

2.1 Grundlagen

2.2 Protokolle und Dienste

2.3 Sockets

2.4 Grundlagen der Datenübertragung

2.4.1 Paketvermittlung

2.4.2 Verzögerungen

2.4.3 Paketverlust

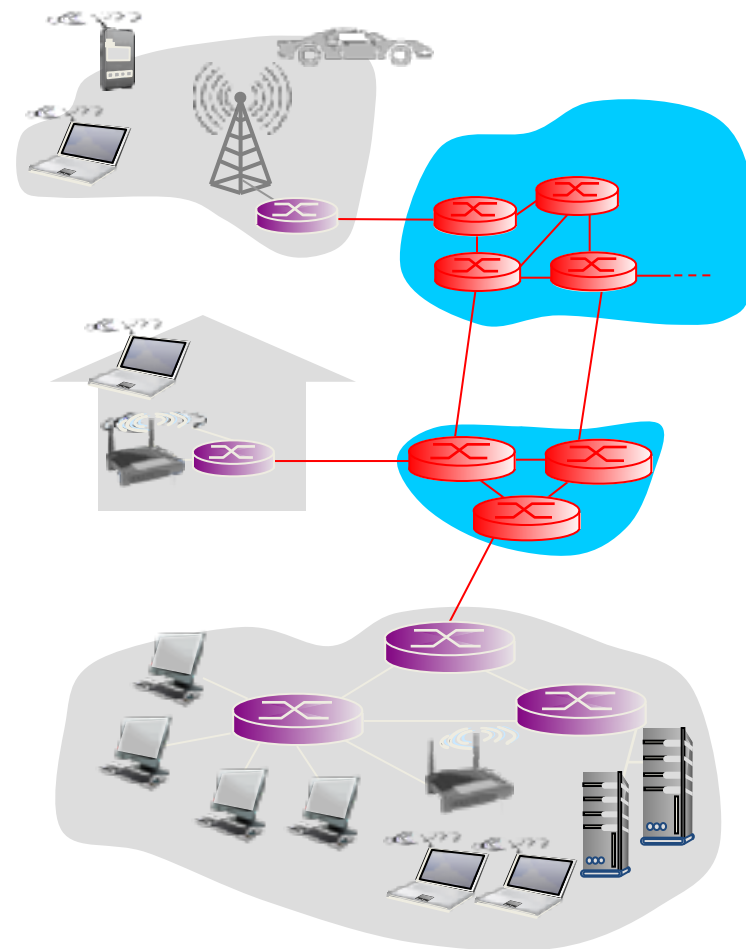
2.4.4 Durchsatz

2.4.5 Messungen im Internet

2.5 Aufbau des Internets

2.6 Zusammenfassu
ng

- Protokolle der Transportschicht teilen Nachrichten der Anwendung in Pakete (Segmente) auf bzw. segmentieren Pakete
- Die Netzwerk- bzw. Netzzugangsschicht schreibt in den Header jedes Pakets seine Zieladresse
- Pakete werden von Netzknoten (Switch, Router) aufgrund der Zieladresse auf dem richtigen Port über den richtigen Link an der richtigen Nachbarknoten weitergeleitet

