

VoIP Strategies for Hybrid MANETs

Prakasha S and Amutha S

Abstract- Last decade, a lot of research has been done in wireless communication technologies. Mobile nodes such personal digital assistants (PDAs), notebooks and cell phones are nowadays used in human's daily life.

MANETs are networks consisting of two or more mobile nodes equipped with wireless communication and networking capabilities, but they don't have any network centralized infrastructure. In last few years, MANETs have been emerged to be an important researched subject in the field of wireless networking.

MANETs are autonomous; however they can communicate with other external networks such the internet. They are linked to such external networks by mobile nodes acting as gateways. This kind of networks is known as hybrid MANETs. Voice over Internet Protocol (VoIP), is a technology that allows you to make voice calls using a Internet connection instead of a regular (or analog) phone line.

The goal of this paper is evaluate the performance of VoIP strategies for hybrid MANETs. Two different aspects are evaluated, the session establishment performance and the voice quality.

Network Simulator 2 is used to run several simulations, two different applications are used to run voice simulations (Session Initiation Protocol and Exponential traffic generator). We evaluate two different cases for voice traffics, voice calls between two MANET nodes and voice calls between MANET nodes and external nodes.

After running the simulations, there are some performance parameters which will reveal the results. The key findings of the simulations are: adding gateways, number of voice traffic flows and the number of hops between source and destinations. There are some interesting results which reveal for example, that adding gateways is not always beneficial.

Key words- MANET Mobile Adhoc Network, DVR Distance Vector Routing, SIP Session Initiation Protocol, TCP Transmission Control Protocol, NS Network Simulator.

I. INTRODUCTION

A wireless mobile ad hoc network (MANET) is a network consisting of two or more mobile nodes equipped with wireless communication and networking capabilities, but they don't have any network centralized infrastructure[5]. Each node acts both as a mobile host and a router, offering to forward traffic on behalf of other nodes within the network. For this traffic forwarding functionality, a routing protocol for Ad hoc networks is needed[6].

Ad hoc networks are self organizing, self healing, distributed networks formed by autonomous wireless nodes. They can communicate without any centralized control mechanism. They differ from traditional wireless networks. Traditional networks need access points and they have centralized organization, however, Ad Hoc Networks do not have centralized organization. Nowadays ad hoc networks are very popular subject of research; however they are originally created for military and emergency purposes[3].

Ad hoc networks are usually autonomous in connectivity between participating nodes, but they do not have connectivity to external networks such as the internet. Internet connectivity is an add-on to standard Ad Hoc Networks. Internet connectivity for MANETS can be possible, with multi-homed nodes (they can connect to the ad hoc network and more external networks)[4]. These nodes act as gateways between the ad hoc network and the external network. When there are nodes acting as gateways, it is necessary to have gateway discovery mechanism. When a MANET is connected to the internet or another wired network, it is called hybrid MANET[21].

Session initiation protocol (SIP) is an application-layer control protocol that can establish, modify and terminate multimedia sessions in an IP network. A session could be a simple two-way telephone call or a multimedia conference session. The goal of the dissertation is to analyze the performance of VoIP calls in hybrid MANETs[10]. When two SIP end users want to communicate in an ad hoc network, they need to send the SIP control messages through the gateways even if both wireless nodes belong to that network[13]. This paper evaluated these situations because of the need to study the capacity of the internet connected Ad Hoc Network scenario and the need to examine the performance of SIP call setup and voice call flow quality in order to design a scalable system[7].

The main aspects that would be evaluated are calls establishment and voice calls quality. The first task is defining all the scenarios (nodes and traffic) making a coherent study in order to be able to make conclusions. The second task is implementing and running the simulations for all the scenarios in Network Simulator 2[17]. The following task is extracting and organizing the information from the simulations. Finally, this paper evaluates the scenarios by comparing situations and scenario results.

Prakasha S, Dayananda Sagar College of Engineering, Bangalore,
E Mail: - sprakashjg@yahoo.co.in

Amutha S, Dayananda Sagar College of Engineering, Bangalore,
E mail:- amuthajj@yahoo.com

II. NETWORK SIMULATOR 2

2.1 Introduction

The Network Simulator 2 is an object-oriented, discrete, event-driven network simulator. It is implemented in OTcl and C++. Using two different programming languages is not so usual in applications, but it has some advantages. A major advantage of this feature is that it makes it possible for users to write the simulation scripts in Tcl (due to NS is an object-oriented Tcl script interpreter).

