Department of Electrical and Information Technology
ETSF10 – Internet Protocols
Home / Laboratory Project II

TCP

The aim of this lab is to let you familiarise yourselves with some of the basic TCP concepts you encounter during the lectures. For this purpose, you are going to analyse the packet capture file of a client-server communication, i.e. communication between two end points in a real IP network such as the Internet. The data you're going to analyse is stored in advance to PCAP files, i.e. they are the outcome of a real communication session but they are not 'live'. What you need to do is to perform the tasks defined later in this document, analyse and interpret your observations as requested, and write a report summarising your findings.

The deadline for the lab report is December 3, 2010, 17:00.

Have fun!

1. Lab Setup

For the analysis, you need a software called 'packet analyser'. There are a number of packet analysers you can actually use for this lab; and you're free to pick the one you like most. Your lecturers, on the other hand, will only assume Wireshark knowledge when they answer possible questions from you.

You will have the opportunity to complete the lab on your personal computers, if you prefer, or use the department's computer laboratory facilities as usual. In the latter case, you can use one of the common computer laboratories in E-huset's basement (varg, val, falk, hacke, panter, lo, venus, mars and jupiter). Please keep in mind that, in the laboratories, you will only be able to work with Wireshark on Linux machines.

It is recommended to familiarise yourselves with the tool you'll work with before getting started. In case you opt for Wireshark but haven't used it before, download (http://www.wireshark.org/download.html) and install it, or go to one of the laboratories in the basement. Play along with the user preferences, capture some network traffic – only if you are working from home, as the department's network security policy won't let you do this in the laboratories, download, open and investigate some previously captured files (http://wiki.wireshark.org/SampleCaptures), etc. Also, check the Wireshark user documentation (http://www.wireshark.org/docs, esp. sections 1.2, 2.5, 2.8, 3, 4.3, 4.5, 4.10, 4.11, 5.2, 5.7, and 6).

2. Groups

You will work in groups of two. One partner from each group needs to register the group by sending an e-mail to Kaan Bür {kaan.bur@eit.lth.se} with the subject "ETSF10 - tcp lab group", and putting the other partner to CC. If you can't find a partner, send an e-mail to Kaan with the subject "ETSF10 - tcp lab group - help"; and he'll be your matchmaker. Nevertheless, we'd really prefer you to do this yourselves, so you can look for a partner in our course pages on the department's moodle system (http://moodle.eit.lth.se) as well.

3. Timeline

This year's lab is scheduled to take place between November 10, 2010, and December 3, 2010. (Always check the course's Gantt chart for possible changes.) You are free to decide when to start with the lab and how much time to spend on it; but we strongly recommend you to not postpone it until the last minute. Drop your report into the course's mailbox in E-huset, 3rd floor, staircase 'A Norr'.

4. Report

The report must be of reasonable length written in your own words; and you need to decide for yourselves what the reasonable length is. (Tip: Your answers shall be clear, complete and self-contained. In other words, you shall not assume that the lecturers already know the answers or what's going on your minds.) You shall describe your approach, i.e. how you found what you found, as well as your findings during the tasks. Whenever necessary, include in the report the relevant information from the PCAP file, e.g. printouts from Wireshark showing the packet header(s) or the data where you got your answer from. Don't forget to mark the data with e.g. a highlighter pen to make it visible. A chapter with your own reflections and comments on the lab assignment concludes the report. Apart from these requirements, the report is free-style, i.e. you are going to have to decide for yourselves the best way to present your results.

Please include as an appendix to your report your personal comments and suggestions on the lab as well as any errors found in this document or any bugs regarding the lab in general. We would also like to know the amount of time you spent with (a) your analysis and (b) the report.

5. Evaluation

Your lab report will be assessed and graded as 'passed' or 'failed'. A grading bonus will be applied to outstanding work. Check the course's homepage for details.