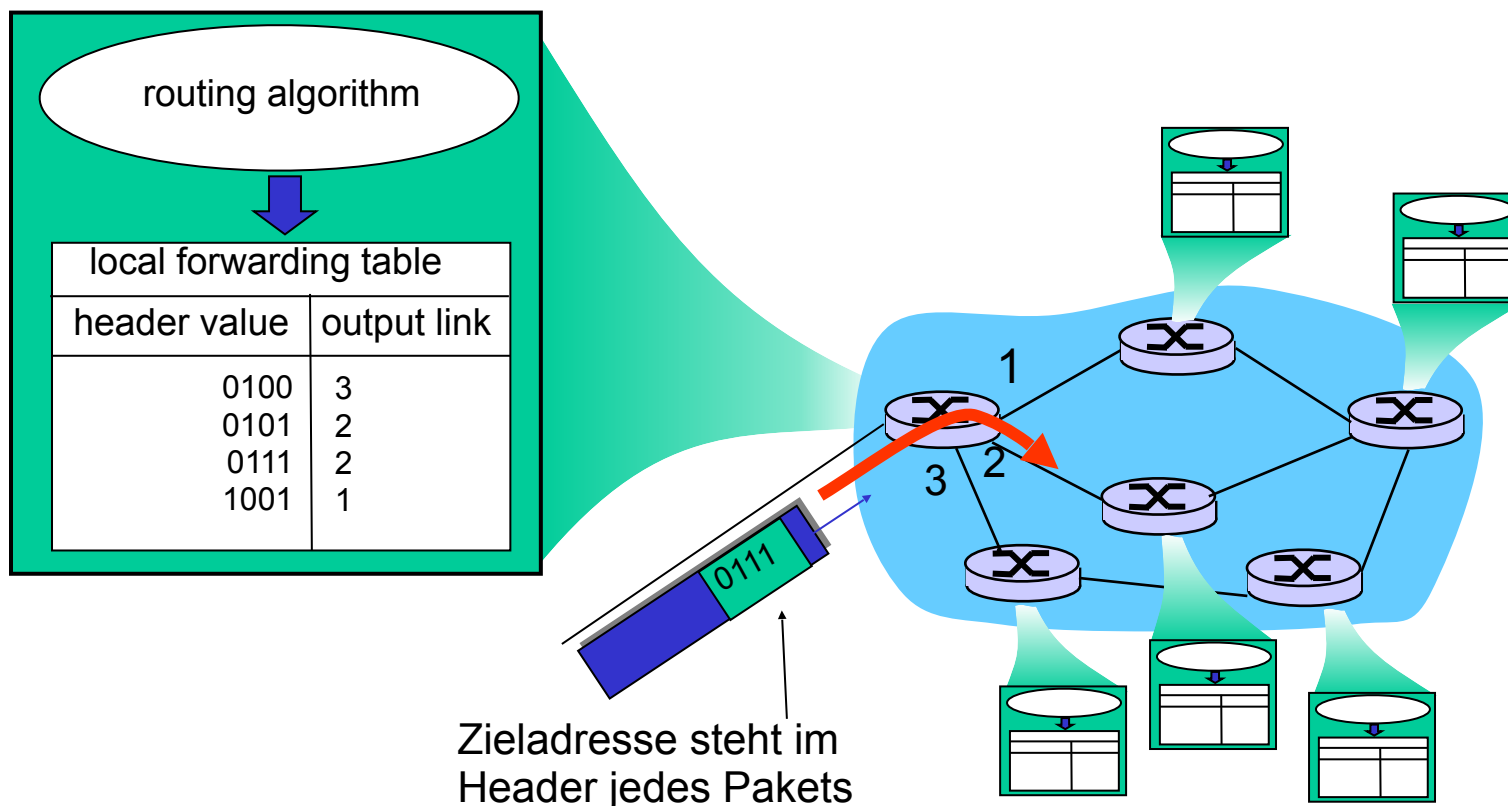


Routing:

- bestimmt für jede Zieladresse die Route, die das Paket vom Ursprung zum Ziel nimmt
- speichert den Ausgangsport und die Adresse des nächsten Routers in der Forwarding Tabelle

