



Communication Networks Prof. Laurent Vanbever

Solution: Exercise 10 - Congestion control & DNS

Congestion Control

10.1 Drawing practice (Exam Question 2018)



Reaction of the CWND (dashed line) if the current link capacity (continuous line) is exceed by at most 2 kB (left) or by more than 2 kB (right).

In this task, you will draw the Congestion Window (CWND) evolution in reaction to the available capacity of a link in a network. The CWND follows the well-known TCP congestion control algorithm using slow-start. Whenever the CWND value exceeds the current link capacity, the CWND algorithm will react in the following way:

- 1. The current CWND value is kept for the entire next RTT (no increase or decrease);
- 2a. If the current link capacity was exceeded by at most 2 kBs, the CWND algorithm will observe three duplicate ACKs during the next RTT and will react appropriately (see the figure above on the left);
- 2b. If the current link capacity was exceeded by more than 2 kBs, the CWND algorithm will reach its timeout during the next RTT and will react appropriately (see the figure above on the right right).

Draw the CWND evolution directly into the figure on the next page in reaction to the link capacity indicated with the continuous line. Start at the bottom left corner (RTT 1, CWND 1 kB) and assume that the CWND corresponds to a flow that just started, e.g. you are in the slow-start phase. You can stop once you reach RTT 22. To help you, a correct portion of the CWND is plotted between RTT 11 and 14 (dashed line).



10.2 File Transfer over TCP



Network topology.

Consider the topology shown on the left-hand side. Host A opens a TCP connection to host B, and starts sending a file of size F=10 bytes, in segments of size MSS=1byte each. As a result of a faulty link, the 5th packet transmitted by A (without counting the SYN packet in the TCP handshake) is lost.

Assumptions:

- The transmission delay for packets is negligible.
- The round-trip time between A and B is RTT1.
- The transmission timer of host A has a fixed duration equal to 2.RTT1.
- TCP has Fast Retransmit disabled.
- A TCP receiver sends an ACK for each packet it receives.
- Host A does not send a separate ACK to acknowledge the SYN-ACK, but instead starts sending the first data segment.
- The first segment that A transmits will have a sequence number of 1.
- Host B stores (does not discard) all out-of-order packets.

Tasks:

a) Complete the sequence diagram on the next page with all packets exchanged between A and B (we have completed part of the diagram to help you get started).

Solution: cf. diagram

b) Calculate how much time it takes for B to finish receiving the file (Note: The one-way propagation delay from A to B is RTT1/2).

Solution: As marked on the sequence diagram on the next page, it takes 6.5 · RTT1 time to complete the file transfer (including the connection-setup time and excluding the time for the last ACK to reach A).



Now, assume A uses an application-layer proxy P to transmit the file to B, as shown on the left-hand side figure.

When P receives a connection request from A, it connects a TCP socket with B. After that, the proxy application receives data from the TCP socket connected to A (the input socket), and writes data out to the TCP socket connected to B (the output socket). P forwards these packets to the output socket the moment it can read them from the input socket. The proxy's operations do not incur any processing delay.

P is located exactly in the middle of the path between A and B, such that the round-trip times between A and P, and between P and B are both equal to RTT2= RTT1/2.

The faulty link described previously is now located on the part of the path between P and B. As such, the 5th packet transmitted on that part of the path is lost. No packet loss occurs on the part of the path between A and P.

Tasks:



Network topology after adding a proxy.

c) Calculate the time it takes for the file transfer to be completed in this new setting (Note: Do not forget to adjust the timeout interval for the two TCP flows; from A to P, and from P to B. The timeout interval for the two flows is equal to 2.RTT2=RTT1).

Solution: In this new setting, the transfer time can be computed as follows:

First, A connects to P and sends a TCP SYN. This takes RTT2/2. As soon as P receives this SYN from A, P initiates the handshake with B. The rest of the communication between A and P is carried out in parallel with the communication between P and B. Hence, for the first half of the path, we only need to consider the time it takes to propagate the first (connection) packet from A to P.

Now for the communication between P and B, the sequence diagram is an exact replica of the one we created in task **a**). The only difference is that now the round-trip time between P and B is half the round-trip time between A and B. Hence, it takes $6.5 \cdot \text{RTT2}$.

Thus, the total file transfer delay is: $RTT2/2 + 6.5 \cdot RTT2 = 7 \cdot RTT2 = 3.5 \cdot RTT1$.

d) Does the introduction of the application-layer proxy in the previous part improve or worsen the file transfer? Which features of TCP are responsible for this?

Solution: The main reason why the introduction of the proxy results in shorter file transfer is the reduction of the end-to-end RTT for each flow. This reduction in RTT has the following effects on TCP:

- The congestion window converges faster to an ideal value (assuming that link capacity is limited). In our example, where the capacity is unlimited, slow start will benefit even more.
- Lower RTT makes TCP more responsive in detecting and correcting channel error (e.g., the 5th packet which was lost). This is because the timeout interval adjusts to the RTT estimate; a smaller RTT means that a packet is retransmitted faster. We would observe the same behavior in the case where fastretransmit was also activated.

Note that there is a benefit even if the link is not faulty.

DNS

10.3 Local DNS server

You can perform DNS lookups with dig (Linux/Mac) or nslookup (Windows). First, perform a lookup for nyu.edu using your default DNS server by running the command dig nyu.edu or nslookup nyu.edu.

• What is the IP address of the server behind nyu.edu?