More complex functionalities based on C++ can be added by the user. This flexibility is very important to enhance the simulation environment that is needed. The most common components are built in the simulator, for example wired nodes, wireless nodes, links, agents and applications. The Network Simulator 2 also uses C++ for efficiency reasons. The NS libraries are written in C++ (Event Scheduler Objects, Network Component Objects, Network Setup Helping, Modules) [29]. Many network components can be configured in detail. Also

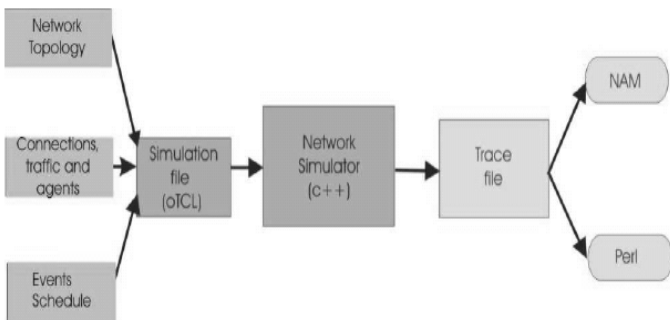


Fig. 3. User's view of a network simulation [23]

traffic patterns can be added to give more reality to the simulations. The Network Simulator 2 can be seen structured in two very different parts; the Tcl interpreter and the NS Simulator Library. Figure 3 shows the work flow in the simulation when the user uses the Network Simulator 2 to run simulations[17][18][23].

2.2 Running simulations

To run a simulation, a Tcl script has to be written (simulation scenario). A simulation scenario is composed of three main components:

- network topology
- connections, traffic and agents (protocols)
- events schedule

A network topology defines the scenario in terms of number of nodes and the connectivity between them. When the simulation script is built, usually it follows a certain pattern consisting of the following items:

- Simulation network instance

The simulation instance has to be defined in every simulation. The simulator object is generated and it is assigned to the variable ns[15]. See Code 1.

```
..Creating object simulator instance...
set ns [new Simulator]
```

Code 1: simulator instance

What this line does is that it initializes the packet format creates a schedule and selects the default address format [28].

- Opening trace files

The trace files and their formats that we want to receive as an output is defined here. It can be the normal NS trace file or the NAM trace file. See Code 2.

```
...
Opening the trace files...
set nf [open trfileName.tr w]
$ns trace-all $nf
```

Code 2: Opening trace files

- Configuration of nodes

The node settings are defined. The nodes will have all the settings defined when they are created. explains with examples how to configure wireless and wired nodes.

- Creation of nodes and links

The nodes are created; they will have the settings defined in above point. The links between the nodes will be established if the nodes are wired. If the nodes are wireless, the connection settings are already defined in above point "configuration nodes".

- Creation of agents and applications

At this point the agents (protocols) are attached to the nodes to create application events. Section explains with example how to create agents and applications.

- Events schedule

The time events are defined, when the events start and when they finish. shows how to implement it in Tcl scripts.

- Finish procedure and call to this procedure when the simulation is finished

In this procedure we have the required actions to finish the simulations. This procedure usually looks like as written in Code 1

```
...finish procedure...

proc finish {} {
    global ns_ tracefd
    $ns_ flush-trace
    #Close the trace file
    close $tracefd
    exit 0
}
```

Code 3: Finish procedure

This procedure is used to finish the trace event and close the trace files.

- Starting the simulation

Next command (Code 4: Starting simulation) is an essential Tcl command which determines when the simulation starts. It works by issuing the run command to the simulator instance.

```
..Starting the simulation...
$ns_run
```

Code 4: Starting simulation

2.3 Node structure

Nodes are fundamental in NS-2 and in the simulations. They are the most important entities in the simulations; they perform processing and forwarding of packets. In this section, two different objects are described, namely the internal structures of wireless and wired nodes.

2.3.1 Wired nodes

A wired node is an object composed by one entry and two classifiers - the address classifier and the port classifier. Figure 4 shows how the different parts are connected in a wired node in NS-2[14].

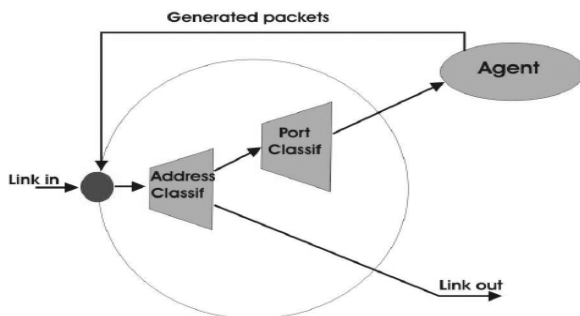


Fig. 4. Wired node structure in NS2

The node entry (Link In in Figure 4) receives the packets; the address classifier checks the packet address field to check if its node is the destination node. If the node is the destination packet node, the port classifier determines which agent will receive the packet. If the node is not the destination packet node, the address classifier will determine the node to which it has to be forwarded. The routing functionality updates the address classifier and the packets will be forwarded through the link out[9][20].

2.3.2 Wireless nodes

In this section, the wireless node structure in the Network Simulator 2 is described. Figure 5 shows the internal work flow of wireless nodes. The node always receives the packets at the node entry. The packet goes through the address classifier where its address field is examined. The agents are connectors. When the connectors receive packets, they execute some functions and deliver the packets to their neighbors or drop them. There are different types of connectors performing different functions, e.g., Agent, Link Layer, MAC Layer and Network Interface. The interaction between them is shown in Figure 5.