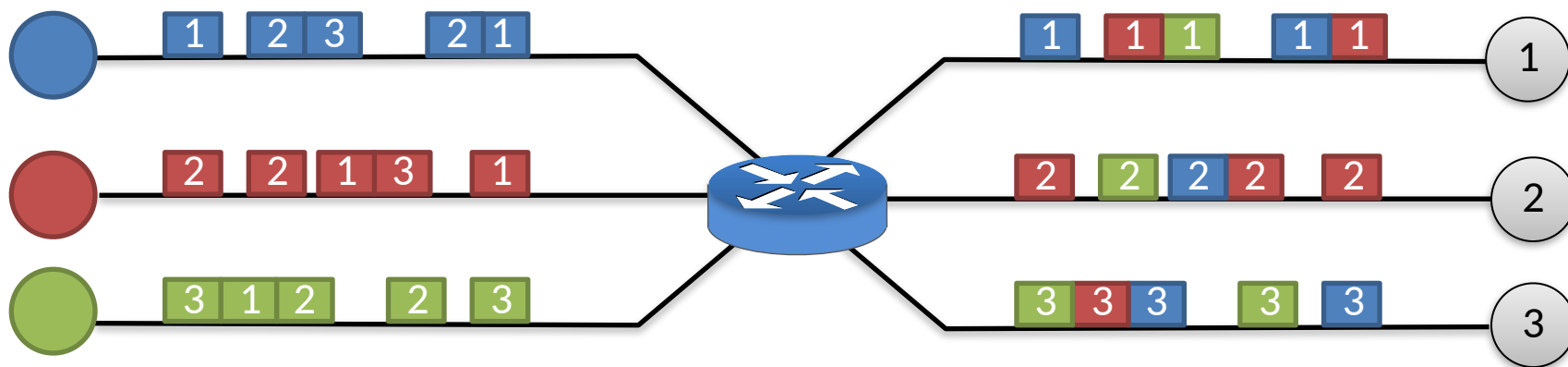
Forwarding:

- schaut in der Forwarding Table den Ausgangsport zur Zieladresse eines Pakets nach
- leitet innerhalb des Routers das Paket vom Eingangsport zum Ausgangsport weiter



Paketvermittlung

- Mehrere Verkehrsflüsse teilen sich einen Link, indem ihre Daten Paket für Paket in der Reihenfolge des Eintreffens nach dem Prinzip FCFS (First Come First Serve) weitergeleitet werden
 - ein Verkehrsfluss belegt einen Link immer nur für die kurze Zeit, während der ein Paket dieses Paketflusses gesendet wird
 - durch die Paketvermittlung kann eine Leitung sehr effizient von verschiedenen Verkehrsflüssen, die auch unterschiedliche Datenraten haben können, gleichzeitig genutzt werden



2.1 Aufbau des Internets

2.2 Protokolle und Dienste

2.3 Sockets

2.4 Grundlagen der Datenübertragung

2.4.1 Paketvermittlung

2.4.2 Verzögerungen

2.4.2.1 Beispiel: WAN-Link

2.4.2.2 Beispiel: LAN-Link

2.4.2.3 Beispiel: Übertragung eines „Packet Bursts“

2.4.2.4 Verzögerungen in einem Knoten

2.4.2.5 Theoretisches Modell

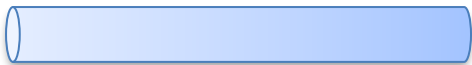

2.4.3 Paketverlust

2.4.4 Durchsatz

2.4.5 Messungen im Internet

2.5 Aufbau des Internets

2.6 Zusammenfassung

- Link 
 - Kapazität C
 - Physikalische Länge l
 - Ausbreitungsgeschwindigkeit: v
- Paket: 
 - Größe L
- Wie lange dauert die Übertragung eines Paket über den Link?
 - Übertragungsverzögerung?
 - Ausbreitungsverzögerung ?
 - Gesamtverzögerung ?

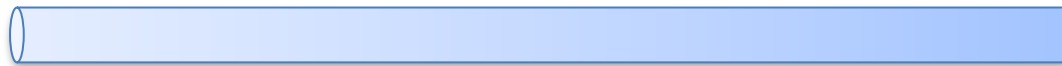
- Beispiel

$$C = 10 \text{ Gbps}$$

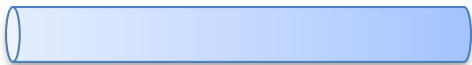

$$l = 200 \text{ km}$$

$$v = 200000 \text{ km/s}$$

$$L = 10 \text{ kbit}$$



Dauer der Übertragung eines Paket über einen WAN-Link

- Link 
 - Kapazität C
 - Physikalische Länge l
 - Ausbreitungsgeschwindigkeit: v
- Paket: 
 - Größe L

- Beispiel

$$C = 10 \text{ Gbps}$$

$$l = 200 \text{ km}$$

$$v = 200000 \text{ km/s}$$

$$L = 10 \text{ kbit}$$

- Wie lange dauert die Übertragung eines Paket über den Link?

