

Communication Networks

Prof. Laurent Vanbever

Exercises 8 – Go-Back-N and DNS

8.1 Go-Back-N Warm-Up Questions

Sender and receiver keep separate windows and buffers for sent and received segments.

- Compare how the sender and the receiver advance their respective windows.
- Which segments does the *sender* buffer? When can segments be removed from the buffer?
- A *receiver* typically buffers out-of-order segments. What is the advantage of such a buffer?

Cumulative ACKs acknowledge that all segments *up to* the acknowledged segment have been received.

- Why are cumulative ACKs used? Do they help with lost data segments, lost ACKs, or both?

When *Fast Retransmit* is used, the sender retransmits a segment after duplicate ACKs.

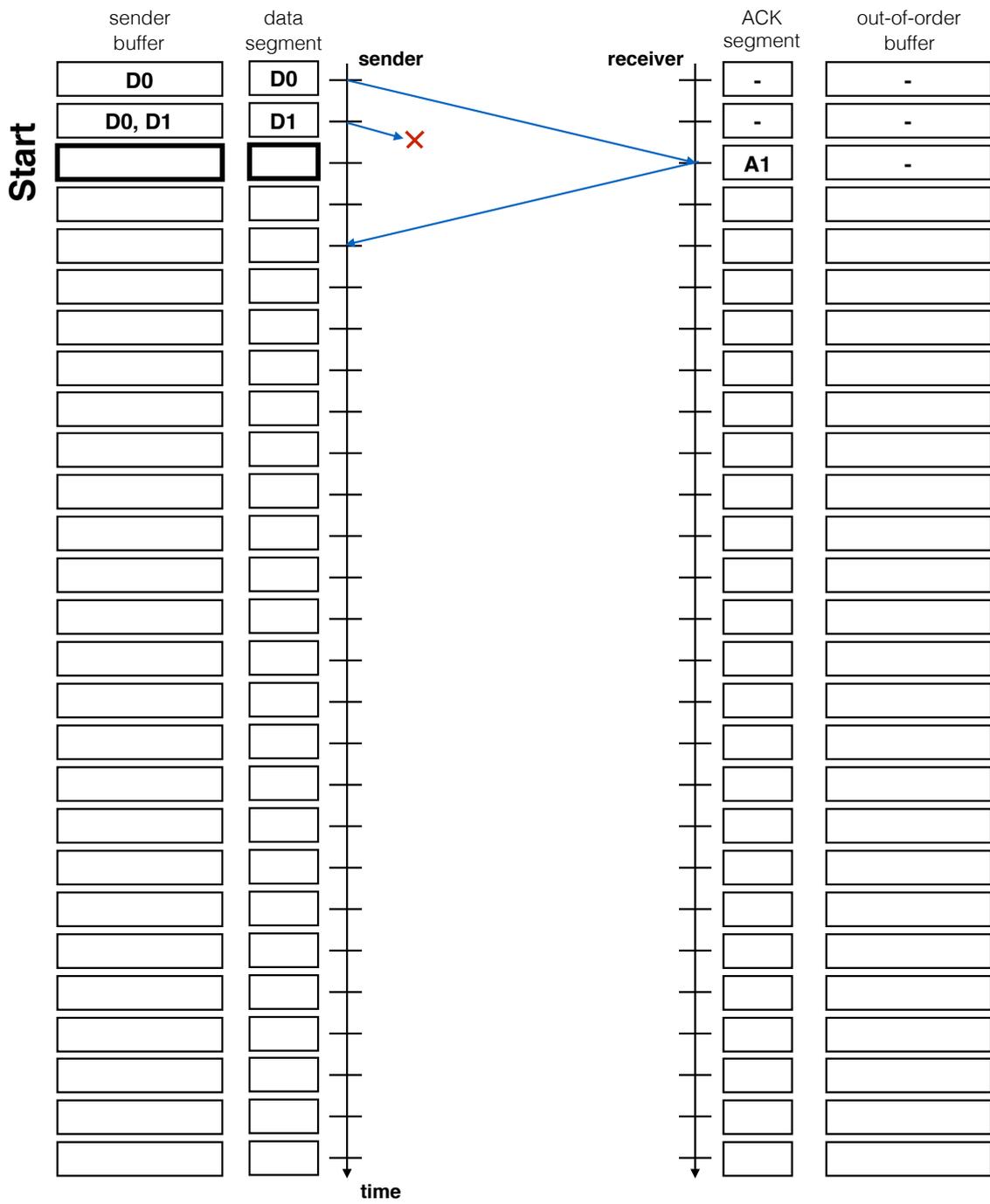
- How does Fast Retransmit improve performance?
- Compare Fast Retransmit in the case of mild congestion (some segment losses) and heavy congestion (nearly all segments lost).

