Analysis of Voice Captured Packet using Wireshark

Nazmia Kurniawati, Suhartati Agoes
Electrical Engineering, Industrial Technology Faculty, Trisakti University
Jalan Kyai Tapa No. 1, Grogol, Jakarta Barat 11440, Indonesia
E-mail: nazmia.kurniawati@trisakti.ac.id

ABSTRACT

Accuracy in data interpretation plays an important part in experiment result analysis. In this paper, a Wireshark captured packet file obtained from a voice call made through Voice over Internet Protocol (VoIP) system is analyzed. From the data, Wireshark added some bytes as additional information containing network adapter, IP, and protocol information. From the analysis, it is concluded that the actual transmitted or received data can only be seen from the packet’s detail information section.

Keywords: Wireshark, Data Analysis, Packet Length

1. Introduction

Wireshark is a software that is used to capture the packets in a network and to present information about those packets in as much detail as possible[1]. The information displayed by Wireshark is captured time, source and destination IPs, communication protocol, packet length, and additional info. Besides those, Wireshark also provides statistics about delay, packet drop, and jitter. Based on those statistics, the network’s performance can be analyzed. The information can also be used to troubleshoot network, investigate security problems, testify network application, and...
debug protocol implementation. The captured information can be taken from various network adapter types, such as Ethernet, Wireless Local Area Network, Bluetooth, USB, etc.

Inside the captured packets, there is much information stored. So when analyzing the data, the desired information must be filtered. The error in data interpretation will lead to incorrect analysis and wrong conclusion. Therefore it is important to know how to read the data correctly.

In this paper, a captured packets data file from Wireshark is analyzed. The data is taken from a voice call experiment using Voice over Internet Protocol (VoIP) system with Global System for Mobile Communication (GSM) audio codec. Since RTP carries audio codec for communication, we applied RTP filtering to analyze the data. The data show that the average transmission time for each GSM packet is 20 ms. From that experiment, the measured transmission time has the same duration as the standard value defined at ETSI EN 300 961 V8.1.1 [2]. Each packet should contain 260 bits or 33 bytes because, based on standard, GSM carries 13 kilobits per second. But when analysing the data length, each RTP packet has length 87 bytes. The measured packet length has different amount with the defined standard. An analysis is conducted to find the possible cause for this different value between the measurement result and the defined standard.

The organization of this paper is as follows. Section 1 gives a brief introduction to what this paper will discuss. Section 2 describes the literature study related to the problem. Section 3 explains the research methodology. Section 4 exhibits the findings and results. Section 5 concludes the discussion.

2. Literature Study

2.1 Voice over Internet Protocol (VoIP)

Voice over Internet Protocol (VoIP) is a technology for delivering voice signals over an internet protocol network [3]. By using this technology, a conventional network delivering voice information only is no longer needed. So the needed network
to be built is only the data network. This could also lower the cost needed for long-distance communication.

One of the protocols used in VoIP technology is Session Initiation Protocol (SIP). This protocol is used to establish the connection, control the signaling during the call, and terminate the connection [4]. This protocol gives an INVITE message when the connection is establishing. An ACK message indicates that the connection has been successfully made. Then a BYE message is sent when the connection is terminated.

There are several important Key Performance Indicators (KPI) used in VoIP, such as delay, jitter, packet loss, and Mean Opinion Score (MOS). Delay represents the time difference between the sent time and the received time. Jitter shows delay variation. Packet loss shows how many loss packets compared to the sent packets in percentage. And MOS represents the perceived quality of communication. Those KPI can be taken from statistics provided by Wireshark.

2.2 Real Time Protocol (RTP)

Real Time Protocol (RTP) is a set of protocols to enable real-time packets delivery over Internet Protocol networks, such as voice, video, or other multimedia service [4]. The standard for RTP is defined at IETF Proposed Standard RFC 3550. Some KPI’s that can be measured by RTP are packet loss, delay, asymmetric routing, and out of sequence packet arrival. RTP is a part of the User Datagram Protocol.