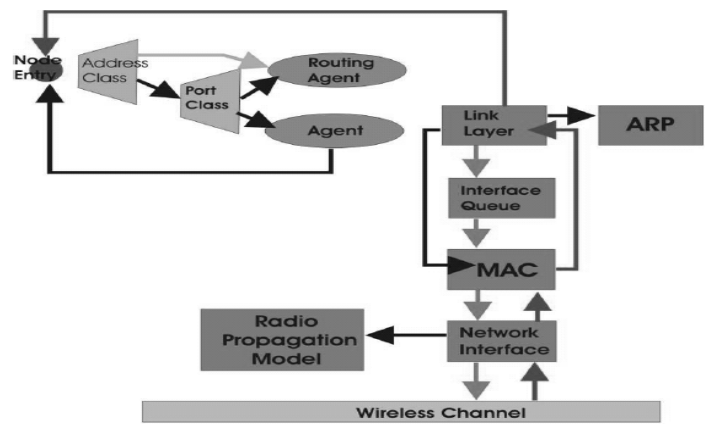


Fig. 5. Wireless node structure in NS2

2.4 Radio Propagation Models

In NS-2 there are three propagation models: free space model, two-ray ground reflection model and the shadowing model. Their implementations in NS-2 are done using the following files:

- for free space model: ~ns/propagation.{cc,h}
- for two-ray ground reflection model: ~ns/tworayground.{cc,h}
- for shadowing model: ~ns/shadowing.{cc,h}

Shadowing Model is discarded due to the simplicity of the simulation. In order to choose the optimum of the two remaining propagation models for our simulations, we need to calculate a cross-over distance. When the distance between transmitter and receiver, d , is less than the cross-over distance, d_c ($d < d_c$), the free space model is used. Otherwise when $d > d_c$ two ray ground model is used.

$$d_c = 4\pi \times ht \times hr / \lambda$$

The cross-over distance is calculated as

- Where:
- λ is the wavelength.
 - ht is the height of transmitter antenna
 - hr is the height of receiver antenna

2.5 Communication range calculation

Communication range settings are done with the code lines shown in Code 6. The Network Simulator 2 has a tool called "threshold". This tool is used to calculate these values (a and b in Code 6) which define the carrier sense and transmission range.

```

..Communication range settings...

Phy/WirelessPhy set CStresh_ 5.53241e-12 (a)
Phy/WirelessPhy set RXThresh_ 1.296e-10 (b)
    
```

Code 6: Communication range settings

The command below illustrates how the tool “threshold” is used: `threshold -m <propagation_model> [options] <distance>`

`-m <propagation model>` can be FreeSpace , TwoRayGround or Shadowing.

`<distance>` is the distance that it is needed to be calculated for carrier sense range or for transmission range in meters.

Next options may or may not be defined depending on the chosen propagation model:

```
-pt <transmit-power>
-fr <frequency>
-Gt < transmit antenna gain >
-Gr < receive antenna gain >
-L <system loss>
-ht < height of the transmitter antenna >
-hr < height of the receiver antenna >
```

2.6 Wired cum wireless simulation

The Network Simulator 2 can be used to run simulations modeling many scenarios. It can work with many applications, protocols and it can simulate wireless, wired and wireless-cum-wired scenarios. Wired cum wireless simulation scenarios will be explained in this section. So far, the configuration of the nodes was not so complicated, its complexity increases a little bit when we run a simulation with wireless and wired nodes simultaneously[22].

When a wired network is connected with a wireless network, base stations are needed to act as interfaces between the networks. Base stations are like gateways, they permit packets to be exchanged between different kinds of nodes[2].

Hierarchical addresses are needed in nodes to route packets between wireless and wired domains. It is not necessary to have different domains for wired and wireless nodes; the wireless nodes will be in the same domain as the gateway which will be the home gateway (the gateway which is the predefined gateway for them). If these mobile nodes change position and they go near other base stations, mobile IP will solve the addressing problem. The gateway has to be connected with the wired nodes as well. If the nodes have mobility in the simulation, we have to define one gateway as the home agent for the nodes which are predefined for this gateway. Foreign agents are gateways outside of the domain[1][8].

III. DESIGN AND IMPLEMENTATION

3.1 Introduction and Goals

The goal of our simulations is to observe the behavior and to determine the performance of the Hybrid MANET working in several different situations with VoIP. A MANET scenario is set up adding several different parameters which will create many different simulations[24]. The network is a hybrid MANET. It is a MANET that always has any gateway providing internet connectivity with an external network. The

number and positions of gateways are varied in different simulations. Some Exponential traffic flows will be created in the network modeling voice conversations. The number of traffic flows and the distance between sources and destinations will be varied during the simulations, many SIP call attempt will be initiated[25]

3.2 Implementation and configuration of VoIP in Hybrid MANETs in NS2

3.2.1 Network topology

The first step is creating the network simulator object. The simulator object is generated and it is assigned to the variable ns. See Code 7.

```
..Creating object simulator instance...
set ns [new Simulator]
```

Code 7: Creating NS object

After having created the simulator object, the topography and the topology are created. How to declare it is showed in Code 8.

```
..Creating topology and topography objects...

# Create topography object
1# set topo [new Topography]
# define topology
2# $topo load_flatgrid 2000 2000
# create God
3# create-god [100]
```

Code 8: Creating topology and topography

How network topology and topography are created in Code 8, is explained in the following lines:

- 1# In this line, a new topography instance called “topo” is created.

