3.1 Introduction

All the simulations are carried out using network simulator (ns-2) [25]. In second step of simulations the traffic is with constant bit rate and variable bit rate. And our transports are User Datagram Protocol (UDP), Transmission Control Protocol (TCP), and Real-time Transport Protocol (RTP). And measured the through, packet loss, delay and jitter which are all Quality of services. Finally plotting our results and comparing them to our theory based predictions.

3.2 Simulation Scenario

To employ our previous QoS strategy as a comparative method, In this study:

- Using NS2 to implement our VoIP network.
- setting up several nodes on either side of two routers
- Use an exponential traffic source to re-create a typical voice conversation over VoIP.
- Different protocols will be used to send voice information between terminals, starting with Real-time Transport Protocol (RTP) for node 0 and 4,and User Datagram Protocol (UDP) for other nodes.
- Measuring throughput, end-to-end delay, packet loss and if the protocol permits and jitter.
- Plotting our results and comparing them to our theory based predictions.

3.3 Design Module

In a standard circuit-switching network, an analog voice signal must be sampled at twice its maximum frequency at 8 bits per sample. Standard human speech reaches about 4000 kHz, thus a bandwidth of 64kbps is required. The advancement of codec technology has improved bandwidth efficiency in telephony by only transmitting information when a person is talking [10]. Therefore, a variable bit rate on each end is required to accurately simulate a VoIP call.

3.3.1 Ns-2 Implementation

In implementation, assume that the commonly used G.711 codec, which transmits information at a rate of 64kbps [13]. The size of the transmitted packets was chosen to be 128 bytes for RTP. The phone call is established between Nodes 0 and 4 as shown in Figure (3-1). Node 0 transmits data with an average "on" time of 1200ms and idle time of 800ms. Node 4 is setup to transmit fewer packets over the 60 second simulation with an average "on" time of 800ms and idle time of 1200ms.

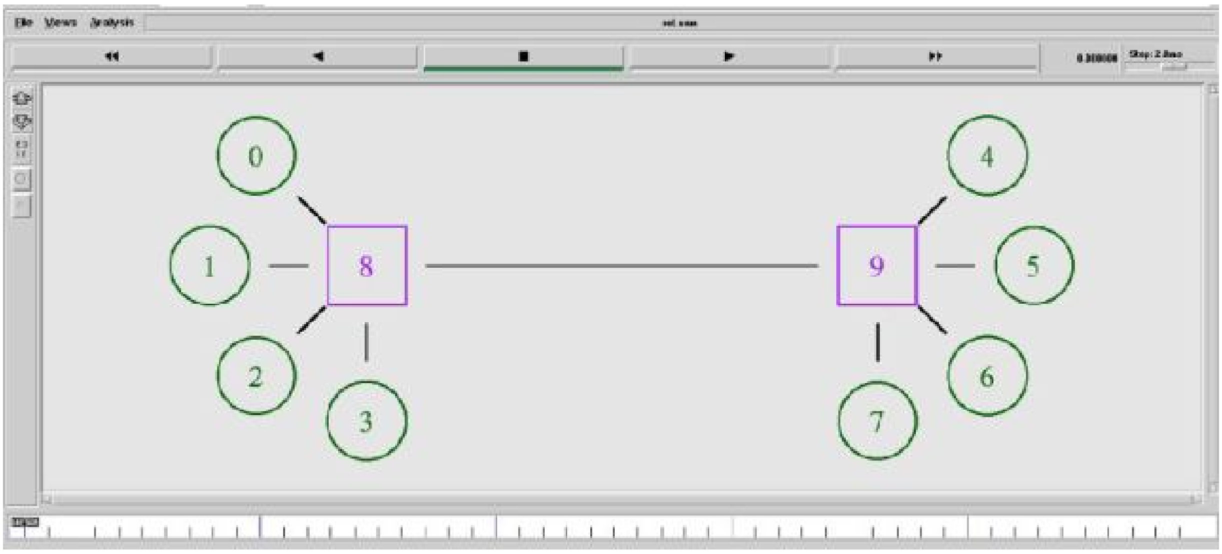

Figure (3-1): NS-2 implementation of a VoIP network

The background traffic of the network is supplied by Nodes 1, 2, 3, 5, 6, and 7 at constant bit rates. As the simulation begins, Nodes 1 and 5 create background traffic at a rate of 25.89Mbps, providing a sub-maximal load for the duplex link between the two routers. Then Nodes 2 and 6 are turned to provide background traffic of 25.91Mbps each. This was chosen to slightly overload the link's capacity and thus cause congestion within the network. Finally, Nodes 3 and 7 are tasked with providing the background traffic at a rate of 25.92Mbps each, greatly exceeding the network's bandwidth. At this point, it is expected that the queues become full and resulting in many dropped packets.