8.2 Go-Back-N (Exam Question 2017)

Consider a Go-Back-N (GBN) protocol with the following implementation choices for sender and receiver.

- The sender and receiver window have a size of 4;
- The receiver saves out-of-order segments in an (infinite) buffer and removes them as soon as the missing segment(s) arrive;
- The receiver uses cumulative ACKs which acknowledge all previous segments and point to the next expected data segment;
- The sender uses Fast Retransmit. After three duplicate ACKs, the sender immediately retransmits the corresponding data segment. For instance, if the sender gets the following ACKs [A1, A1, A1], it will immediately retransmit the data segment D1;
- For each tick in the diagram below, the sender can send one data segment and the receiver can send one ACK. Sender and receiver will first analyze the incoming packet and then send a data segment/ACK;
- The sender uses a retransmission timer of 5 ticks. Each time it sends a data segment or receives an ACK, the timer is reset. After a timeout, the sender retransmits all current segments in its sender buffer (in order, one segment per tick);
- A data segment or ACK needs two ticks to travel to the other end of the connection. See the given start in the diagram.

Your task: Use the diagram on the next page to draw the successful transmission of 6 data segments (D0 to D5) if the **first data segment** (D1, already indicated) is **lost** as well as **ACK A5 is lost the first time it is sent**. For each tick, indicate which data segment or ACK is transmitted (if any) as well as the content of the sender and out-of-order buffer.



8.3 Local DNS server

On Linux and Mac computers you can use the command line tool `dig` to perform DNS lookups. The corresponding tool for Windows is `nslookup`. First, perform a lookup for `nyu.edu` using your default DNS server by running the command:

```
dig nyu.edu
```

```
nslookup nyu.edu
```

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

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

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

```
nslookup nyu.edu a.root-servers.net
```

- Why does the answer differ compared to the one from your local DNS server?
- How would you proceed with this answer to find the IP address behind `nyu.edu`?

8.4 Local vs. authoritative DNS server

Perform a DNS query for `uzh.ch` using first the authoritative DNS server (`ns1.uzh.ch`) and then your local server.

Note: When using `nslookup` on Windows, you need to specify the `-debug` flag to get the relevant information for this task. For example:

```
nslookup -debug uzh.ch
```

- Compare the ANSWER SECTION of the responses. Can you see differences between the answers from your local DNS server and the authoritative server? Run the query to your local server multiple times to make the differences more obvious.
- What is the reason for this difference?
- As you have seen in the lecture, DNS can be used to balance the incoming load. What are the considerations one has to make when using DNS load balancing with respect to the TTL?

8.5 Multiple answers

Whenever a client (e.g., your computer) receives multiple IP addresses as answer to a DNS lookup, it picks the very first one. Only if that one does not work, it tries the next one in order.

When you run `dig yahoo.com`, you receive multiple IP addresses as an answer compared to, for example, `dig google.com`.

Can you think of a reason for providing multiple IP addresses? Run the lookup for `yahoo.com` multiple times.