- 2# In this line, the new topology instance called “topo” is defined as a flat grid. Since these simulations are run in a grid of 10 x 10 wireless nodes separated by 200 meters, the flat grip is 2000 meters x 2000 meters.

- 3# This line creates a God instance. The number of wireless nodes is passed as an argument. It is used to create a matrix to store connectivity information of the topology.

3.2.2 Traffic and agents

Two different applications are used in the application layer. The simulations are basically implemented to Exponential traffic generator and SIP. The implementation of these two different applications in the Network Simulator 2 is explained in details below.

3.2.2.1 Exponential Traffic

Two voice standard codes are considered to be used in this paper: ITU2 G711 a-Law and G729. The family G 7xx is standard of audio protocols with varying specifications. G 711 and G 729 are standard voice codecs[33][34].

G711 generates 160 bytes of payload per 20ms without any compression and G729 generates 20 bytes of compressed audio data per 20ms. We can assume that VoIP conversations are established over real-time transport protocol (RTP), which uses UDP/IP between RTP and link layer protocols. The bandwidth and payload in each layer[20][9]. With header compression, RTP+UDP+IP header size can be reduced to 2-3 bytes, but this is not considered in our simulations.

G729 is the chosen standard for our simulations. For simulation purposes we are modelling bi-directional VoIP conversation as a Markov process with idle and burst time representing the voice activity in the conversation according to [28].

As stated, assuming each voice source as an on-off Markov process, the alternating active ('on') and silence ('off') periods are exponentially distributed with average durations of 350ms ('on') and 650ms ('off'). Therefore, we consider an average talk spurt of 35.0% and average silence period of 65.0% as recommended by the ITU-T3 specification for conversational speech[33]. We assume that the voice source generates Exponential traffic of 80 Kbps (at IP-layer) when 'on', and 0 Kbps when 'off'. [28]

To modulate phone calls between two end nodes, traffic flows with determined characteristics are created. In a first step, two UDP agents are attached to the source node and the destination nodes are defined. The identical process in the opposite way is done as well; due to a bidirectional traffic is needed to simulate the phone calls.

3.2.2.2 Session Initiation Protocol

Some entities and traffic agents are needed in the simulation code in order to run the simulation with SIP technology. The SIP proxy, SIP users and SIP agents are implemented.

The first and indispensable step is to create a SIP proxy server which is located in the central node in the wired network (W(0) node).

```

...defining SIP Proxy server...
#Proxy server
1# $W(0) label "proxy.atlanta.com"
2# set serveraddrATL [$W(0) node-addr]
3# set sipATL [new Agent/SIPProxy atlanta.com]
4# $W(0) attach $sipATL 5
    
```

Code 16: SIP proxy definition

As in the code above, a SIP Agent is created in the SIP proxy, which manages all the SIP messages. The SIP proxy server definition is explained below:

- 1# This line makes node W(0) call "proxy.atlanta.com"
- 2# This code line set the variable "serveraddrATL" to have node W(0)'s address.
- 3# This line creates a SIP proxy agent "atlanta.com".
- 4# Finally, in this line, SIP proxy agent is attached to node W(0) with port number 5.

As important as creating a SIP Proxy, is creating the SIP users which are the nodes that send and accept the SIP invitations.

3.2.3 Events schedule

After defining all the needed elements to run the simulations, the next step is to define the events schedule, for example, which kind of events are chosen to run and when.

For Exponential traffic flows an example is given in Code 19.

...Schedule events for the Exponential agents...

```

$ns_ at 5.0 "$e start"
$ns_ at 5.0 "$e1 start"

$ns_ at 185 "$e stop"
$ns_ at 185 "$e1 stop"
    
```

Code 19: Schedule for Exponential traffic

As was defined earlier, the Exponential agents are "e" and "e1". In the four lines in Code 19, at second 5

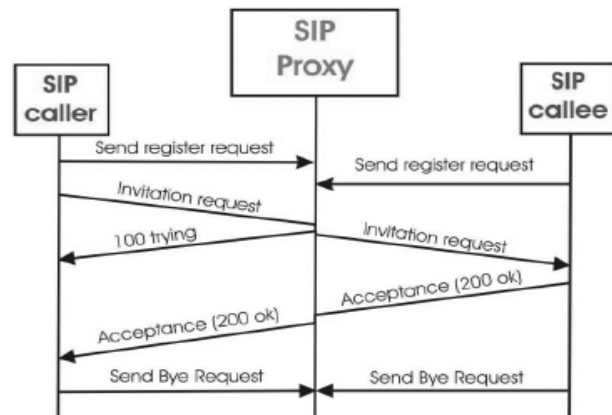


Fig. 8. SIP invitation event tasks

the Exponential traffic is started and stopped at second 185.

