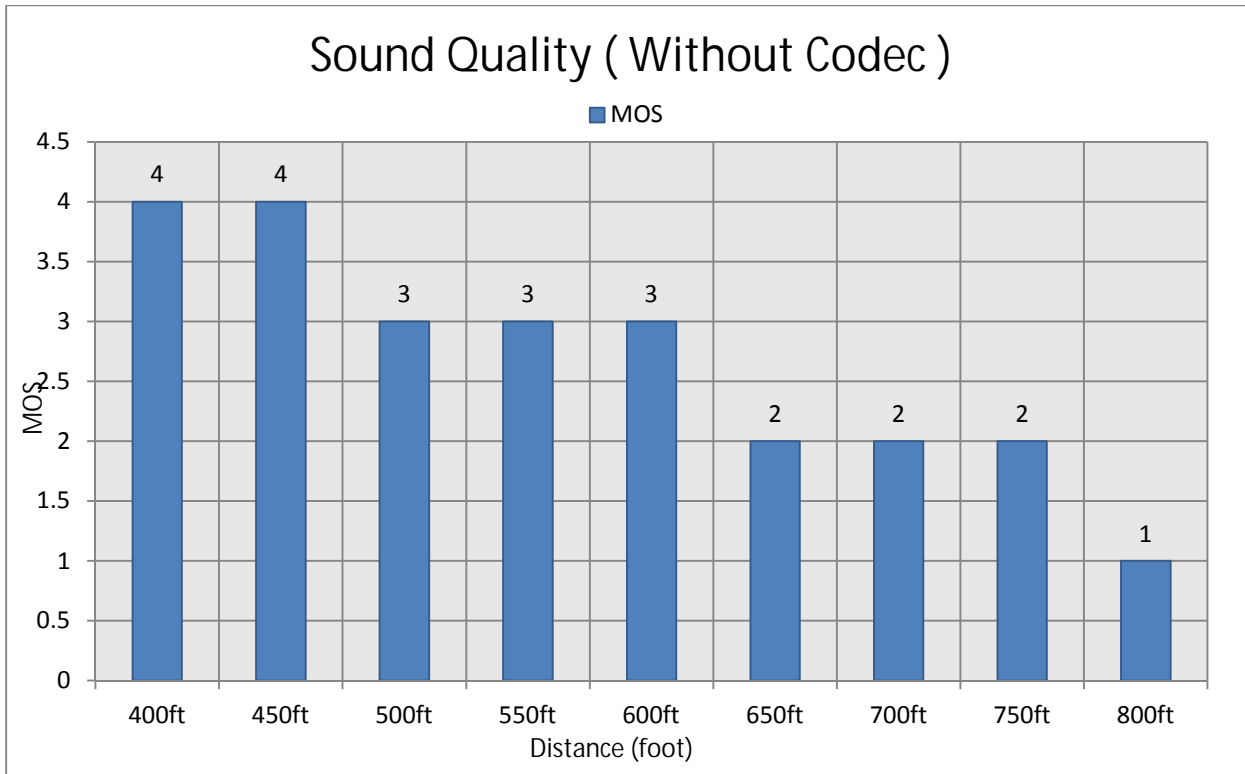


Figure 49: Sound Quality without CODEC

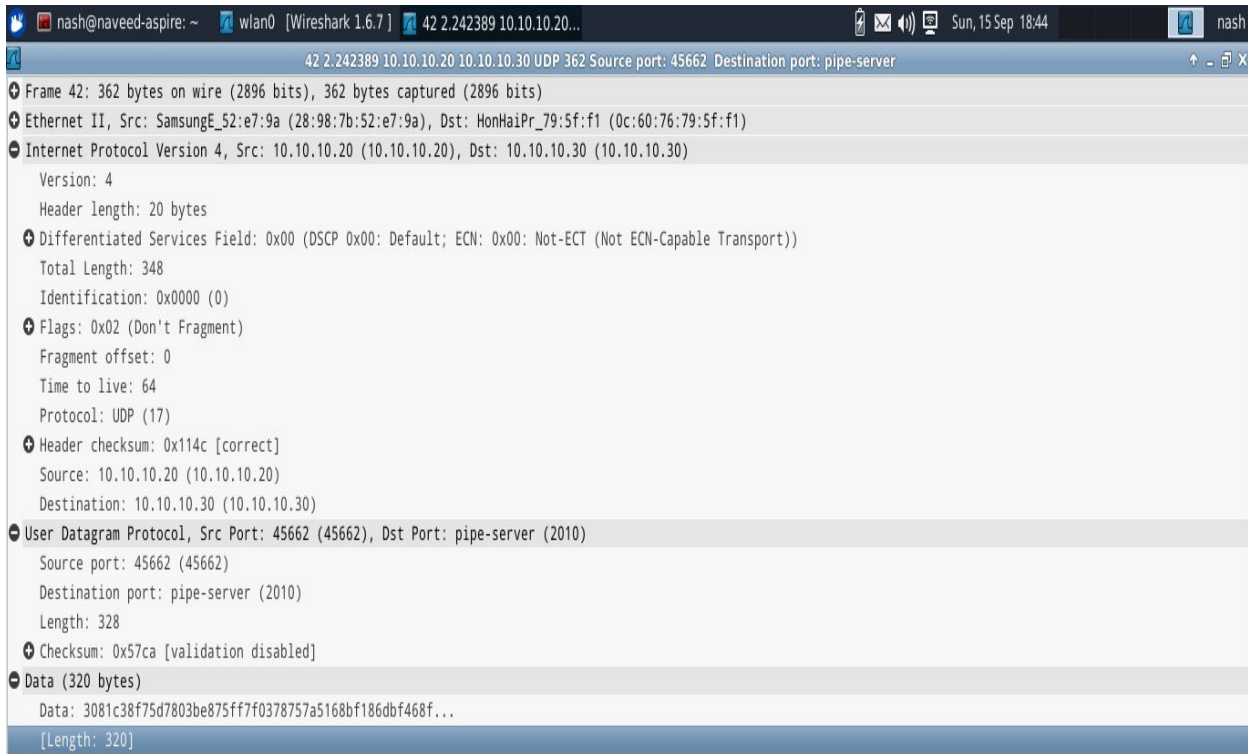


Figure 50: VoIP Packet without CODEC

CHAPTER 9: CONCLUSION

We have successfully developed a prototype system which has established secure VoIP communication and also it provides an extremely flexible method for VoIP in mobile ad hoc networks. Hence, we can safely conclude that we were able to establish a cooperative mesh based mobile Ad-hoc networks for rapidly deployable VoIP communication with survivable, efficient dynamic networking.

CHAPTER 10: FUTURE WORK

Based on our implementation and study we found that below areas are still open that can be considered as a future work.

1) Dynamic IP Addressing

In our Thesis only a single person assign the IP address; this can be enhanced like Microsoft Windows APIPA service of automatic IP assigning based on MAC Addresses.

2) Encryption of Packets

The existing system does not include any type of encryption; Packet can be encrypted using any famous algorithms.

3) More Secure Network

A service can be implemented which will be able to hide the nodes making the mesh network seemingly invisible for tapping or attacks.

4) Shifting from Half Duplex to Full Duplex Mode

As of current scenario the system only supports a half-duplex VoIP service the enhancement can be done to make it work in Full Duplex Mode.

5) Increasing GUI Feature

The base application of Ad-Hoc Networking lacks many GUI feature like modifying the OLSR from within - or on the fly making it more flexible, also adding OLSR status view from within the application will be added advantage.

6) Dialing Feature / Application Merger

The VoIP application is very simple; it can be enhanced like Skype by adding a feature of dialing on IP or on username to initiate communication. The VoIP application can be imbedded into the base Ad-Hoc application for a better user experience.

