Übung 11: TCP

Aufgabe 1: TCP – Unidirektionale Kommunikation

Host A und Host B kommunizieren über eine TCP Verbindung. Host B hat von A bereits ***alle*** Bytes bis einschließlich Byte 126 korrekt empfangen und bereits erfolgreich an A bestätigt. Nun schickt Host A direkt hintereinander 2 Segmente an Host B (*Segment 1* und *Segment 2*), ohne nach dem Senden von *Segment 1* auf das dazugehörige Acknowledgment zu warten:

- Das *Segment 1:* 80 Bytes Nutzdaten, Sequenznummer 127, Source-Port 302, Destination Port 80.
- Das *Segment 2* 40 Bytes an Nutzdaten.

Annahme: Host *B* sendet immer sofort ein ACK an *A*, sobald ein Segment von *A* empfangen wurde.

a) Was ist die Sequenznummer, der Source- und der Destination-Port im *Segment 2?*

b) Szenario 1: *Segment 1* kommt vor *Segment 2* bei *B* an. Was ist die ACK-Nummer, Source- und Destination-Port im Acknowledgment, das B unmittelbar nach Erhalt von *Segment 1* an A sendet?

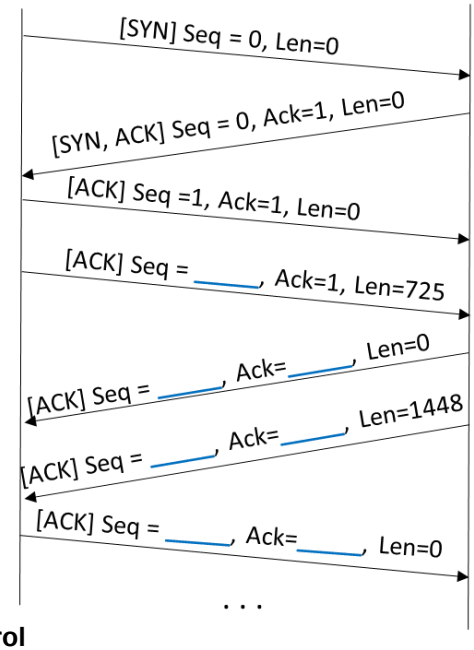
c) Szenario 2: *Segment 2* kommt vor dem *Segment 1* bei B an. Was ist die ACK-Nummer im Acknowledgment, das *B* unmittelbar nach Erhalt von *Segments 2* an *A* schickt?

d) Es gilt nun wieder Szenario 1: Das Acknowledgment für *Segment 1* geht auf dem Weg zu *A* verloren, während das Acknowledgment für *Segment 2* erst *nach* dem TCP Timeout des *Segment 1* bei A ankommt. ***Zeichnen*** Sie ein Sequenzdiagramm. Nehmen Sie an, dass kein weiterer Paketverlust auftritt!

Aufgabe 2: TCP – Bidirektionale Kommunikation

Das rechte Sequenzdiagramm zeigt den Beginn einer TCP Verbindung. Es gibt keine Paketverluste oder Retransmissions. Beide Empfänger bestätigen den Empfang von Nutzdaten immer mit der nächsten Nachricht. Das Feld `Len` gibt für jedes Paket die Größe der mitgesendeten Nutzdaten in Bytes an.

Ergänzen Sie die 7 fehlenden SEQ-/ACK-Nummern an den **blau** gekennzeichneten Stellen. Die Zuweisung muss Sinn ergeben und den obigen Annahmen genügen.



Aufgabe 3: TCP Connection Management und Flow Control

Quelle RFC 793: https://tools.ietf.org/html/rfc793

a) Lesen Sie den Beginn von Seite 27 der RFC. Warum kann es ungünstig sein, wenn eine TCP-Seite bei Verbindungsaufbau als initiale Sequenznummer immer die 0 wählt. Überprüfen Sie anhand des gegebenen Wireshark-Traces, ob in der Praxis initial die 0 gewählt wird.

b) Schauen Sie sich in RFC793 den Zustandsautomaten von TCP auf Seite 23 genauer an.
   - Welche Zustände durchläuft ein TCP Client beim Verbindungsaufbau?
   - Welche Zustände durchläuft ein TCP Server beim Verbindungsaufbau?

c) Flusskontrolle: Während einer laufenden Verbindung informiert der Empfänger B den Sender A, dass er überlastet ist und sendet `rwnd=0`. Was passiert?

d) Ein bösartiger Dritter *C* möchte eine zwischen *A* und *B* bestehende TCP Verbindung beenden. *C* sendet deshalb von seiner IP Adresse ein TCP-FIN Paket an den Host *B* unter Verwendung des korrekten Ports. Was passiert?

Aufgabe 3: TCP mit Wireshark

Der bereitgestellte Wireshark Trace hat das Hochladen einer großen Textdatei per „HTTP-Post“ auf einen HTTP Server aufgezeichnet.

a) Welche IP Adressen und Portnummern hat der TCP Client und TCP Server?

b) Suchen Sie das `HTTP Post` Paket. Sie erkennen, dass das HTTP Paket wegen der Größe der Textdatei in mehrere TCP Pakete aufgeteilt werden musste. Wireshark macht dies im zuletzt übertragenen TCP Paket durch eine „Zwischenschicht“ zwischen TCP und HTTP mit dem Namen „Reassembled TCP Segments“ deutlich. Wie viele TCP Pakete wurden demnach benötigt, um en ***HTTP Request*** zu übertragen?

c) Welche Sequenznummer hat das erste Paket der TCP Verbindung? Woran können Sie erkennen, dass es sich um ein SYN Paket handelt?

d) Welche Sequenz- und Acknowledgment-Nummer hat das dazugehörige SYNACK Paket? Woran können Sie und Wireshark erkennen, dass es sich um ein SYNACK Paket handelt?

e) Welche Paket- und Sequenznummer hat das TCP Segment, das den Text `HTTP Post` enthält? *Tipp*: Aufpassen, ggfs. das „Packet Content Field“ ganz unten in Wireshark beachten.

f) Betrachten Sie das TCP Segment, das das `HTTP Post` enthält, siehe letzte Teilaufgabe. Tragen Sie in folgende Tabelle die Paketnummern (aus Wireshark) und die TCP Sequenznummern der ersten 4 Segmente vom Client zum Server ein. Geben Sie ferner jeweils an, in welchem Paket der Server erstmalig den Erhalt bestätigt.

	Paketnummer in Wireshark	Sequenznummer	Paketnummer, in der Segment bestätigt wird.
1. Segment			
2. Segment			
3. Segment			
4. Segment			

g) Wie groß ist das Empfangsfenster (`rwnd`, Flusskontrolle), das der Server in der SYN ACK Nachricht an den Empfänger meldet. Überfliegen Sie den Trace und entscheiden Sie ob der Empfänger (Server) den Sender im Laufe der Verbindung abbremst.

h) Gab es in dem Trace Retransmissions? Begründen Sie ihre Antwort unter Zuhilfenahme des sogenannten Time-Sequence-Graphs (Stevens), den Sie unter `Statistics` → `TCPStreamGraph` (Stevens) erstellen können. Dazu muss ein Datenpaket markiert sein.