- Übertragungsverzögerung: $t_{tx} = \frac{L}{C} = \frac{10 \text{ kbit}}{10 \text{ Gbps}} = \frac{10e3 \text{ bit}}{10e9 \text{ bps}} = 1 \mu s$
- Ausbreitungsverzögerung: $t_{prop} = \frac{l}{v} = \frac{200 \text{ km}}{200000 \text{ km/s}} = \frac{2e5 \text{ m}}{2e8 \text{ m/s}} = 1 \text{ ms}$
- Gesamtverzögerung: $t_{link} = t_{tx} + t_{prop} = 1 \mu s + 1 \text{ ms} = 1,001 \text{ ms}$

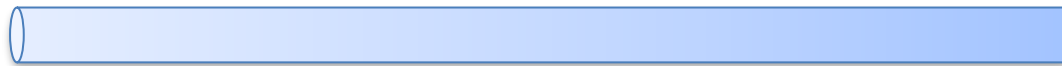
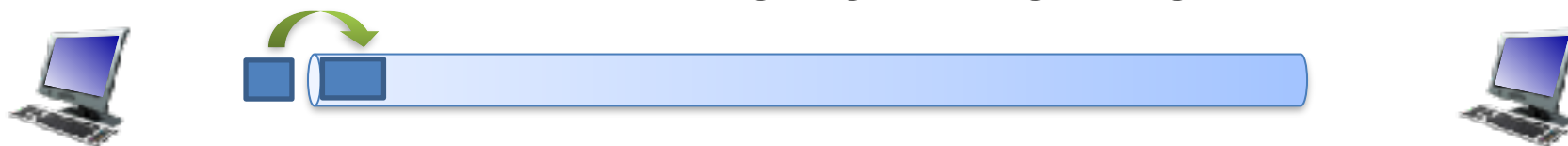


Illustration des Übertragungsvorgangs (WAN-Link)

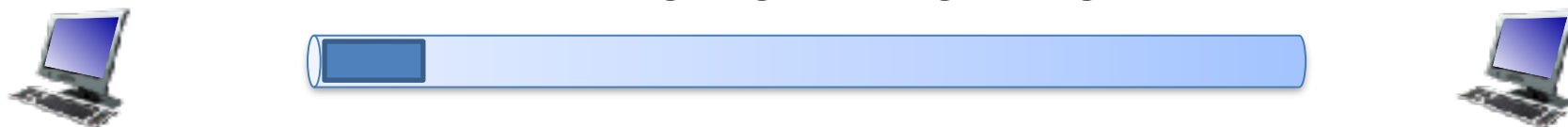
- Situation vor dem Versenden



- Situation während der Übertragungsverzögerung



- Situation nach der Übertragungsverzögerung



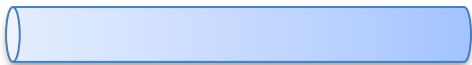


- Situation während der Ausbreitungsverzögerung



- Situation nach der Ausbreitungsverzögerung



Dauer der Übertragung eines Paket über zwei WAN-Links

- Links 
 - Kapazität C
 - Physikalische Länge l
 - Ausbreitungsgeschwindigkeit: v
- Paket: 
 - Größe L
- Wie lange dauert die Übertragung eines Paket über zwei Links?
 - Antwort 1: $t_{e2e} = t_{tx} + t_{prop}$
 - Antwort 2: $t_{e2e} = t_{tx} + 2 \cdot t_{prop}$
 - Antwort 3: $t_{e2e} = 2 \cdot t_{tx} + t_{prop}$
 - Antwort 4: $t_{e2e} = 2 \cdot (t_{tx} + t_{prop})$  $t_{e2e} = 2,002ms$



Packet-Switching Prinzip: Store-and-forward

- Store-and-forward (Speichern und Weiterleiten):
 - Ein Paket muss erst vollständig empfangen werden, bevor es zum nächsten Router weitergeschickt werden kann.