6. Help

Help will be available throughout the project timeline. The preferred method for asking questions is to use the students' forum in our course pages on the department's moodle system (http://moodle.eit.lth.se) so your fellow students can also benefit from an answer you get for your question. If, for some reason, this doesn't seem to work, send an e-mail to Kaan {kaan.bur@eit.lth.se}. Finally, check the office hours on the course's home page and drop in during these hours.

7. Further Reading

http://tools.ietf.org/html/rfc793 on TCP
http://tools.ietf.org/html/rfc2581 on TCP congestion control
http://en.wikipedia.org/wiki/Pcap on packet capture
http://en.wikipedia.org/wiki/Packet_analyzer on packet analysers
http://en.wikipedia.org/wiki/Comparison_of_packet_analyzers on a list of available tools
http://www.wireshark.org on our favorite tool

8. Tasks

To complete the following tasks, download the necessary PCAP file from the TCP Lab section of the course's project page (http://www.eit.lth.se/index.php?id=241&ciuid=393&coursepage=1665&L=1).

Then, run the packet analyser of your choice (e.g. Wireshark) and open the PCAP file you downloaded. To answer the questions, it is sufficient to look into the listing of the captured packets (in Wireshark, the colourful lines usually in the upper half of the application window) and the details of the selected packet header (the middle part in Wireshark), unless specified otherwise. You won't need to go into the details of packet contents.

When answering the questions, imagine that we are looking at the message exchange log of a file transfer. The node requesting the file is the client, whereas the one sending it is the server.

8.1. Connection establishment

Investigate the packets exchanged for connection establishment and find the answers to the following questions. (Refer to § 23.3, pp. 723-725, in Forouzan's book if you need to refresh your memory.)

8.1.1. Who's the client, who's the server? Identify with IP addresses.

8.1.2. How many packets are exchanged to establish the connection? Which ones? Give index numbers. How do you identify these packets? What do we also call the connection establishment process?

8.1.3. For each packet you identify as a connection establishment packet, identify also the TCP header fields. Prepare a table representing the TCP header and fill it in with actual data you read from each packet. In your report, use tables like the one shown in Fig. 23.16, p. 721, in Forouzan's book. Use decimal numbers instead of binary except for the 1-bit flags where you only can enter 1s and 0s. You do not need to fill in the "checksum", "urgent pointer", and "options and padding" fields.

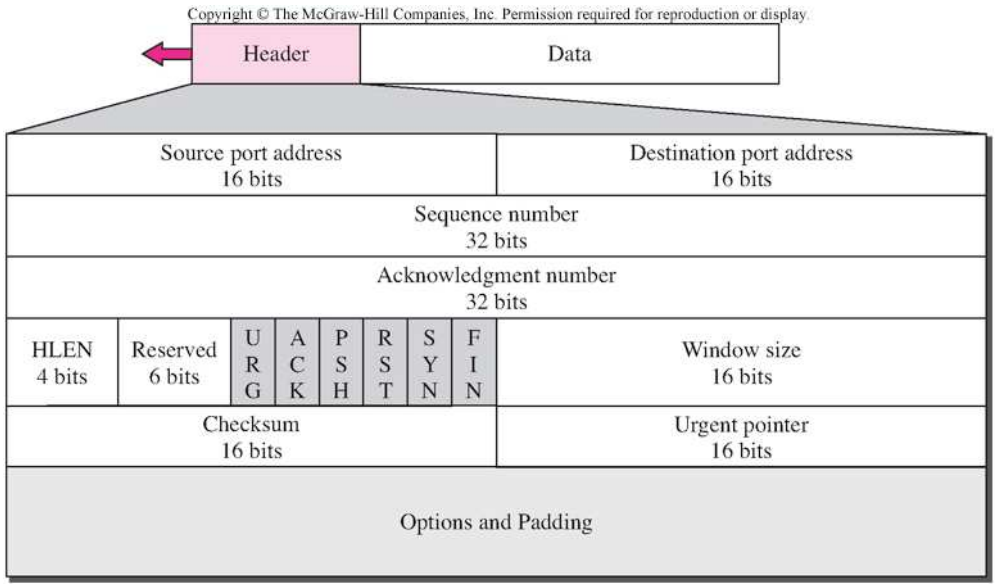

Figure 23.16: *TCP segment format* – Note that we are interested in the header only!

