Technische Universität München
Informatik VIII
Netzarchitekturen und Netzdienste
Prof. Dr.-Ing. Georg Carle
Christian Grothoff, PhD
Dr. Nils Kammenhuber

Masterkurs Rechnernetze / Master Lecture on Computer Networks (IN2097) — Tutorial

Class Assignment No. 1, WS 2009/2010

Abgabedatum / To be handed in by: 2009-11-20, 14h

Exercise 1 — TCP fundamentals

Goal: Basic understanding of TCP; understanding connection setup and teardown; sequence numbers and acknowledgement numbers; ports.

Consider the following scenario: A Web client with IP address A downloads a Web page from a server with IP address B, using the HTTP protocol. The MSS (maximum segment size) is 1460 Byte.

a) Draw a table with the contents of the first five TCP segments being exchanged between A and B. For each packet, provide in human-readable form:
   - Sender ID (IP address)
   - Receiver (IP address)
   - Source port
   - Destination port
   - TCP Flags
   - Sequence number
   - ACK number
   - Body with payload data (if any)

   If deemed useful, you may use variables. Assume that the HTTP request is of the most simple form, i. e., without any additional headers.

b) What is the earliest point in time that A can send a packet with the FIN flag set? Why?

c) What is the earliest point in time that B can send a packet with the FIN flag set? Why?

d) Could A end the TCP connection by sending a packet with the RST flag set instead of the FIN? Discuss briefly.

e) Suppose that A wants to open multiple HTTP connections to B simultaneously. Is this possible? If so, how can A and B distinguish TCP segments from different connections? Explain.

Exercise 2 — TCP windows and throughput

Goal: Understanding characteristics of sliding window protocols; TCP congestion control; TCP performance issues in high-speed networks

Host A in Germany wants to send a huge amount of data to host B in New Zealand as fast as possible using a TCP connection. The network operator can guarantee a constant througput of of 10 Gbit/s, an

MSS of 600 Byte, and an RTT of 250 ms between A and B at all times[1].

a) If the full bandwidth of 10 Gbit/s is to be used, how many bytes of data can we expect to be in-flight, i. e., in the process of being forwarded through the network, at any given point in time?[2]

b) What does this mean for (i) the receiver window, (ii) the sender window, (iii) the congestion window? Looking at the TCP header as explained in the lecture, (iv) do you see any problems?

c) Host A starts sending its data at time $t = 0.0$ s. Let us assume that some friendly magician has bewitched our TCP stacks such that the problem with TCP that you just encountered has been fixed. Let us furthermore assume that no packet losses occur as long as the sending rate is lower than 10 Gbit/s, and let us for now assume that the Slow Start phase ends right when the window has grown to our required ≈ 300 MByte that you calculated before. When does this happen?

d) We find out that receiver B initially announces a `rwnd` of 500 Mbytes, i. e., more than we actually need. Let us again assume that no packet loss occurs. Furthermore, let us assume[3] that sender A initially sets $\texttt{ssthresh} = \frac{\texttt{rwnd}}{2}$. How long does it take now until A can send its data at the full rate of 10 Gbit/s?

Exercise 3 — TCP congestion control and packet losses

Goal: Understanding the effects of packet losses on TCP congestion control

We now investigate a TCP connection that has been running for some time, and that already has finished its Slow Start phase. For this phase, you may use an idealised periodic model that assumes that the congestion window always reaches the same maximum value W before a packet loss occurs. In this model, any such loss will be signalled by a Selective ACK message and fixed. Following this detection of a lost data segment, the congestion window is halved, and TCP resumes into Congestion Avoidance (i. e., TCP Reno).

a) Assume initially that $W = 16 \cdot \mathsf{RTT}$. Draw a diagram that show the current size of the congestion window (`cwnd`) on the Y axis over time on the X axis. Express the window size in units of 1 MSS; express time in units of 1 RTT. Assume that the congestion window size has just been halved at time $t = 0$. Draw three "saw teeth".

b) At the end of the Congestion Avoidance phase, one packet is dropped due to congestion. Calculate the loss rate[4] L in relation to the window size W (assume that W is expressed in RTTs).

c) Using this result, show that the average TCP sending rate during Congestion Avoidance is $R \approx \frac{1.22\mathsf{MSS}}{\sqrt{L}\mathsf{RTT}}$ if the packets are of maximum segment size and the packet loss rate is L. Literature tells[5] that $R = \frac{3}{4} \cdot \frac{W}{RTT}$.

Exercise 4 — Tweak your computer

Do some online research to find an implemented solution to at least one of the problems this exercise has revealed, and that can be activated in an operating system of your choice (e. g., via `/proc/sys/ipv4/tcp_*` in Linux). Briefly explain how you can activate the respective option in your chosen operating system, and briefly sketch how it alters the behaviour of the TCP stack. Do you notice any difference?

---

[1] 25000 km (distance with some detours) $\cdot\ 200000\ \frac{\mathsf{km}}{\mathsf{s}}$ (speed of light in glass) $\cdot\ 2$ (not one way, but round trip) $= 0.25$ s

[2] If you do not understand this question, read section 3.4.2 "Pipelined Reliable Data Transfer Protocols" in the Kurose/Ross book.

[3] This is not necessarily the case for all TCP implementations.

[4] The loss rate is defined as *number of packets lost / total number of transmitted packets*

[5] Reading the corresponding section in the Kurose/Ross book could render this exercise easier.