For SIP events to use this protocol, the event schedule is a little bit more complex due to the actions needed for having the invitation event[16]. Figure 8 shows the action needed by a SIP user in order to make an invitation call. As shown in Figure 8, SIP users have to register, and then the SIP caller user sends an invitation request. And finally, to end the call, both SIP users involved in the event send a bye request.

3.3 Post-tracing

The post-tracing function extracts all the desired information from the trace files. The possible trace formats have also been shown in the same chapter. It was also mentioned that the PERL language is used to obtain needed information from these trace files[19]. The performance parameters are then calculated using this information. How to obtain the required information is

described here. Three kinds of Perl scripts are created. They are used to:

- Obtain all the needed information such as Exponential packets delay, SIP calls setup delay, SIP hops and dropped packets with reasons.
- Calculate all the performance parameters with all the available information.

- Organize the performance data parameters in different files prepared to plot with GnuPlot program.

The information that is extracted from the trace files is:

- Exponential Packet Delay
- SIP call set up delay
- SIP number of hops
- Dropped packets and their dropping reasons.

With this information, all the performance parameters can be calculated. How the values are obtained is explained in the following sections.

3.4 Performance Parameters and Scalability

To understand the simulation results that we obtain is important. Varying the number of gateways, the number of hops and the number of voice traffic flows, there many different simulations.

The basic scenarios are 1 gateway scenario, 2 gateway scenario and 4 gateway scenario. Each scenario has 6 different numbers of hops between sources and destinations in simulations (2 hops, 3 hops, 4 hops, 5 hops, 6 hops and 7 hops).

And each one has 6 different number of background traffic flows (4 flows, 8 flows, 12 flows, 16 flows, 24 flows, 32 flows). Therefore the simulations are already 108 (3 x 6 x 6), and each simulation will be repeated 5 times with different SIP call participants. All these simulations generate a lot of information. However, this is too much information to evaluate manually how the network performance in the different situations is. Some important parameters (performance parameters) are defined to see the most important and relevant information about the network behavior. The parameters calculated are discussed in the following sections:

3.4.1 Background traffic

A number of performance parameters are obtained from the background traffic (exponential). These will be discussed in the following sections.

Average delay

The average delay is calculated by taking all the sum of all time differences between packets received and packets sent. The total time is divided by the number of packets that we measured the time for.

Maximum delay

This is the maximum delay time that an exponential packet takes when it is sent by the sender until it is received by the receiver. It is calculated for each simulation.

Minimum delay

This is the minimum time delay that an exponential packet takes when it is sent by the sender until it is received by the receiver. It is calculated for each simulation.

Packet loss rate

This is calculated as:

Sent packets and received packets are calculated per flow, and then, an average of all the flows in the same simulation is calculated.

3.4.2 SIP invitation calls

In this section, the performance parameters obtained from SIP flows are explained.

$$PacketLossRate = \frac{NumberOfPacketsSent - NumberOfPacketsReceived}{NumberOfPacketsSent}$$

Call set up Delay Average

The call set up delay is the time elapsed since the SIP caller node sends an invitation until the same node receives the acceptance message. The average is calculated with all the call setup times of the same simulation. For example, one simulation setup is 1 gateway, 2 hops, 4 flows, 5 repetitions in which there are 100 call invitations in total.

Minimum set up delay

This is the minimum value of all call setup delays taken from all SIP call invitations of each simulation.

Maximum set up delay

This is the maximum value of all call set up delays taken from all SIP call invitations of each simulation.

95 percentile of the call set up delays

It is calculated in the same way as is in the 95 percentile for background traffic, with all the SIP call times set up instead.

4.5.2.5 Block call probability

The call block probability is calculated as:

$$CallBlockP = \frac{TotalCalls - AcceptedCalls}{TotalCalls}$$

TotalCalls and AcceptedCalls are taken per simulation. There are 5 repetitions of each simulation. Each repetition consists of 20 calls, so these call block probabilities are calculated from the 5 repetitions, i.e. 100 SIP calls.

It has to be commented that the block call probability is divided into more than one parameter. Due to the mentioned recommended call setup times by ITU-F [29], call block probability can be defined as the calls which are out of different times, the first group indicate how many calls are established at any time, the second group indicates how many calls are established within the first two seconds and so on within five and ten seconds.

- Calls block probability for all the accepted calls.
- Calls block probability for calls accepted within 2 seconds.
- Calls block probability for calls accepted within 5 seconds.
- Calls block probability for calls accepted within 10 seconds.