After the connection has been established in SIP, the multimedia session is handled by RTP. During the session, codec selection is essential because it determines the data rate. The higher the codec rate, the higher the data rate. Therefore codec selection plays an important part in RTP.

2.3 GSM Audio Codec

An audio codec is a computer program that has the ability to encode and decode audio signals. During the process of encoding, audio information is captured in the
highest possible sampling rate, then is compressed but still maintains the desired quality [5].

GSM supports various voice codecs, such as Regular Pulse Excited-Linear Predictive Coder (RPE-LPC), half-rate, full-rate, and Adaptive Modulation Rate (AMR). Based on ETSI EN 300 961 V8.1.1, GSM audio codec used for the digital telecommunication system carries 13 kilobits per second with transmission time per packet is 20 mili second [2]. Therefore in one second, there are 50 packets containing 260 bits or 33 bytes.

3. Research Methodology

In this paper, a Wireshark captured file obtained from a voice call made through VoIP system is analyzed. The developed VoIP system consists of a Kamailio SIP Server Software installed inside the server computer. The computer used as the server is a Raspberry Pi 3b with a Linux operation system. Kamailio SIP server software provides numbers and passwords for user identity. One smartphone and one laptop are used as clients and are installed SIP client software. In that client software, users must enter the given number and password as defined at Kamailio to be able to connect to each other. Wireshark that is used to capture the network packet during the experiment is installed on the laptop. The system used in the experiment is based on the VoIP system built at [6]. The communication between server-client and client-client is done through the wireless network. Figure 1 shows devices connection while Table 1 gives IP information for the server and clients.

![Figure 1 Devices Connection](image-url)
Figure 2 shows the whole experiment process. The first step is configuring Kamailio SIP Server software installed on Raspberry Pi. In this step, numbers and passwords that will be used by users are defined. The audio codec for users is set at GSM. After that, SIP Client software is configured. The defined numbers and passwords are set inside the client software. Audio codec setting at client software is also set at GSM. After server and clients’ software have been set up, the next step is to do the test call to ensure that every device, server, and clients, are connected and communication can be successfully made. If the test call failed, then troubleshooting is required. The work can proceed to the next step if only the test calls can already be successfully made.

The next step is preparing Wireshark to capture the packet in the network during the call. After Wireshark is ready, click CAPTURE to record all activities in the network. Then do a test call for two minutes [7]. After two minutes, disconnect the call and click STOP at Wireshark. Test calls can be done several times for a more accurate result. The captured file will be saved in .pcapng format. After the captured file is saved, then it is analyzed. Finally, a conclusion is taken from the analysis result.

In this paper, the discussion will be focused on analyzing the packet’s length from the captured file and a conclusion is made based on that. In the analysis process, the experiment result will be compared with standard defined by ETSI EN 300 961 V8.1.1. The analysis is focused on two aspects: packets’ transmission time and length.
4. Finding and Result

Inside Wireshark captured file, there is much information as Wireshark captures every packet in the network. Therefore, filtering is needed to avoid incorrect data interpretation due to massive information existence.
Based on data, SIP handles communication only during connection is establishing and terminating. When a connection has been successfully established, it is RTP’s responsibility to handle the session. Therefore we applied RTP filtering to the data to separate RTP packets from packets obtained from other protocols. Since audio codec has a fixed bit rate, RTP packets also have a fixed length. As it has been mentioned earlier, GSM audio codec with transmission time 20 ms and bit rate 13 kbps is used in the experiment. Therefore theoretically each packet’s transmission time is 20 ms and length is 260 bits or around 33 bytes.

Figure 3 displays a part of the experiment result obtained by Wireshark. The result shows the time of measurement, source and destination IPs, the protocol used by the packet, packet’s length, and information about the packet. Time of measurement is indicated in a minute while packet length is in byte.

