Solving the Delay problem in Voice over Internet Protocol (VOIP) Over Wireless LAN.

Author’s Details:
(1)Ghazi.I. Raho  (2)Jazia Moftah Ahmad  (3)Alaa Hussein Al-Hamami  1,3 Amman Arab University 2 Tripoli University

Abstract
Voice over Internet Protocol (VoIP) technology has been rapidly growing up these days and is expected to be supported widely in wireless network. There are several issues in VoIP; the Quality of Service (QOS) is the most important one. There are many factors that affect QOS such as throughput, echo, delay, jitter, packet loss, bandwidth and delay variation. The effect of delay on speech quality in VOIP will be improved. The source of the delay is the buffer that already exists in the multiplexing node and also the process that can be done on the packets before sending them to the multicast node. This problem can be solved by using the proposed multiplexer –multicast algorithm that combine the data which come from several streams into a single large packet, where the packet that comes first is send over multiplexer for multicast. The result will be compared with original multiplexer-multicast algorithm that combine multiple channels into one channel, then multiplexer send all packets at the same time for multicast. We used OPNET IT Guru Academic Edition simulation for network performance analysis.

Keywords: Multicast; Multiplexer; Delay; QoS; and VoIP.

1. Introduction
Voice over Internet Protocol (VoIP) is a technology that allows making telephone call using Internet connection instead of analog phone line. VoIP allows calling others also receiving calls over the Internet. Some VoIP services require computer, while others allow using landing phone to place VoIP calls through special adapter [1]. VoIP involves digitization of voice as packet over IP network [2]. VoIP can be implemented by converting the analog voice call to digital, where packet generation of digital single according to the Transport Control Protocol(TCP), User Datagram Protocol (UDP) and Internet Protocol (IP), this packet reception and analog single reconstruction at destination. Voice signals will transmit through packet –switch instead of transmitted through circuit-switched [3]; using packet will reducing completely the expensive telecommunication company, so VoIP is gaining popularity because it provides a low cost of voice communication.

2. Multiplexing and Multicast Algorithm
Multiplexing is the process that combined multiple channels for transmission over a common transmission path [4]. Multiplexer makes it possible for several signals to share one device or resource. Where the advantages of multiplexes are simplifies operational complexity, increase management conversation visibility and Reduce high network costs. While main disadvantage of multiplexes is very expensive. Multicast is the delivery of information to a group of destination computers simultaneously in a single transmission from the source creating copies automatically to network elements [5]. Where the advantage of multicast are support distributed applications, scalability, reduces traffic and server loads by simultaneously delivering a single stream of information to thousands of users. But the disadvantages are the drops to be expected, no congestion avoidance, and duplicate of data. Delay in voice over internet protocol (VoIP) system is not constant, and varies depending on some technical factors. This variation in delay is called jitter, which causes damage to voice quality. Delay causes echo in VoIP calls. The acceptable delay value between 0 ms and 300 ms, where it comes from Variaty sources such:- Encoding Delay, Decoding Delay, Router (Queuing Delay), and Capacity [6].

3. The Proposed Algorithm
In the proposed algorithm, we will focus on reducing the delay as most as we can. The technique that will be used is sending the packet immediately after receiving it; that means the multiplexing algorithm will work as usual by collecting the packets received from many channels to be sent in one channel. The difference in the proposed algorithm, will not be buffering to store the packet in it, so any packet will be received, will be sent immediately unless the packet reach after the specific time or in different order. The changes will be in the multiplexing algorithm, not in multicast algorithm. The proposed algorithm is shown in Figure (1).

http://www.ijmsbr.com
Upon receipt of a packet
if received = expected
    send the received packet
set expected = (received + 1) mod 2^16
else
    calculate \( D = ( (\text{expected-received}) \mod 2^{16} ) - 2^{15} \)
if \( D > 0 \) then
    \{ packets expected, expected+1, ... received-1 are lost \}
        Send the received packet
set expected = (received + 1) mod 2^16
else \{ late packet arrived \}
discard packet

Figure (1) – The Proposed Algorithm.
For the multicast algorithm in the server and client side will remain as it is in the original algorithm as shown in Figure (2).