8.1.4. At the end of connection establishment, what are the values for the maximum segments size (MSS) and receiver-advertised window (RWND) agreed upon, both for the client as well as for the server?

8.2. Data transfer

Once a connection is established, the actual data transfer can begin. So, now, investigate this second phase in our TCP session and answer the following questions. (Refer to § 23.3, pp. 725-727 and 728-735, in Forouzan's book if you need to refresh your memory.)

8.2.1. By now, you should know about the essential data networking concepts such as the OSI layered network architecture and the corresponding TCP/IP suite, also shown in Fig. 23.8, p. 709, in Forouzan's book. So, you know that TCP is activated, at least in our lab, by an application layer protocol. Which one is it? Where do you read that information from?

8.2.2. The application layer protocol on the client side uses a method to request a video from the server. What is this method called? What is the title of the video being requested? How do you find that out?

8.2.3. When does the actual data transfer start? (We count the time starting from the beginning of the capture. In other words, the very first packet should have a timestamp showing 0 s. You can easily set the relative time in Wireshark Menu\View\ Time Display Format\Seconds Since Beginning of Capture.) Which one is the first packet carrying actual data? Why?

8.2.4. Starting from the first packet carrying data bytes, show visually the packets being exchanged between the client and the server for the first five rounds, i.e. until the 5. ACK is sent by the client. For visualisation, use a flow graph style similar to that of Fig. 23.19, p. 726, in Forouzan's book. Give sequence and acknowledgement numbers as well as the number of the first and last data bytes contained in each packet. Skip the information on the flags' status.

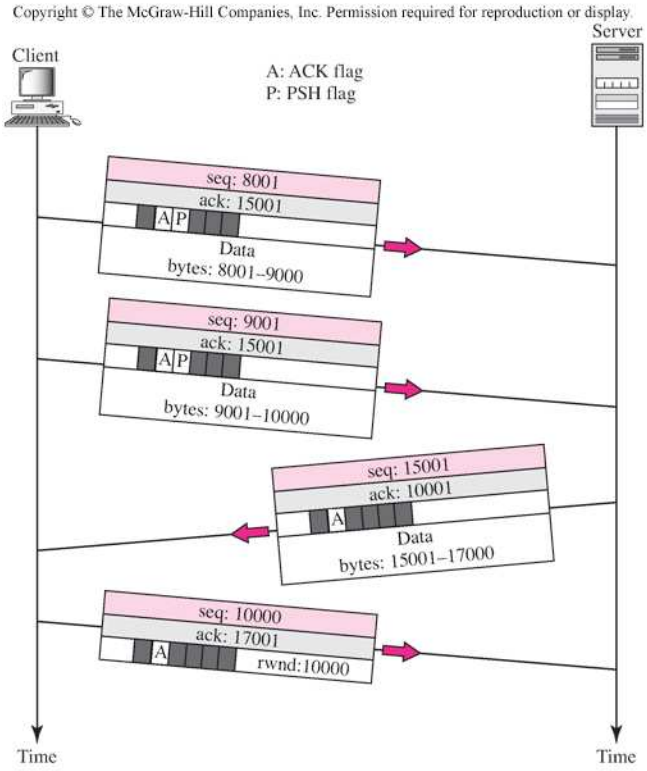

Figure 23.19: *Data transfer* – Note that we are not interested in the flags!

8.2.5. Why is 192.168.0.105 acknowledging (ACK) every other data packet (DATA) and not every single one? Is that normal? If so, what is this behaviour called? How is the value of the ACK selected? What does it tell to the sender?

8.2.6. Match the 5 acknowledgements with their corresponding data packets. Show in tabular format the times for DATA, its ACK, and the round trip time (RTT). (Look at the very bottom of the packet header details, you'll find some help under [SEQ/ACK analysis].) This is the actual RTT experienced by the network and is calculated as the difference of DATA and ACK. This is how you can validate your results.