![Figure 3 RTP Captured Data](image)

**Table 2 Measurement Result for Time Difference of Transmission Time**

| Time (in min) | Source    | Destination | Protocol | Length (in Byte) | Time Difference (in ms) |
|---------------|-----------|-------------|----------|-----------------|------------------------|
| 84.28271073   | 192.168.1.6 | 192.168.1.12 | RTP      | 87              | 20.808494              |
| 84.30351922   | 192.168.1.6 | 192.168.1.12 | RTP      | 87              | 18.486304              |
| 84.32200552   | 192.168.1.6 | 192.168.1.12 | RTP      | 87              | 20.790963              |
| 84.34279649   | 192.168.1.6 | 192.168.1.12 | RTP      | 87              | 19.837796              |
| 84.36263428   | 192.168.1.6 | 192.168.1.12 | RTP      | 87              | 17.521053              |
| 84.38015534   | 192.168.1.6 | 192.168.1.12 | RTP      | 87              | 19.777379              |
To calculate the transmission time, firstly the file is exported into the .csv file and is opened using Microsoft Excel. After that the data is grouped based on source IP, then the time difference of the measured packets is calculated. Table 2 shows the measurement result for the time difference of transmission time. From the table, it can be seen that each RTP packet from one client is sent about every 20 ms. Therefore it can be concluded that the measured transmission time has the same duration as the defined standard.

For packet length, the ETSI standard stated that each GSM audio codec packet is 33 bytes long, for a total transmitted bit each second is 13 kilos. As can be seen from Fig 3, each RTP packet length is 87 bytes. So it can be concluded that the measured packet length isn’t the same as the standard defined. We look at each packet’s detail to find what probably causes the difference.

By default, Wireshark displays each packet information in three sections. The first section shows captured packets sequence containing the measured time, source and destination IP, protocol, packet’s length, and information. The second section shows the packet’s detail information such as the number of captured bits, interface, IP version, source and destination port, and information about the protocol. The last section shows the packet’s byte information. If the byte has hovered, a description will appear informing what the byte indicates. Table 3 summarizes the information provided by the Wireshark. From the table, it can be seen that the RTP packet contains detail information about four aspects: network adapter, IP, UDP, and RTP. From byte 0 until 13 contains ethernet information, where byte 0-2 and byte 6-8 shows IG bit and byte 0-5 and byte 6-11 shows ethernet type. Byte 14 until 29 gives IP information where byte 14 shows IP header length, byte 15 shows IP explicit congestion notification, byte 16-17 shows IP total length, byte 18-19 shows IP identification, byte 20-21 shows IP fragment offset, byte 22 shows IP time to live (TTL), byte 23 shows IP protocol, byte 24-25 shows IP header checksum, byte 26-29 shows IP source, and byte 30-33 shows IP destination. Byte 34-35 gives information about UDP source port, byte 36-37 shows UDP destination port, byte 38-39 displays UDP length, and byte 40-41 displays UDP checksum. Byte 42 shows RTP contributing source identifier count,
byte 43 shows RTP payload type, byte 44-45 shows RTP sequence number, byte 46-49 displays RTP timestamp, byte 50-53 shows RTP synchronization source identifier, and byte 54-86 contains RTP payload.

From the table, it can be seen the part of the RTP packet that has the same byte amount as GSM audio codec standard, which is 33 bytes long, is rtp.payload which located at byte 54 until 86. Compared to the other part, the RTP payload has the most bytes while other parts are only 1 until 6 bytes long. From the information summarized by the table, we can make an assumption that GSM audio codec information is actually stored only at RTP payload. While other parts are added by Wireshark as detail information to inform where the packet comes from.

For comparison with the RTP packet measurement, the ping command is sent from one computer to another computer then captured by Wireshark. Each ping
command is 32 bytes long, as can be seen from Figure 4. From Figure 5 it can be seen that Wireshark measured the captured packet is 74 bytes long.