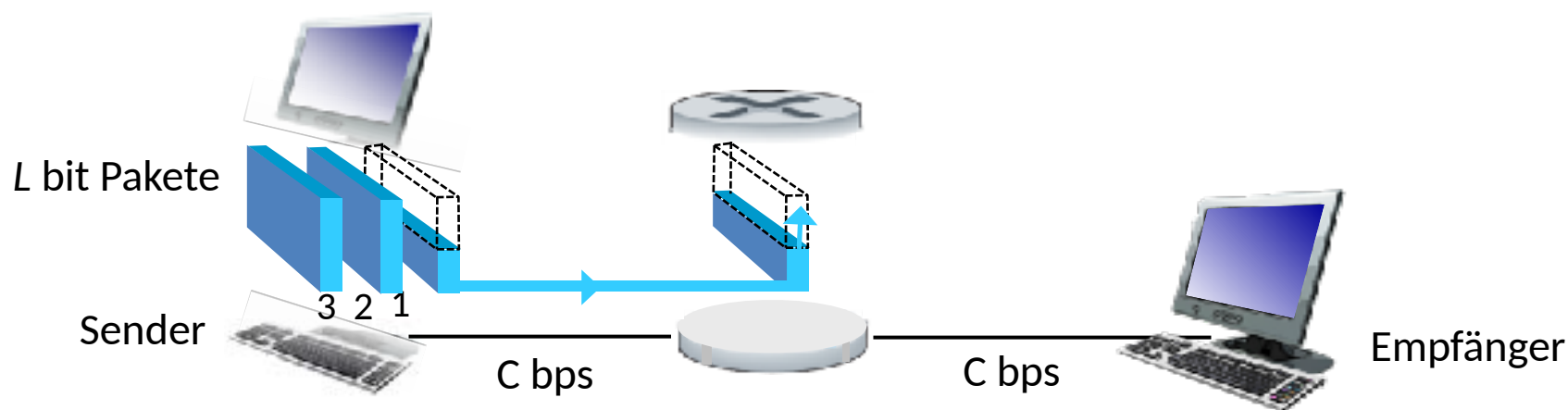


Illustration des Übertragungsvorgangs

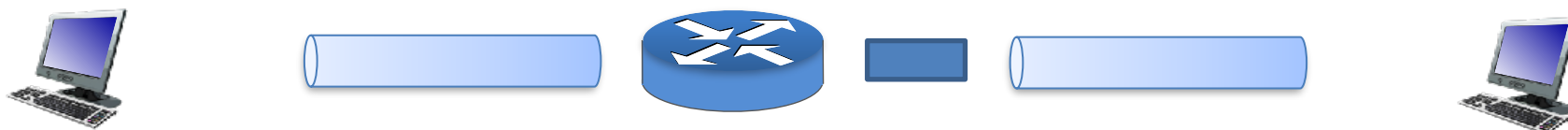
- Situation vor dem Versenden



- Situation nach Übertragungs- und Ausbreitungsverzögerung Link 1



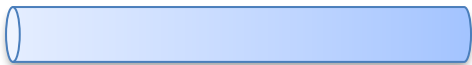


- Situation nach Bearbeitung (Überprüfen der Korrektheit des Pakets, Forwarding) im Router



- Situation nach Übertragungs- und Ausbreitungsverzögerung Link 2



Dauer der Übertragung von zwei Paketen über einen WAN-Link

- Link 
 - Kapazität C
 - Physikalische Länge l
 - Ausbreitungsgeschwindigkeit: v
- 2 Pakete: 
 - Größe L
- Wie lange dauert die Übertragung von zwei Paketen?
 - Antwort 1: $t_{e2e} = t_{tx} + t_{prop}$
 - Antwort 2: $t_{e2e} = t_{tx} + 2 \cdot t_{prop}$
 - Antwort 3: $t_{e2e} = 2 \cdot t_{tx} + t_{prop}$  $t_{e2e} = 1,002 \text{ ms}$
 - Antwort 4: $t_{e2e} = 2 \cdot (t_{tx} + t_{prop})$