Solution: Note that the actual IP address can depend on the local DNS server you use. We got the following answer with dig:

;; ANSWER SECTION: nyu.edu. 60 IN A 216.165.47.10

Note that the format is slightly different for nslookup:

Non-authoritative answer: Name: nyu.edu Address: 216.165.47.10

Now, perform the same lookup, but use one of the DNS root servers (e.g., a.root-servers.net) by $running^1$

dig @a.root-servers.net nyu.edu nslookup nyu.edu a.root-servers.net

¹ If the nslookup command does not yield helpful output for you, try adding -type=soa after nslookup. (For some, it may be the other way round—adding this option may hide the relevant part in the output.)

• Why does the answer differ compared to the one from your local DNS server?

Solution: The request is not sent to an open DNS resolver, but to a DNS server that only provides answers about its own zone. Therefore, the root DNS server only points you to the name servers responsible for the next zone in the hierarchy, the edu zone.

We got the following answer with dig:

;; AUTHORITY SECTION: edu. 172800 IN NS a.edu-servers.net. edu. 172800 IN NS c.edu-servers.net. edu. 172800 IN NS d.edu-servers.net. ...

For nslookup, the output had this format (without -type=soa):

• How would you proceed with this answer to find the IP address behind nyu.edu?

Solution: Now that we know which servers are responsible for the edu zone, we can continue step-by-step just like your local DNS server would. Next, we would send a request to one of the edu name servers:

dig @a.edu-servers.net nyu.edu

The reply points us to the name servers in charge of the zone of NYU. By sending a request to them, we finally get the IP address behind the URL nyu.edu.

pagebreak if solutions

10.4 DNS Setup (Exam Question Summer 2021)

Assume you are the operator for the DNS sub-domain ee.ethz.ch. The responsible of the Networked Systems Group (NSG) reaches out to you as they want to create and independently maintain their own DNS sub-domain nsg.ee.ethz.ch. Among others, they want to create DNS names for:

- their group webpage www.nsg.ee.ethz.ch, pointing at 129.132.30.1;
- a project webpage fun.nsg.ee.ethz.ch, also pointing at 129.132.30.1;
- their chat room chat.nsg.ee.ethz.ch, pointing at 129.132.30.2; and
- a mail server mail.nsg.ee.ethz.ch, pointing at 129.132.30.3 and which should receive any email sent to a @nsg.ee.ethz.ch email address.

Unfortunately, the responsible of NSG does not seem to understand much about DNS. He needs your help to set this up.

You start by setting up two DNS name servers ns1.nsg.ee.ethz.ch and ns2.nsg.ee.ethz.ch which you host on 129.132.20.1 and 129.132.20.2, respectively.

a) Indicate the resource records that these two DNS servers should store. For each record, indicate its corresponding name, type, and value. *Note you might not need all 6 records.*

Solution:

- www.nsg.ee.ethz.ch A 129.132.30.1
- fun.nsg.ee.ethz.ch A 129.132.30.1
- chat.nsg.ee.ethz.ch A 129.132.30.2
- nsg.ee.ethz.ch MX mail.nsg.ee.ethz.ch
- mail.nsg.ee.ethz.ch A 129.132.30.3

The group webpage, project webpage, and chat room each need an IPv4 address record (A) s.t. the name server can map their respective domain names to the corresponding IPv4 address. For the mail address resolution, the name servers needs an MX record that points to the mail server's name, and again a corresponding IPv4 address record that resolves the mail server's name to an address. Note that the type of *either* the fun.nsg.ee.ethz.ch *re*cord could also be CNAME.

b) Indicate the resource records that need to be added (if any) to the name servers responsible for ee.ethz.ch. Briefly explain the usage of these extra records or why no such extra record is needed. Note you might not need all 4 records.

Solution:

- nsg.ee.ethz.ch NS ns1.nsg.ee.ethz.ch
- nsg.ee.ethz.ch NS ns2.nsg.ee.ethz.ch
- ns1.nsg.ee.ethz.ch A 129.132.20.1
- ns2.nsg.ee.ethz.ch A 129.132.20.2

c) Indicate the resource records that need to be added (if any) to the name servers responsible for ethz.ch. Briefly explain the usage of these extra records or why no such extra record is needed. *Note you might not need all 4 records.*

Solution: No new entries are required: This name server only needs to know the IP address the name server for ee.ethz.ch. Since this subdomain was already in use beforehand, no new entries are needed. Concretely, the ethz.ch name server will point to the ee.ethz.ch name server during address resolution, and the ee.ethz.ch name server will then take care of pointing to the NSG name servers. The NSG name servers can then resolve the custom NSG subdomains to the correct IP address. (If any of the involved name servers already have the record cached, they can of course directly reply with it and the resolution stops.)

d) While setting up the records, you observe that NSG wants to host two webservers (for www and fun) on the same IP address (129.132.30.1). Is that possible? If so, briefly explain how this works. If not, briefly explain why.

Solution: Yes, it is possible with virtual hosting: Here, the requesting side puts the target host name into the HTTP Host header. With this, the server hosting the two webservers can distinguish between the requests and deliver the correct response.

e) It looks like you underestimated NSG's popularity! It appears the two name servers ns1.nsg.ee.ethz.ch and ns2.nsg.ee.ethz.ch are completely overloaded with (legitimate) DNS requests.

Explain two *distinct* techniques you could use to scale the system further.

Solution: Technique 1: For each name server, one can add more devices with different IPs. Then, one installs new A records in the **ee.ethz.ch** name server, s.t. the NSG name servers' domains map to several IPs. The **ee.ethz.ch** name server then round-robins between the different IPs. This has a load-balancing effect. Technique 2: One can add another nameserver (ns3), and again add the corresponding A records to the **ee.ethz.ch** name server.

(Many more techniques are possible, e.g. using a CDN)