There is an important reason for calculating the call block probability at any time. But when the call is not accepted it is due to a dropped message fact. When the call block probability at any time is high, the call block probability within a determinate time is already high

because there are a lot of calls which will not be accepted anymore.

SIP invitation attempts

SIP invitation attempts mean how many invitation messages the caller nodes send. There are two reasons of why one node sends more than one invitation message. The first is that the SIP caller user does not get the “100 trying” message. When the caller node sends the first invitation message, it waits 0.5 seconds for the “100 trying” message which comes from the SIP proxy. If “100 trying” message does not arrive, the node will resend the invitation, the second time, it will wait the same time plus the double of the time, so 1.5 seconds and so on.

The calculation of this performance parameter will be divided in two steps. In the first step the number of invitation messages in total the SIP caller users send is calculated and in the second step the number of resent invitation messages is calculated because of dropped messages and because of the delay reason.

3.4.3 Scalability of VoIP call setup and delay limits proposed

After analyzing the simulations we will have the results. As important as to have the results, it is to know the important information to be compared with other simulations. What are good results? How do we know if they are good? For these questions, there is not an official answer telling which the limits are.

After reading some papers, some proposals or recommendations can be used to have an idea about what are good or bad results. In this case, the ITU-F gives a recommendation called E.721; these recommendations (previously as CCITT recommendation) give different time values depending on some parameters. In Table 11 these recommendation time values are shown. In one hand, there are three different scalability networks and in the other hand the network can be considered as low

TABLE 11: E-721 ITU RECOMMENDATIONS [34]

Event	Network	Low load	High load
Delay (post-selection delay)	Local connection	3 sec	4,5 sec
	Toll connection	5 sec	7,5 sec
	International	8 sec	12 sec
Probability of end-to-end blocking	Local connection	2 %	3 %
	Toll connection	3 %	4,5 %
	International	5 %	7,5 %

load network or high load network.

There are three end-to-end network connections:

- Local connection 1-4 nodes.
- Toll connection 5-7 nodes.
- International connection 8-10 nodes.

Delay (post-selection) is defined as the interval time from the first bit of the initial setup message until the first message indicating disposition received by the calling party. Probability of end-to-end blocking is defined as the probability that any call is unsuccessful. Note that low and high network load may be geographically distributed and the international connection is assumed to be with one satellite link in the end user. [29]

IV. EVALUATION OF SIMULATIONS

4.1 Traffic Scenario

As stated earlier, in the application layer there are two types of events in the simulations. On one hand, different traffic flows modeling phone calls are running in a full time during all the simulations and these are called background traffic calls. On the other hand, a determinate number of SIP users try to establish SIP sessions at different times. Both cases are explained further below.

4.1.1 Background Traffic Scenario

In this scenario, background traffic is running during the simulations. Why is called background traffic? In these simulations, SIP invitation calls are the most important aspect. However in every simulation there are some exponential traffic flows modeling phone calls (VoIP traffic) called background traffic because it is running in background during the whole simulation.

For each simulation there are different numbers of background traffic flows representing different network load; these background traffic flows are differentiated depending on the number of hops between source and destination. The simulations are classified depending on the number of traffic flows. The traffic flows are always divided in two groups (75% of them are inside the MANET and 25% of them goes to outside the MANET). They are called inside and outgoing traffic flows respectively. One of the aims of our approach is that 25% of the exponential traffic flows are between nodes inside the network and nodes outside the network, and the 75% of exponential traffic flows are inside the network.

As defined in our approach, each simulation is repeated for the following values of number of hops in the background traffic: 2 hops, 3 hops, 4 hops, 5 hops, 6hops and 7 hops.

The number of hops is between the source node and destination node when the traffic is inside the Ad Hoc Network. However it is between the source node and the gateway when the traffic is between a wireless node and a wired node.

Note that the source and destination nodes in the simulation scenarios are defined with one gateway in terms of number of hops. And afterwards, two and four gateways simulations are added maintaining the same source and destinations background traffic flows. Therefore the number of hops between the source and the gateways can be different when there are more gateways.

4.1.2 SIP calls traffic scenario

The SIP call traffic scenario is the most important part of the simulations. Since the goal of the dissertation is to determine the behavior and the performance of VoIP in hybrid MANETS in different traffic scenarios, the SIP calls scenario will be critical in determining our results.

During the simulation we adopt the premise that a node can not participate in more than one voice conversation.

In order to have a good average, it was decided that having 100 SIP invitation calls is a good number to have accurate results in order to see and study SIP calls behavior. For this approach, 200 nodes are needed (100 caller parties and 100 called parties). Five groups of 40 nodes (20 callers and 20 callees) are defined randomly. The nodes are attached in appendix section B.1.3.

To make a more realistic situation, typically it is not common that 20 nodes in a 100 nodes network want to perform a SIP session simultaneously. Thus since the second 20 until the second 115, each 5 seconds, one of these caller nodes will start a SIP call attempt.