✓ The server side
- Open the connection // through the socket
- Chose the group that will receive the data
- Chose the type of socket // LAN, Internetwork, that mentioned before
- Chose the protocol // in our algorithm UDP will be used.
- Put the IP multicast range // such as 244.100.0.1 that mentioned before
- Put the port number // such as 9050 that mentioned before
    IPEndPoint = new IPEndPoint (range, port)
- Data=" this variable will include the data"
- Server.send (IPEndPoint, Data)
- Close server connection
✓ Client side
- Open connection // through socket
- Listen to the selected port // in this algorithm was 9050 that’s means the client will keep listening to his port.
- Listen to the selected IP address multicast // such as 224.100.0.1
- Receive the data.
- Close the connection // close the socket

For the multicast algorithm in the server side and also in the client side will remain as it’s, no change will be done on them as in Figure (3.4).

✓ The server side
Open the connection // through the socket

Chose the group that will receive the data

Chose the type of socket // LAN, Internetwork, that mentioned before

Figure (2) Multicast Algorithm

4. Comparison between old and proposed algorithm

If we pay attention to the differences between the two algorithms, we will notice that the processing cost in the old algorithm is more than the cost in the proposed technique. The old algorithm will be suitable for application that doesn’t care about a little latency in information receiving. It’s true that there is some latency in the old algorithm but the data that will be delivered to the user will be more correctable and reliable [7].

For the proposed algorithm, the latency will be reduced since the received packet will be transmitted directly to the next receiver. The limitation of the proposed algorithm is that some data will be lost according to the algorithm as we will clarify it in Table (1).

Table (1): A comparison between old and proposed algorithm

<table>
<thead>
<tr>
<th>Source</th>
<th>Old Algorithm</th>
<th>Proposed Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source of Latency</td>
<td>Existing in the following: If received = expected If not over-run then Place packet contents into play out buffer - There is a buffer to be checked for a space and the elements in the buffer will be counted. - Consuming Time. - Increase Latency, packet stay in buffer for a T time. - Lost packet will be indicated.</td>
<td>If received = expected Send the received packet - No process to keep packet in buffer. - There is no buffer.</td>
</tr>
<tr>
<td>Case of Late Packets</td>
<td>- Wasting a lot of time for processing, calculation and checking some condition. - The received packets in different order, their places will be marked in the buffer as a lost packet.</td>
<td>- There is no such calculation. - The late packet will be discarded. - Lost packet will be more.</td>
</tr>
</tbody>
</table>

5. Simulation Approach

The simulation approach uses the popular OPNET IT Guru Academic Edition simulation, Release: 9.1.A a vast amount of models of commercially available network elements have been exist in OPNET Modeler, and various real-life network configuration capabilities already exist in OPNET Modeler. This makes the simulation of real-life network environment close to reality.

The simulation model of the LAN network is an exact duplication of the real network. In OPNET, many vendor specific models are included in the pre-defined component libraries such as Cisco, 3com, and act. In the used network, the Ethernet routers are used to represent the routers.

The main parameter to be set in the router configuration is the forwarding rate. In the network, router has a forwarding rate of 25000 bps. The memory size needs to be increased to 32 MB to suit with the buffer that will be used to store the
received packet. By the way, the QoS will not be needed in the proposed technique since any packet will be received will be sent immediately.

VoIP Application and Profile Settings: One approach to model the VoIP traffic in OPNET is to use the predefined voice application. Essentially, any application that already predefined in OPNET is a collection of tasks of which each task is defined as a set of phases. In turn, each phase takes place between two endpoints and has a configurable traffic behavior. After defining and configuring VoIP application, it is required to configure the approach in which workstations will be implementing this application. Profile will be used to configure the behavior of network workstations. Profile is a collection of applications that need to be configured to control their start time and also the end times, in addition to the repeatedly of these applications.

