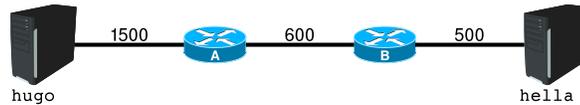


Übungsblatt 9

Abgabe bis **27.06.2014** in der Vorlesung, oder per Uniworx.

Hinweis: Schreiben Sie unbedingt Ihre Übungsgruppe auf Ihre Abgabe!

1. Fragmentierung (H)



Der Rechner **hugo** möchte Daten an den Rechner **hella** übertragen. Die Abbildung zeigt die beiden Rechner und dazwischen befindliche Router, sowie Leitungen, die mit ihrer MTU beschriftet sind.

- Bei der Vermittlung der Daten zu **hella** tritt Fragmentierung auf. Wieviele IPv4-Fragmente empfängt **hella** mindestens, bis 5000 Bytes Nutzdaten empfangen wurden? *Hinweis:* **hugo** verschickt pro Rahmen maximal viele Nutzdaten.
- Erstellen Sie eine Tabelle die in chronologischer Reihenfolge, die Länge in Bytes, gesetzte Header-Flags und das Fragment Offset der von **hella** empfangenen IPv4-Nachrichten zeigt!
- Erstellen Sie analog zu Teilaufgabe b eine Tabelle, für den Fall, dass für die Übermittlung IPv6 zum Einsatz kommt!

2. Staukontrolle bei TCP (H)

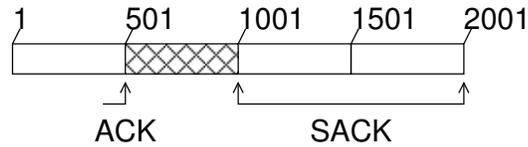
Bei TCP kommt Slow-Start als Mechanismus zur Staukontrolle zum Einsatz. Es sollen über eine TCP-Verbindung mit Netzverzögerung von $100ms$ (d.h. $RTD = 200ms$) $7500Byte$ Nutzdaten in 15 Segmenten gleicher Größe $S = 500Byte$ übertragen werden.

Hinweis: Nehmen Sie einen nicht erreichbaren Wert für **Threshold** an, und vernachlässigen Sie Verluste, Empfangspuffergröße, Verarbeitungsverzögerung, die Übertragungszeit für Protokollheader sowie den Verbindungsabbau. Gehen Sie also davon aus, dass die beiden TCP-Instanzen sofort nach dem Verbindungsaufbau die Übertragung beginnen und danach die Slow-Start-Phase nicht verlassen.

- Sei die Übertragungsrates $R = 20kByte/s$. Erstellen Sie ein Sequenzdiagramm für die Übertragung und tragen Sie die Größe des jeweils aktuellen Überlastfensters (**CongWin**) in das Diagramm ein!
- Bestimmen Sie die Übertragungsdauer gemessen vom Absenden des SYN des Clients bis alle Nutzdaten empfangen wurden:
 - mit Slow-Start.
 - mit reinem Go-Back-N mit Fenstergröße 20 (also „ohne Slow-Start“.)
- Wie lange würde die Übertragung jeweils mit und ohne Slow-Start für $R = 500kByte/s$ dauern?

3. Selektive Quittungen

Der Verlust einzelner Segmente kann zur unnötigen Wiederholung großer Datenmengen führen, insbesondere bei Pfaden mit hoher Netzverzögerung. Durch die Einführung selektiver Quittungen (SACK, siehe RFC 2018) kann dieses Problem gemildert werden.



Statt wie bei den „normalen“ kumulativen TCP-Quittungen den bis dahin korrekt empfangenen *zusammenhängenden* Byte-Strom zu quittieren, kann ein Empfänger mit selektiven Quittungen zusätzliche Segmente (oder zusammenhängende Folgen von Segmenten, sogenannte Blöcke) im `Options`-Feld des TCP-Headers als empfangen notieren. Hierzu wird ein Bereich des Byte-Stroms mit Anfangs- und End-Byte notiert. Auf Grundlage der obigen Abbildung würde etwa eine Quittung mit `AckNr=501` und `SACK-Block=(1001,1500)` gesendet; das verlorene Segment mit Bytes 501–1000 (schraffiert) wird so ausgespart. Es können mehrere solche Blöcke in den Optionen angegeben werden.

Gehen Sie von einem Sender mit aktueller `SeqNr=5000` und einem Empfänger (`AckNr=2001`) aus. Der Sender sendet 8 Segmente von jeweils 500 Byte Länge. Wie werden kumulative und selektive Quittungen benutzt, wenn:

- (a) die ersten vier Segmente empfangen werden, die letzten vier aber verloren gehen?
- (b) das zweite, vierte und sechste und achte Segment verloren gehen?