V. CONCLUSIONS AND FUTURE WORK

In this chapter, we present conclusions and suggestions for future work. The evaluation of all simulation results and the evaluation can be found.

5.1 Conclusions

In this paper, we have researched the performance of VoIP calls in hybrid MANETs. When Mobile Ad Hoc networks are connected to the internet, it is important that voice communication is supported. For this reason, the support of SIP protocol is required. However, if two partners in the MANET need to communicate, SIP messages need to be sent to the SIP proxy first in order to discover the location of the called party. In hybrid MANETs, that means that SIP messages need to pass the gateway twice, which limits the performance.

Also, the capacity of MANET in terms of voice calls is important, and it is important to mention the behavior of intra and outgoing traffic flows, where the intra traffic flows interfere in the network causing congestion and the outgoing traffic flows interfere mainly in the gateways.

Therefore, we have set up a set of simulation scenarios to evaluate performance of VoIP in hybrid MANET. We have evaluated the scalability with respect to the number of hops, number of voice traffic flows, and number of the gateways in the scenario. Our results in which we differentiate between difference between session call establishment aspects and voice quality aspects are show below.

Session call establishment

Since to completing a session call requires the messages between caller and called SIP users and the proxy go through the gateways, the evaluation of results indicate that adding gateways to the scenario, the call setup delay is smaller. Though voice traffic flows are not directly explained in here, the evaluations indicate that voice traffic flows interfere in the session call establishment performance parameters.

When there are more voice traffic flows, the call setup delay is higher. The same behavior is present when the number of hops between source and destination in these voice traffic flows is large. These behaviors are due to the congestion in the MANET caused by increasing the number of voice traffic flows and the number of hops. Implicitly, number of invitation attempts increases when the congestion is higher in the

network. The invitation attempts are caused due two reasons, the delay in the "100 trying" message or the dropped messages. Both reasons are directly caused by the congestion, so, the conclusion about session call establishment is that adding gateways creates a better performance but when the number of hops in the voice traffic flows and the number of traffic flows increase, the performance of session call establishment is worse.

Voice quality

Upon the evaluation of the voice quality performance, it was decided to differentiate between inside (when source and destination are inside the MANET) and outgoing (when one of the end parts is outside of the MANET) traffic flows, due to their different behaviors.

It is very important to mention regarding the outgoing traffic flows, adding more gateways decreases the number of hops when the gateways are more than three when the scenario has one gateway.

For inside traffic flows, the packet loss rate increase when the number of flows increases. However, adding more gateways makes packet loss rate higher. This is because around the gateways there is a congested zone due to the outgoing traffic flows and SIP messages which go through the gateways, therefore, to more congestion zones, more intra traffic flows badly influenced. Around these congested zones, the nodes suffer from congestion problems, so the packet loss rate is higher when there are more gateways[24][25].

For outgoing voice traffic flows the behavior is totally different, packet loss rate is lower due two reasons; the traffic that each gateway supports is lower and the distance in number of hops between nodes and gateways is usually smaller. The average end to end delay of the packets for intra flows has no significant improvement by adding gateways[29][30]. For outgoing flows, adding gateways improves significantly the end-to-end delay.

As a conclusion, it can be said that adding gateways create a congested zones but it gives a better performance in general.

5.2 Future work

In future work, next points explain briefly what we plan: The first suggestion is to introduce mobility into the scenarios. In this paper, all the simulations have been run in static scenarios with one, two and four gateways. It would be interesting to study the performance of the same parameters varying the mobility of normal nodes and nodes acting as gateways. With this approach, mobile IP or any mobility management protocol will be required.

The second suggestion tries to implement session initiation protocol (SIP) totally inside a MANET that means, the caller, the callee and the SIP proxies are nodes in a MANET. Several approaches can be applied: One suggestion is trying to make the gateways acting as SIP proxies, if there are gateways in the MANET.

The second suggestion does not need to have gateways in the MANET. The approach can be making the SIP callers act as SIP proxy as well. For that, one solution can be that

all the SIP user nodes send register event to all the MANET nodes.

The third suggestion tries to solve some problems encountered in this paper with the SIP protocol. As it was commented when call block probability was evaluated, SIP proxy has vulnerability when some packets are dropped.

When the “200 ok” message from the called node to the proxy is dropped, the call will never be established. The internal reason of this is because when the called node sends the invitation acceptance, this node goes into “busy” state, and when a node is in “busy” state, it can not listen one invitation. So if the acceptance message is dropped, the call will never be established.

There can be many ways to solve this problem; the next ones can be some suggestions: The node can send more acceptance messages after a while, until the call is established.

The called node can identify which the caller node is; therefore, if the caller node wants to send another invitation because it did not receive the acceptance, the called node can receive and consider another invitation in “busy” state if the invitation is from the same caller node which made being in “busy” state.

The fourth suggestion is use MAC protocol 802.11e with Quality of Service, where SIP and routing messages will have High Priority. The expected results is that the call establishment will improve while the voice quality will continue being poor.