8.2.7. Calculate the estimated RTT for the end of this 5-segment sequence. Note that you need to do it step-by-step, i.e. after the reception of each ACK. So, assume the estimated RTT to be equal to the sampled (actual) RTT for the first step (i=1). Then, use the following formula for each step (i=2,3,4,5):

```
EstimatedRTT(new) = 0.875 * EstimatedRTT(old) + 0.125 * SampleRTT(i++)
```

8.2.8. Starting from packet #20, node 192.168.0.105 starts decrementing its receiver-advertised window (RWND). Let's try to understand what happens to the other node's sliding window, i.e. that of 208.117.253.88, from this point on. We know that both nodes start with a congestion window (CWND) of 1 MSS. We also know from the connection establishment phase the size of MSS. As of the time of packet #20, 208.117.253.88 has received 6 ACKs, so it's CWND has become 7 MSS. Now, given the updates on RWND in the ACK packets #23, #26, #29, and #32 coming from 192.168.0.105, show how the sizes of CWND and of the sliding window change in bytes after each of these 4 packets. Remember that the sliding window's size is calculated as follows:

```
WindowSize = min (RWND , CWND)
```

8.2.9. Now, go to packet #238. Starting from this packet, you shall draw a flow graph similar to that of Fig. 23.26, p. 735, in Forouzan's book, for the packet exchange between the client and the server. The last packet on this graph should be #253. For the packets sent from the server to the client, identify the packets by their sequence numbers. For the packets from the client to the server, identify them by their acknowledgement numbers. Show the lost segment(s) from the server to the client, too. On the server side, also note the number of bytes in flight for each DATA segment sent to the client. (Look at the very bottom of the packet header details, you'll find some help under [SEQ/ACK analysis].)

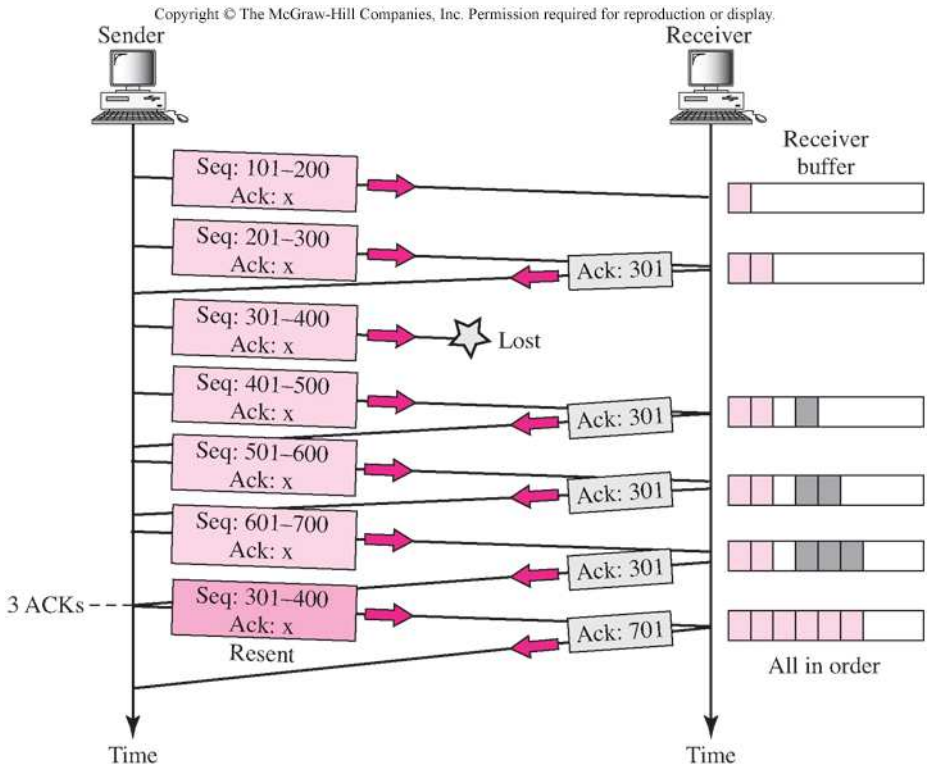

Figure 23.26: *Fast retransmission* – Note that we are not interested in the receiver buffer!

