Dashboard
Home
Profiles
Grey 360
Plans
Assignments
Discussions
Grades
Search
SELF
Assignment Help
Course Readings

Wireshark TCP lab quiz

Due: 24 Jan at 17:00 | Points: 12 | Questions: 12
Available: 10 Nov at 20:00 - 24 Jan at 17:00 | Time Limit: None | Allowed attempts: 1

Take this quiz again

Attempt history

	Attempt	Time	Score
LATEST	Attempt 1	30.044 minutes	12 out of 12

✓ Quiz successfully submitted. Congratulations! You scored 12 out of 12.

Score for this attempt: 12 out of 12
Submitted 10 Nov at 20:30
This attempt took 30.044 minutes.

Test 1 of 3
Question 1 (1 / 1 pts)
In 1988 VJ Jacobson & M. Karels - 10 Nights Research Project

In this lab, we'll investigate the behavior of TCP in more TCP lab in the exciting context of the TCP congestion control algorithm. We'll use the Wireshark packet sniffer to examine the set of Cisco Cisco's classic introductory network course. Wireshark implements the set of TCP and UDP protocols, TCP congestion control and congestion avoidance, TCP slow start, TCP congestion avoidance, TCP fast retransmit, TCP fast recovery, TCP SACK, TCP Reno and TCP Vegas congestion control algorithms.

Before beginning lab, you'll probably want to review sections 3.7 and 3.7 in the text.

1. Capturing a bulk TCP transfer from your computer to a remote server

Before beginning our exploration of TCP, we'll need to use Wireshark to obtain a packet trace of a TCP session between your computer and a remote server. We'll do this by uploading a 150 page long text file (which you have the rights to do) from your computer (which contains the text file of Alice's Adventures in Wonderland) and then transfer the file back to your computer. We'll analyze the TCP segments sent and received for this file, using the TCP sniffer Wireshark and plot the TCP sequence and ack numbers as a function of time from your computer to another computer. Of course, we'll be viewing the TCP segments as they are sent and received from your computer, and not as they are sent from other computers.

Do the following:

- Go to the Ethereal Ethereal Ethereal Home Page (http://www.ethereal.com/download.html) and download and install the WinPcap version of Ethereal (Wireshark). Since this file contains packets for your computer.
- Run an FTP client (http://www.cuteftp.com/download/cuteftp/70/setup/cuteftp70x0.exe) and
- Now, consult your network administrator.
- Now, consult your network administrator (www.ethereal.com/download.html) and
- Now, consult your network administrator (www.ethereal.com/download.html) and

Make sure you can identify these three types of events before proceeding. If your capture is large, it may be helpful to filter the packets by Destination Port = 21 (FTP control) or Source Port = 21 (FTP control). Also, be sure to set the PORT to that you can focus on the packets sent to/from ftp.

Note that IP address and port numbers of both the source and destination of the PORT request and the response.

Fill in the items below that have the same value

source port of PORT request	destination port of packet =
source port of acknowledgment (ACKs)	destination port of PORT ACK =
source IP of PORT request	destination IP of connection =
destination IP of PORT request	source IP of acknowledgment =

The IP address is a TCP segment identifies the host the segment is from/ to. The port number identifies the TCP application (process) on the host. A particular process on the host the segment is from/to receives packets in its destination port.

As an analogy, consider you are sending a letter. The address is to from: Jim Kurose, 140 Hicks Building, Amherst, MA 01003 and to from: Mrs. Ethel Jones, 24 Hicks Hall. The address has to/from fields both at both Hicks Hall and Amherst. Jim Kurose's office is in Hicks Hall, and Mrs. Ethel Jones is in Hicks Hall. The port number is like the Hicks Hall and Amherst.

2. A first look at the captured trace

Before starting the analysis of the TCP connection in detail, let's take a high level view of the trace.