VI. REFERENCES

- Periodicals
- [1] Nico Bayera, Bangnan Xua, Sven Hischkeb. An Architecture for connecting Ad hoc Networks with the IPv6 Backbone (6Bone) using a Wireless Gateway.
 - [2] Dragos Nicolas, Samrat Ganguly, Kyungtae Kim, Rauf Izmailov “Performance of VoIP in a 802.11 Wireless Mesh Network”
 - [3] Perkins C & Bhagwat, P. (1994) highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers.
 - [4] Johnson, D. & Maltz, D. & Broch, J. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks
 - [5] Pedro M. Ruiz, Antonio F. Gomez-Skarmeta. Adaptive Gateway Discovery Mechanisms to Enhance Internet Connectivity for Mobile Ad Hoc Networks.
 - [6] Alex & Ali Hamidian. A Study of Internet Connectivity for Mobile Ad Hoc Networks in NS 2.
 - [7] C.Siva Murthy and B. S. Manoj. Ad Hoc Wireless Networks Architectures and Protocols
 - [8] Urban Olofsson and Rickard Sjostr m. A prototype for Extending Global IP Connectivity for Ad Hoc Networks. October 2004.
 - [9] Alexander Zurkinden. Performance Evaluation of AODV Routing Protocol. June 2003.
 - [10] Radvision, the VOIP experts. Understanding SIP Servers. Radvision, August, 2002.
 - [11] Enrico & Minack “Evaluation of the influence of channel conditions on Car2X Communication” November 2005.
- Text Books
- [12] Abraham Silberschatz and Peter B. Galvin. Operating system Concepts. Addison Wesley, 4th edition, 1994.
 - [13] Alan B. Johnston. Sip: understanding the Session Initiation Protocol. Artech House Boston, second edition, 2004
 - [14] Ilyas, M. (2003). The Handbook of Ad Hoc Wireless Networks
 - [15] Larry Wall, Tom Christiansen & Randal L, Schwartz. Programming Perl. Second Edition, September 1996.
- [16] Dorgham Sisalem Jiri Kuthan. Galvin. Sip tutorial. Mobile Integrated Services.
 - [17] Marc Greis. Tutorial for the Network Simulator “ns”. VINT group
 - [18] Eitan, Altman and Tania Jiménez. Venezuela. NS Simulator for beginners, December, 2003
 - [19] Fredrik Thernelius. SIP, NAT, and Firewalls. May, 2000
- Tech Reports
- [20] Bjorn Wigerg. Porting AODV-UU Implementation to ns-2 and Enabling Trace-Based Simulation. Uppsala University, December 2002.
 - [21] Christer Ahlund. Extended Mobile IP and support for Global Connectivity in Hybrid Networks. Lulea university, May 2005.
 - [22] Alba Batlle Linares and Jonas Karlsson. Evaluation of TCP Performance in hybrid Mobile Ad Hoc Networks. Karlstad University, June, 2006.
 - [23] Erik Nordstrom, Per Gunningberg, Christian Tschudin. Comparison of forwarding Strategies in Internet Connected MANETS. Uppsala University.
 - [24] Tuulia Kullberg. Performance of the Ad Hoc On Demand Distance Vector Routing Protocol. Helsinki University.
 - [25] Jin Xi and Christian Bettstetter. Wireless Multihop Internet access: Gateway discovery, routing and addressing. Technische University München
 - [26] Jae Chung, Mark Claypool. Ns by example. Worcester Polytechnic Institute. <http://nile.wpi.edu/NS/>
 - [27] http://www.tkn.tu-berlin.de/research/802.11e_ns2/techreport.pdf.
 - [28] The Network Simulator. Universidad de la Republic. Uruguay
 - [29] A. Trivi o-Cabrera, E. Casilari-Pérez, A. Ariza-Quintana. Implementation protocolos en el Network Simulator. Universidad de Malaga
 - [30] Enrico Minack. Evaluation of the influence channel conditions on Car2X Communication. November, 2005. Chemnitz University of Technology.
 - [31] <http://www.grc.upv.es/manets/local/docs/GriswoldNS-2%20Traces.htm>
 - [32] Fall, K. Varadhan. The ns manual (formerly ns Notes and Documentation). <http://www.isi.edu/nsnam/ns/ns-documentation.html> April 2006.
 - [33] ITU-T Recommendation P.59, “Artificial conversational speech”, 1993
 - [34] ITU-T Recommendation Recommendation E.721, May, 1999.

VII. BIOGRAPHIES



Prakasha S born in Bangalore 01-1-1972, he did masters B.E from SJCE Mysore university, and M.Tech from MSRIT, VTU Belgaum, his area of interest are Computer networks, Mobile computing Wireless networks



Amutha.S born in Coimbatore she did masters from MGR university and working in Dayananada Sagar college of Engineering Ph.D. Her area of interest are Computer networks, mobile computing wireless networks