6. Simulation Results

The duration of OPNET simulation has been configured to be 20 minutes. The number of events that have been done on the networks as shown in Figure (3) is 46975482; the average speed was 27531 events/sec. Where event anything can be sent or received the signal such multiplexer or multicast.

![Figure (3) - Numbers of events.](image)

The corresponding VoIP End-to-End delay Shawn in Figure (4), it’s appearing that the End to End delay in the proposed algorithm is less than the End to End delay in the original algorithm. After 20 minutes of simulation, the maximum point of End to End delay in the original algorithm is nearly to 1.65 second while the maximum point in the proposed algorithm is nearly to 0.5 second. Its look also that the End to End delay is increased while the numbers of calls have been generated. But there is a significant difference between the increased ratios for both algorithms. Furthermore, we can read from the Figure that in the proposed algorithm is that in a certain point of time, the End to End delay approximately stays a constant while the End to End delay in the original algorithm keep increasing.

![Figure (4) - End to End Delay](image)
Figure (4) shows the corresponding VoIP End-to-End delay. From the figure, it’s appearing that the packet Delay variation in the proposed algorithm is less than the packet Delay variation in the original algorithm. After 20 minutes of simulation, the maximum point of packet Delay variation in the original point is approximately to 0.35 second while the maximum point in the proposed algorithm is nearly to 0.02 second (which considers nothing). Its look also that the packet Delay variation is increased while the numbers of calls have been generated until a certain point then the packet Delay variation start decreasing. But there is a significant difference between the increased ratios for both algorithms. Furthermore, we can read from the figure that in the proposed algorithm is that in a certain point of time, the packet Delay variation approximately stays a constant weather packet Delay variation in the original algorithm keep increasing until a certain point then start decreasing.

Figure (5) shows the corresponding VoIP packet Delay variation. From the figure, it’s appearing that the packet Delay variation in the proposed algorithm is less than the packet Delay variation in the original algorithm. After 20 minutes of simulation, the maximum point of packet Delay variation in the original point is approximately to 0.35 second while the maximum point in the proposed algorithm is nearly to 0.02 second (which considers nothing). Its look also that the packet Delay variation is increased while the numbers of calls have been generated until a certain point then the packet Delay variation start decreasing. But there is a significant difference between the increased ratios for both algorithms. Furthermore, we can read from the figure that in the proposed algorithm is that in a certain point of time, the packet Delay variation approximately stays a constant weather packet Delay variation in the original algorithm keep increasing until a certain point then start decreasing.

Figure (5) - Corresponding VoIP packet Delay variation.

Figure (6) shows also, the Data Dropped in the network, it’s appearing that the Data Dropped in the proposed algorithm is more than the Data Dropped in the original algorithm as we expected it. After 20 minutes of simulation, the maximum point of Data Dropped in the original point is nearly to 9500 bits/sec while the maximum point in the proposed algorithm is nearly to 11250 bits/sec. Its look also that the Data Dropped is increased while the numbers of calls have been generated in both of the algorithm.

Figure (6) - Data Dropped in the network.

7. Conclusions

The main conclusions of this work can be summarized as follows:

- When compare proposed multiplexer-multicast algorithm with the original one, the following conclusion can be drawn:
  - The End to End delay in the proposed algorithm is less than the End to End delay in the original algorithm.
The packet delay variation in the proposed algorithm is less than the packet delay variation in the original algorithm.

The data dropped in the proposed algorithm is more than in the old algorithm.

The big challenge in multicasting is to minimize the amount of network resources utilized to compute and setup multicast network. But there is a drawback in the proposed algorithm, which is the number of packets dropped in the proposed algorithm, is more than the number of packets will be dropped in the original algorithm.

References