8.2.10. There is a sudden increase in the number of bytes in flight on the server side. Why? (Bytes in flight are those sent but not yet acknowledged by the receiver. See Fig. 23.23, p. 731, in Forouzan's book.)

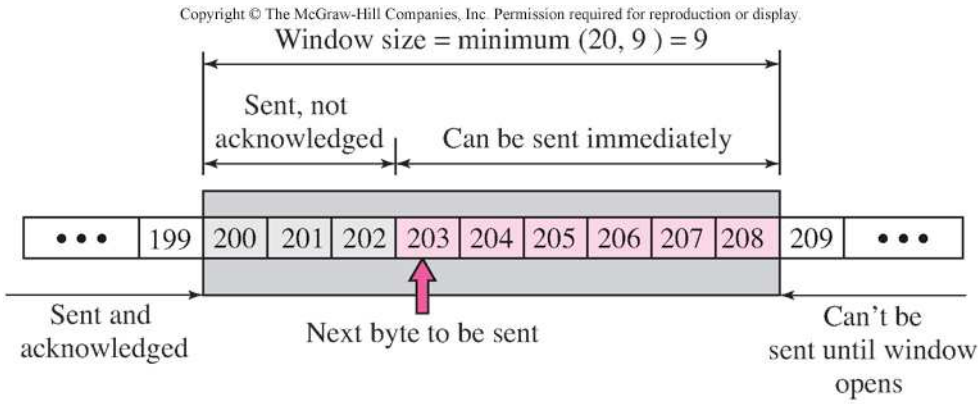

Figure 23.23: *Sliding window* – Note that the numbers are just showing an example!

8.3. Connection termination

As you can guess, the last couple of packets are designated for connection termination. Look at them and find the answers to the following questions. (Refer to § 23.3, pp. 727-728, in Forouzan's book if you need to refresh your memory.)

8.3.1. How many packets are exchanged this time? How do you recognise them?
8.3.2. Is the process similar to that of connection establishment? If not, what is different?
8.3.3. There is a special name for this particular termination process. What is it?

8.4. Overall assessment

We have seen a TCP connection being established, used for data transfer, and torn down. Now it is time to analyse from a holistic perspective the TCP session we've just closed. Answer the following questions. (Refer to §23.3, pp. 715-723, and § 24.4, pp. 768-773, in Forouzan's book if you need to refresh your memory.)

8.4.1. What is the average throughput for the whole duration of the TCP session? How do you calculate this value? Compare the result to the instantaneous throughput, which you can see by plotting the throughput graph under Wireshark Menu\Statistics\TCP Stream Graph\Throughput Graph. Remember that we're interested in the throughput of the connection from the server towards the client. So, make sure you have clicked on a packet which is sent in the right direction before plotting the graph.
8.4.2. Following the same menu, also plot the graph for the round trip time. What is the maximum RTT experienced during the TCP session? When did it happen according to the graph?
8.4.3. Finally, plot the Time/Sequence (Stevens's style) graph, which shows the sequence number of the segments received by the client versus time. (You can zoom in and out by clicking on a location on the graph.) What can you tell about the data flow in this TCP session by looking at this graph? What could be happening as the line goes diagonally upwards; what as it goes more or less horizontally?
8.4.4. Can you identify where TCP's slow start phase begins and ends, and where congestion avoidance takes over? Justify your answer.

9. Final Remarks

Please don't forget to finish your report with a conclusion section summarising what you have learned, together with your own reflections and comments on the lab. Finally, try to keep track of time, for we would like to know the amount of time you spent with this lab.

Congratulations! You've completed the TCP lab.