![Figure 4 Windows Ping Data Capture](image1)

![Figure 5 Wireshark Ping Data Capture](image2)

Then we analyzed the packet’s detail information and found that the part of the packet which has the same amount as the sent data is located at byte 42-73. While other parts of the packets show information about ethernet, IP, and protocol just like what is found at RTP packet information. Detail information about the ICMP packet can be seen in Table 4.
Table 4  ICMP Packet Structure

| Location Byte | Information                                      | Total Byte |
|---------------|--------------------------------------------------|------------|
| 0-2           | IG bit : individual address (unicast) (eth.ig)   | 3          |
| 0-5           | Source or Destination Hardware Address (eth.addr)| 6          |
| 6-8           | IG Bit (eth.ig)                                  | 3          |
| 6-11          | Source or Destination Hardware Address (eth.addr)| 6          |
| 12-13         | Type (eth.type)                                  | 2          |
| 14            | Header Length (ip.hdr_len)                       | 1          |
| 15            | Explicit Congestion Notification (ip.dsfield.dcn) | 1          |
| 16-17         | Total Length (ip.len)                            | 2          |
| 18-19         | Identification (ip.id)                           | 2          |
| 20-21         | Fragment Offset (ip.frag_offset)                 | 2          |
| 22            | Time To Live (ip.ttl)                            | 1          |
| 23            | Protocol (ip.proto)                              | 1          |
| 24-25         | Header Checksum (ip.checksum)                    | 2          |
| 26-29         | Source (ip.src)                                  | 4          |
| 30-33         | Destination (ip.dst)                             | 4          |
| 34            | Type (icmp.type)                                 | 1          |
| 35            | Code (icmp.code)                                 | 1          |
| 36-37         | Checksum (icmp.checksum)                         | 2          |
| 38-39         | Identifier (LE) (icmp.identifier)                 | 2          |
| 40-41         | Sequence Number (LE) (icmp.seq_le)               | 2          |
| 42-73         | Data (data.data)                                 | 32         |

5. Conclusion

In this paper, a file containing the Wireshark’s voice captured packet is analyzed. By default, Wireshark displays the captured packet’s information in three sections. In captured packet sequence section, each GSM audio codec packet has 87 bytes length whereas each packet should contain 33 bytes only. From the analysis, the length difference is caused by additional information provided by Wireshark. Those information including network adapter used by the device to receive or send the packets, source and destination IP address, and protocol information. The actual transmitted or received data, excluding additional information provided by Wireshark, is located at the end of the packet. At that location, the amount of measured data is the same as the defined standard. Therefore in order to know how long is the transmitted or received packet can only be seen from the packet byte’s information section.
References

[1] G. Combs, “Chapter 1. Introduction.” [Online]. Available: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html. [Accessed: 19-Aug-2019].

[2] L. Mobile, “Etsi en 300 341-1,” vol. 1, no. Rp 02, pp. 1–69, 2000.

[3] I.-H. M. and E. I. L.Sun, E.Jammeh, Guide to Voice and Video over IP. London: Springer-Verlag, 2013.

[4] A. B. Jhonston, Understanding the Session Initiation Protocol. Norwood: Artech House Inc., 2004.

[5] R. Soriano, “GSM Audio Codec: Vocoder: AMR, CELP etc .. » Electronics Notes.” [Online]. Available: https://www.electronics-notes.com/articles/connectivity/2g-gsm/audio-codecs-vocoders-amr-celp.php. [Accessed: 09-Oct-2019].

[6] N. Kurniawati, A. Affandi, I. Pratomo, and K. Gyoda, “Raspberry Pi-Based VoIP System for Rural Area,” in Proceeding - 2018 International Seminar on Intelligent Technology and Its Application, ISITIA 2018, 2018, pp. 33–38, doi: 10.1109/ISITIA.2018.8711237.

[7] Menteri Komunikasi dan Informatika, “1565237401-PM_14_2011_Standar_kualitas_jasa_internet_teleponi.pdf.” Menteri Komunikasi dan Informatika, Jakarta, 2011.