References:

[1] SHEIKH, R.; SINGH CHANDE, M.; KUMAR MISHRA, D., "SECURITY ISSUES IN MANET: A REVIEW" WIRELESS AND OPTICAL COMMUNICATIONS NETWORKS (WOCN), 2010 SEVENTH INTERNATIONAL CONFERENCE, SEPTEMBER 2010 76

[2] http://www.us-cert.gov/reading_room/understanding_VoIP.pdf

[3] Yi Sun, Gengfa Fang, Jinglin Shi, "Research on the Implementation of VoIP Service in Mobile Ad-hoc Network", CNKI, 2005

[4] S. Corson, J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", January 1999.

[5] Adamson, B., "Tactical Radio Frequency Communication Requirements for IPng", RFC 1677, August 1994

[6] Jiazi Yi, "A survey on the application of MANETS", Polytech Nantes, February 2008

[7] Malik P. "Consequences of Limited MANET Resources", International Journal of Advanced Research in Computer Science and Software Engineering, Vol 2, Issue 1, January 2012

[8] Keshtgary M. Babaiyan V. "Performance Evaluation for Reactive, Proactive and Hybrid Routing Protocols for MANET", International Journal on Computer Science and Engineering, Vol 4, Issue 2, February 2012

[9] On Self Adaptive Routing in Dynamic Environments An Evaluation and Design Using a Simple, Probabilistic Scheme Haiyong Xie Lili Qiu Y ang Richard Y ang and Yin Zhang Computer Science Department, Yale University , New Haven, CT 06520

[10] <http://www.transnexus.com/index.php/issue-5-january-2013/four-VoIP-trends-to-watch-for-in-2013>

[11] <http://www.VoIP-news.com/articles/VoIP-blog/developing-a-comprehensive-approach-to-VoIP-security-55899>

[12] <http://shrikantandroidinfo.blogspot.com/2013/01/android-architecture.html>

[13] Security Threats in Mobile Ad Hoc Networks. / Sen, S; Clark, JA. In: www-users.cs.york.ac.uk, 2009.