Depending on our simulation, UDP, TCP, or RTP agents were attached to Nodes 0 and 4. For the background traffic, used UDP for all three scenarios. Furthermore, Traffic from Node 0 to Node 4 is color coded as blue, whereas traffic from Node 4 to Node 0 is color coded as red. The simulation can be visualized in NAM.

3.3.2 Filtering The Trace File With AWK

The NS2 code opens an output trace file, *out.tr*, and records every single event of the simulation for each individual packet, such as entering queues or being dropped. An important step in identifying the data pertaining to the phone call itself is to assign unique flow IDs – packets transmitted from Node 0 have a flow ID of 1, while packets transmitted from Node 4 have a flow ID of 2.

The NS-2 trace file is structured in columns as follows:

- Event type

"+": packet enters queue

"-": packet leaves queue

"r": packet received

"d'' packet dropped

- Timestamp
- Source Node
- Destination Node
- Packet Type (ie: "exp" for the variable bit rate used to simulate the call)
- Packet Size
- Flags (unused)
- Flow ID
- Source Address
- Destination Address
- Sequence Number
- Unique Packet ID

Sample of trace file contents:

```
+ 6.642252 4 9 tcp 1040 ---A--- 2 4.0 0.1 3 420372
- 6.642252 4 9 tcp 1040 ---A--- 2 4.0 0.1 3 420372
+ 6.642252 0 8 tcp 1040 ---A--- 1 0.0 4.1 3 420373
- 6.642252 0 8 tcp 1040 ---A--- 1 0.0 4.1 3 420373
r 6.777252 4 9 tcp 1040 ---A--- 2 4.0 0.1 3 420372
+ 6.777252 9 8 tcp 1040 ---A--- 2 4.0 0.1 3 420372
r 6.777252 0 8 tcp 1040 ---A--- 1 0.0 4.1 3 420373
+ 6.777252 8 9 tcp 1040 ---A--- 1 0.0 4.1 3 420373
- 6.779185 8 9 tcp 1040 ---A--- 1 0.0 4.1 3 420373
- 6.779185 9 8 tcp 1040 ---A--- 2 4.0 0.1 3 420372
r 6.829506 8 9 tcp 1040 ---A--- 1 0.0 4.1 3 420373
+ 6.829506 9 4 tcp 1040 ---A--- 1 0.0 4.1 3 420373
- 6.829506 9 4 tcp 1040 ---A--- 1 0.0 4.1 3 420373
r 6.829506 9 8 tcp 1040 ---A--- 2 4.0 0.1 3 420372
+ 6.829506 8 0 tcp 1040 ---A--- 2 4.0 0.1 3 420372
- 6.829506 8 0 tcp 1040 ---A--- 2 4.0 0.1 3 420372
r 6.964506 9 4 tcp 1040 ---A--- 1 0.0 4.1 3 420373
+ 6.964506 4 9 ack 40 ------- 1 4.1 0.0 4 440766
- 6.964506 4 9 ack 40 ------- 1 4.1 0.0 4 440766
r 6.964506 8 0 tcp 1040 ---A--- 2 4.0 0.1 3 420372
+ 6.964506 0 8 ack 40 ------- 2 0.1 4.0 4 440767
- 6.964506 0 8 ack 40 ------- 2 0.1 4.0 4 440767
r 6.974506 4 9 ack 40 ------- 1 4.1 0.0 4 440766
+ 6.974506 9 8 ack 40 ------- 1 4.1 0.0 4 440766
d 6.974506 9 8 ack 40 ------- 1 4.1 0.0 4 440766
r 6.974506 0 8 ack 40 ------- 2 0.1 4.0 4 440767
+ 6.974506 8 9 ack 40 ------- 2 0.1 4.0 4 440767
```

Figure (3-2): NS-2 trace file

A 60 second simulation with such a large amount of background traffic could result in trace files that are hundreds of MBs large, making parsing impossible. only interested in the data pertaining to the VoIP session between Nodes 0 and 4, the trace file is filtered using an AWK command:

awk '$8==1 || $8==2' out.tr > out.txt

This command takes lines containing flow ID 1 or 2 and transfers them to a new file.

