Lab 1 - Part I

Due: Monday Oct 21, 2019 (23:59:59 PM)

1. Goal

In this lab, we are going to use Wireshark to investigate the 802.11 wireless network protocol and closely observe how web latency is created by doing a cross layer analysis. We are going to see how WLAN layer interacts with TCP layer and also implement a full-stack processing to track web latency.

2. Wireshark Introduction

We will be using the Wireshark packet sniffer [http://www.wireshark.org/] for this lab, allowing us to display the contents of messages being sent/received from/by protocols at different levels of the protocol stack. (Technically speaking, Wireshark is a packet analyzer that uses a packet capture library in your computer). Wireshark is a free network protocol analyzer that runs on Windows, Linux/Unix, and Mac computers. It operates in computers using Ethernet, serial (PPP and SLIP), 802.11 wireless LANs, and many other link-layer technologies (if the OS on which it's running allows Wireshark to do so).

See more information on the following websites:

- a user-guide (http://www.wireshark.org/docs/wsug_html_chunked/)
- man pages (http://www.wireshark.org/docs/man-pages/)
- a detailed FAQ (http://www.wireshark.org/faq.html)

3. Instructions

3.1. Investigate the 802.11 wireless network protocol

For convenience, we'll provide ***Wireshark_802_11.pcap***, a trace of captured 802.11 frames, for you to analyze and assume in the questions below. This trace was collected using AirPcap and Wireshark, consisting of a Linksys 802.11g combined access point/router, with two wired PCs and one wireless host PC attached to the access point/router. Fortunately, other access points in neighboring houses are available as well. In this trace file, we'll see frames captured on channel 6. Since the host and AP that we are interested in are not the only devices using channel 6, we'll see a lot of frames that we're not interested in for this lab, such as beacon frames advertised by a neighbor's AP also operating on channel 6. The wireless host activities taken in the trace file are:

The host is already associated with the *30 Munroe St* AP when the trace begins.

- At *t = 24.82*, the host makes an HTTP request to http://gaia.cs.umass.edu/wireshark-labs/alice.txt. The IP address of gaia.cs.umass.edu is 128.119.245.12.
- At *t = 32.82*, the host makes an HTTP request to http://www.cs.umass.edu, whose IP address is 128.119.240.19.
- At *t = 49.58*, the host disconnects from the *30 Munroe St* AP and attempts to connect to the *linksys_ses_24086*. This is not an open access point, and so the host is eventually unable to connect to

this AP.

- At *t = 63.0*, the host gives up trying to associate with the *linksys_ses_24086* AP, and associates again with the 30 Munroe St access point.

1) Beacon Frames

Recall that beacon frames are used by an 802.11 AP to advertise its existence. To answer some of the questions below, you'll want to look at the details of the "IEEE 802.11" frame and subfields in the middle Wireshark window. Whenever possible, when answering a question below, you should hand in a printout of the packet(s) within the trace that you used to answer the question asked.
Annotate the printout to explain your answer. To print a packet, use *File->Print*, choose *Selected packet only*, choose *Packet summary line*, and select the minimum amount of packet detail that you need to answer the question.

1. What are the SSIDs of the two access points that are issuing most of the beacon frames in this trace?

2. What are the intervals of time between the transmissions of the beacon frames the *linksys_ses_24086* access point? From the *30 Munroe St.* access point? (Hint: this interval of time is contained in the beacon frame itself).

3. What (in hexadecimal notation) is the source MAC address on the beacon frame from *30 Munroe St.*? Recall that the source, destination, and BSS are three addresses used in an 802.11 frame. For a detailed discussion of the 802.11 frame structure, see the IEEE 802.11 standards document.

4. What (in hexadecimal notation) is the destination MAC address on the beacon frame from *30 Munroe St.*?

5. What (in hexadecimal notation) is the MAC BSS id on the beacon frame from *30 Munroe St.*?

6. The beacon frames from the *30 Munroe St.* access point advertise that the access point can support four data rates and eight additional "extended supported rates." What are these rates?

2) Data Transfer

Since the trace starts with the host already associated with the AP, let first look at data transfer over an 802.11 association before looking at AP association/disassociation. Recall that in this trace, at *t = 24.82*, the host makes an HTTP request to http://gaia.cs.umass.edu/wireshark-labs/alice.txt. The IP address of gaia.cs.umass.edu is 128.119.245.12. Then, at *t=32.82*, the host makes an HTTP request to http://www.cs.umass.edu.

7. Find the 802.11 frame containing the SYN TCP segment for this first TCP session (that downloads alice.txt). What are three MAC address fields in the 802.11 frame? Which MAC address in this frame corresponds to the wireless host (give the hexadecimal representation of the MAC address for the host)? To the access point? To the first-hop router? What is the IP address of the wireless host sending this TCP segment? What is the destination IP address? Does this destination IP address correspond to the host, access point, first-hop router, or some other network-attached device? Explain.

8. Find the 802.11 frame containing the SYNACK segment for this TCP session. What are three MAC address fields in the 802.11 frame? Which MAC address in this frame corresponds to the host? To the access point? To the first-hop router? Does the sender MAC address in the frame correspond to the IP address of the device that sent the TCP segment encapsulated within this datagram?

3) Association/Disassociation

Recall that a host must first associate with an access point before sending data. Association in 802.11 is performed using the ASSOCIATE REQUEST frame (sent from host to AP, with a frame type 0 and subtype 0) and the ASSOCIATE RESPONSE frame (sent by the AP to a host with a frame type 0 and subtype of 1, in response to a received ASSOCIATE REQUEST). For a detailed explanation of each field in the 802.11 frame, see page 34 (Section 7) of the 802.11 spec at http://gaia.cs.umass.edu/wireshark-labs/802.11-1999.pdf.