- First, filter the packets displayed in the Wireshark window by entering "ftp" (lowercase, no quotes, and then hitting return) in the "Filter:" field near the top of the Wireshark window.
- Next you should see a series of TCP and FTP message between your computer and the remote server. Select any FTP packet in the top window pane. There are three main sequence of events you should see:
  - You should see the initial three way handshake containing a SYN message followed by a SYN ACK and an ACK. Then you should see a packet containing a PORT command. This packet contains the IP address and port number that the remote server should use to initiate a TCP connection back to the sender (FTP Wireshark ftp). Next is such thing as an FTP PORT command. Note that the PORT command is sent from the FTP client on the FTP control TCP port, and is sent to a single FTP message.
  - You should also see TCP ACK segments being sent back and forth between your computer and the remote server. Finally, a connection for FTP data between your computer and that have been received.
  - Finally, you'll see the FTP 150 OK being returned from the server in response to the PORT command.

Make sure you can identify these three types of events before proceeding. If your capture is large, it may be helpful to filter the packets by Destination Port = 21 (FTP control) or Source Port = 21 (FTP control). Also, be sure to set the PORT to that you can focus on the packets sent to/from ftp.

Note that IP address and port numbers of both the source and destination of the PORT request and the response.

Fill in the items below that have the same value

source port of PORT request	destination port of packet =
source port of acknowledgment (ACKs)	destination port of PORT ACK =
source IP of PORT request	destination IP of connection =
destination IP of PORT request	source IP of acknowledgment =

The IP address is a TCP segment identifies the host the segment is from/ to. The port number identifies the TCP application (process) on the host. A particular process on the host the segment is from/to receives packets in its destination port.

As an analogy, consider you are sending a letter. The address is to from: Jim Kurose, 140 Hicks Building, Amherst, MA 01003 and to from: Mrs. Ethel Jones, 24 Hicks Hall. The address has to/from fields both at both Hicks Hall and Amherst. Jim Kurose's office is in Hicks Hall, and Mrs. Ethel Jones is in Hicks Hall. The port number is like the Hicks Hall and Amherst.

Question 2 (1 / 1 pts)
Test 1 of 3

Since the ftp is about TCP rather than UDP, let's change Wireshark's "listing of captured packets" window, so that it shows information about the TCP segments containing the FTP messages, rather than about the FTP messages. To have Wireshark do this, select "Preferences" from the "Edit" menu, then "TCP" from the "Protocols" box and select OK. You should now see an Ethereal window that looks like:

This is what we're looking for - a series of TCP segments sent between your computer and your's computer. We will use the packets that you have captured to study TCP behavior in the rest of the lab.

3. TCP connection establishment

Examine what is printed in the first TCP segments sent. Compare the header information in the first three TCP segments sent between the two hosts. Note which TCP header and flags are used, and the order in which TCP and the segments sent.

By default, wireshark will show only relative sequence numbers (a connection starts with sequence number 0). Wireshark can also show the actual sequence numbers contained in the TCP segments, you need to choose preferences from the wireshark menu and choose TCP under protocols. Make sure "Relative sequence numbers" is unchecked.

If you are using actual Ethernet Ethernet, read http://ethereal.ethereal.org/TCP_Checksum_Offloading.html

Examine the TCP header fields. Note which code, are actually used. Compare the first three packets sent connection, and the second that packets from the client and to receive what the new connection is established.

Note that the TCP header contains information about connection start data. Looking at the connection establishment segments. What control bits in the header allows the server to know whether to deliver the segment to the SynAckSocket the one socket waiting for a connection request, or to the socket created for communication once the connection is established?

SYN

A control connection is from the destination port is different. So it accepting the connection request and other segments from the client, you can see that the client and server are using the same port segments from the client and on a server port. Note that the source port segments from the client and on a server port. Note that the source port segments from the client and on a server port.

A second connection (connection) is filtering the sequence number info (this is a connection) is filtering the sequence number info (this is a connection) is filtering the sequence number info (this is a connection) is filtering the sequence number info (this is a connection) is filtering the

Question 3 (1 / 1 pts)
Test 1 of 3

TCP sequence numbers are critical to reliable protocols that can reorder segments to allow the receiver to identify whether a received segment is a duplicate that has been seen in another or is a new segment. They are also needed for re-ordering segments that are received out of order. Sequence numbers allow the receiver network layer data not guaranteed in order delivered.

Examine the sequence numbers used by the client and server in the connection establishment segments. Fill in the following:

What are the following statements are true?

Select all that apply:

- neither of the other statements are true
- TCP sequence numbers are not sent in the order sent/received. The client SYN ACK and SYN ACK sequence number are sent/received