3.3.3 Parsing And Plotting With MATLAB

The filtered output file is then parsed with MATLAB, by saving the data in each column into different vectors. This data can then be used to calculate and plot throughput, packet loss, end-to-end delay, and jitter. The main challenge is the fact that these measurements must be done based on intervals of time; however, the trace file only tracks events the exact moment when they occur and so the data inside the file is not listed at equal time intervals. Therefore, all calculations done in MATLAB must be based on the vector indices of the parsed data. For example, wish to make measurements at 2 second intervals.

For the interval 2 to 4s, scan the timestamp vector, *time,* to find the first index *i1* where $time[i1] \geq 2$ and similarly find first index *i2*, where $time[i2] \geq 4$. Thus, indices *i1*and *i2* essentially correspond to the beginning and end of a time interval, respectively. These indices are then used as upper and lower limits when scanning all other vectors necessary to compute our desired quantities.

3.3.4 Simulation to measure QoS

NS-2 is the simulator tool[16] used for designing the network and deploying VOIP technology view in the figure (1-1).Attached with TCP, UDP and RTP to get result from Matlab.

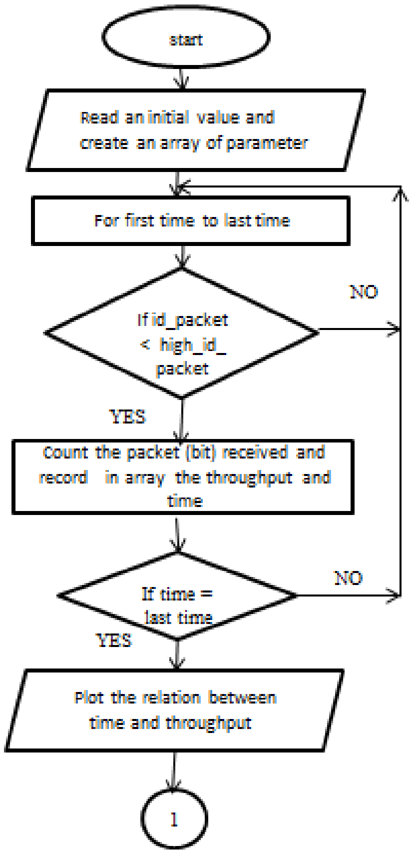

Figure (3-3-a):Simulation flowchart (Throughput)

From start the first event the program read the parameter from trace file and but it in mat-lab file (array form), trace file can filtered between end point event based on flow ID, and the filtered event saved in Colom vector this data can used to calculate and plot, to get the throughput for VoIP node filtered the received packet and it occur time using flow chart in figure (3-3-a) from start point to point (1) .

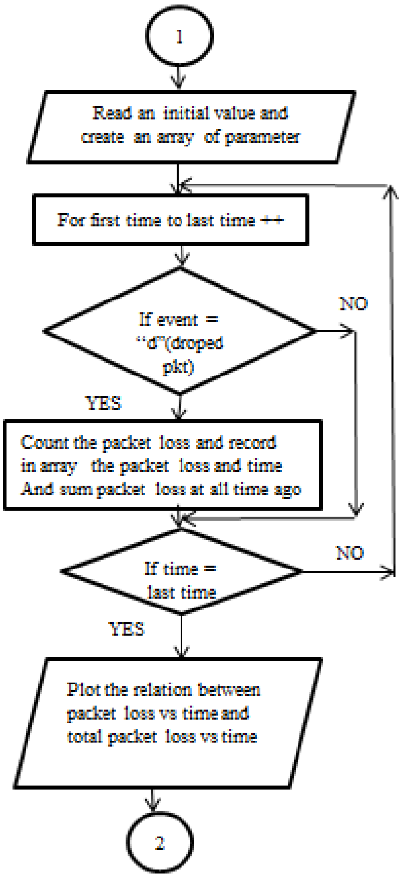

Figure (3-3-b):Simulation flowchart (Packet loss)

Packet loss filtered the drop packet and it event time using flow chart in figure (3-3-b) from point (1) to point (2).