9. What two actions are taken (i.e., frames are sent) by the host in the trace just after *t = 49*, to end the association with the *30 Munroe St* AP that was initially in place when trace collection began? (Hint: one is an IP-layer action, and one is an 802.11-layer action). Looking at the 802.11 specification, is there another frame that you might have expected to see, but don't see here?

10. Examine the trace file and look for AUTHENICATION frames sent from the host to an AP and vice versa. How many AUTHENTICATION messages are sent from the wireless host to the *linksys_ses_24086* AP (which has a MAC address of Cisco_Li_f5:ba:bb) starting at around *t = 49*?

11. Does the host want the authentication to require a key or be open?

12. Do you see a reply AUTHENTICATION from the *linksys_ses_24086* AP in the trace?

13. Now let's consider what happens as the host gives up trying to associate with the *linksys_ses_24086* AP and now tries to associate with the *30 Munroe St* AP. Look for AUTHENICATION frames sent from the host to and AP and vice versa. At what times are there an AUTHENTICATION frame from the host to the *30 Munroe St.* AP, and when is there a reply AUTHENTICATION sent from that AP to the host in reply? (Note that you can use the filter expression "wlan.fc.subtype == 11 and wlan.fc.type == 0 and wlan.addr == IntelCor_d1:b6:4f" to display only the AUTHENTICATION frames in this trace for this wireless host.)

14. An ASSOCIATE REQUEST from host to AP, and a corresponding ASSOCIATE RESPONSE frame from AP to host are used for the host to associated with an AP. At what time is there an ASSOCIATE REQUEST from host to the *30 Munroe St* AP? When is the corresponding ASSOCIATE REPLY sent? (Note that you can use the filter expression "wlan.fc.subtype < 2 and wlan.fc.type == 0 and wlan.addr == IntelCor_d1:b6:4f" to display only the ASSOCIATE REQUEST and ASSOCIATE RESPONSE frames for this trace.)

15. What transmission rates is the host willing to use? The AP? To answer this question, you will need to look into the parameters fields of the 802.11 wireless LAN management frame.

4) Other Frame types

Our trace contains a number of PROBE REQUEST and PROBE RESPONSE frames.
16. What are the sender, receiver and BSS ID MAC addresses in these frames? What is the purpose of these two types of frames? (To answer this last question, you'll need to dig into the online references cited earlier in this lab).

3.2. A WLAN/TCP cross layer analysis

For convenience, we'll provide captured traces for you to analyze. We used iperf [https://iperf.fr] to run a TCP connection. We set up a server whose IP address is 222.29.98.58 running **iperf server** (by running

`iperf3 –s`). We captured a trace ***iperf_tcp.pcap*** of above-IP network data on a client device running **iperf client** to send uplink data and a trace ***iperf_wlan.pcap*** of 802.11 frames on a co-located device. The client running the **iperf client** has the MAC address b8:8a:60:f6:79:1c. Analyze the traces and do the tasks below:

1) Calculate the 3 way TCP handshake duration on both devices. Are they the same? Explain the reason.
2) Calculate the mean and standard deviation of the time to retransmit bad packets in ***iperf_wlan.pcap***. Does the retransmission affect the TCP? Why? (Hint: check FCS status to find bad packets)
3) Calculate the mean and standard deviation values of the TCP out-of-order delay, retransmission rate and RTT observed from the two devices. Describe the difference and explain the reason in detail.
4) Plot the time series of throughput, RTT, and retransmission rate observed on both devices. Analyze how the path loss in L1/L2 affects TCP performance (e.g. throughput and RTT).
5) Other insights you get from the traces.

3.3. Web latency breakdown

For convenience, we'll provide captured traces for you to analyze. We ran Chrome devtools on an Android phone to capture 10 websites selected from Alexa top-50 sites and saved them as **.har** files. The websites we chose are: *login.tmall.com; Sohu.com; 360.cn; Qq.com; Weibo.com; Csdn.net; Bing.com; Office.com; Xinhuanet.com; Apple.com*. Chrome browser offers Devtools to analyze web latency. Please refer to the demo at https://developers.google.com/web/tools/chrome-devtools/network/. We also ran tcpdump to capture the packet-level information on the cell phone. This part requires you to analyze web latency observed in the above collection. Do the tasks below:

1) Implement a processing program to calculate the latency of DNS Lookup, Initial TCP Connection Establishment (i.e. TCP Handshake Time), Request Sending, TTFB (Time to First Byte), Content Downloading Time. Show the results for each website (by charts or figures).
2) Choose 3+ page loading procedures (from different websites) and draw visual waterfall graphs to show the breakdown of web latency.
3) You are encouraged to collect extra dataset considering the mobility (e.g. roaming caused by walking), collecting the 80211 header (note that some devices may not support), or choosing other applications (e.g. videos, instant messages).

4. Project Submission

1. Put all your files into a directory, named "lab1_Name_ID" where ID is your student number.
2. The directory should include the following files:
   a) All the source codes for analysis.
   b) A report of your lab (pdf or word format).
   c) A readme that describes how to run your codes and explains the output files
   d) If you collect your own data, please include the data logs and explain how you collected them and also add the results required into the report.
3. Submit the zipped file to jing.wang@pku.edu.cn.

Reference:

Wireshark Lab: Getting Started v6.0, © 2005-21012, J.F Kurose and K.W. Ross
Wireshark Lab: 802.11 v6.0 © 2005-21012, J.F Kurose and K.W. Ross