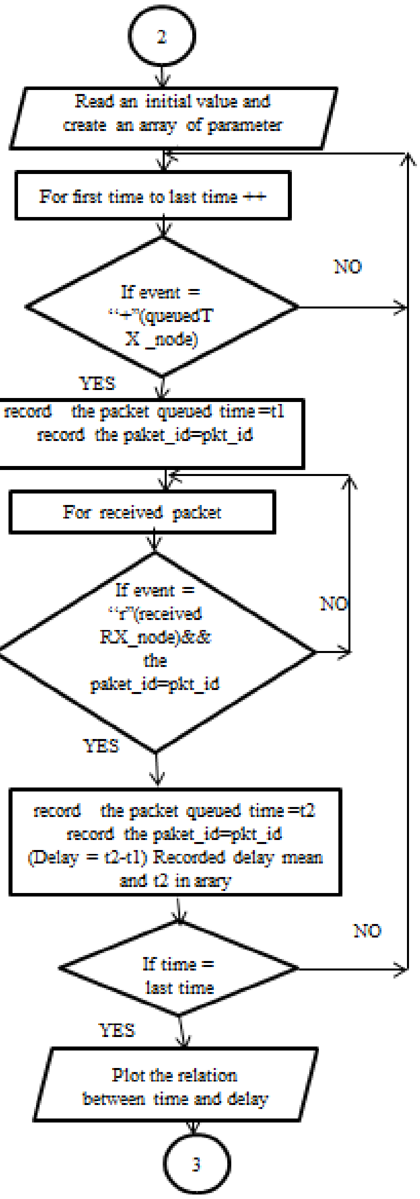

Figure (3-3-c):Simulation flowchart (delay)

the end to end delay between terminal using flow chart in figure (3-3-c)from point (2) to point (3) based on the time interval and the event of node 0 queuing packet and node 4 receiving packet

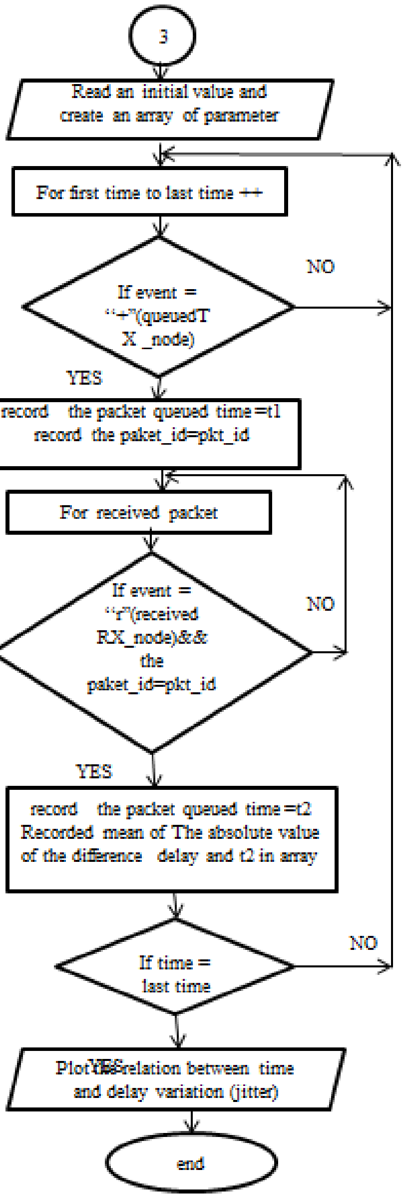

Figure (3-3-d):Simulation flowchart* (jitter*)

The figure (3-3-d) from point (3) to end point view delay variation or jitter depend on deferent between delay packet and time.

3.4 Summary

This chapter presented the methods and steps used to run the three protocol scenarios. Each scenario was tested. The simulation cases reflected VoIP traffic and parameters, and implements the simulation output in mat-lab to plot the relation between (QoS) parameters and time, to compare and evaluate the better protocol that it can discuss it in the next chapter.