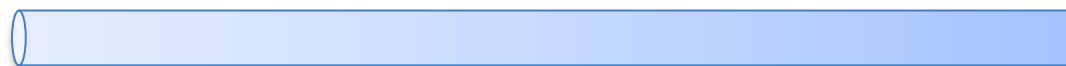
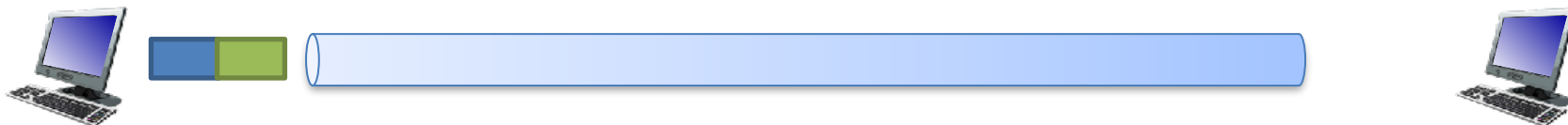


Illustration des Übertragungsvorgangs

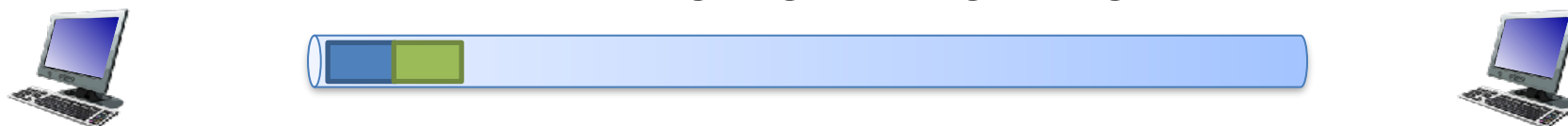
- Situation vor dem Versenden



- Situation nach einer Übertragungsverzögerung



- Situation nach zwei Übertragungsverzögerungen



- Situation nach Übertragungs- plus Ausbreitungsverzögerung



- Situation nach zwei Übertragungs- und einer Ausbreitungsverz.



Übertragung von zwei Paketen über zwei WAN Links

- 2 Pakete: 
 - Größe $L = 10kbit$

- Link 1: 

$$C = 10 \text{ Gbps}$$

$$l = 200 \text{ km}$$

$$v = 200000 \text{ km/s}$$

$$t_{tx,1} = 1 \mu s$$

- Link 2: 

$$C = 1 \text{ Gbps}$$

$$l = 200 \text{ km}$$


$$v = 200000 \text{ km/s}$$

$$t_{tx,2} = 10 \mu s$$

- Wie lange dauert die Übertragung von zwei Paketen?

- Antwort 1: $t_{e2e} = t_{tx,1} + t_{tx,2} + 2 \cdot t_{prop}$

- Antwort 2: $t_{e2e} = 2 \cdot t_{tx,1} + t_{tx,2} + 2 \cdot t_{prop}$

- Antwort 3: $t_{e2e} = t_{tx,1} + 2 \cdot t_{tx,2} + 2 \cdot t_{prop}$  $t_{e2e} = 2,021 \text{ ms}$

- Antwort 4: $t_{e2e} = 2 \cdot t_{tx,1} + 2 \cdot t_{tx,2} + 2 \cdot t_{prop}$

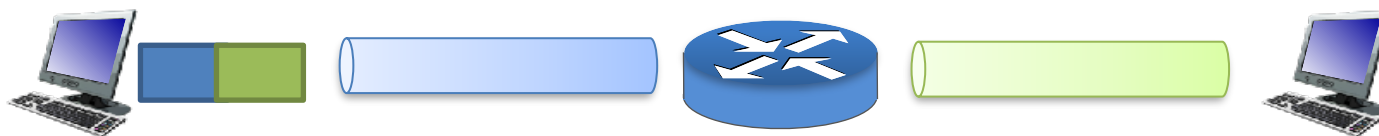


Illustration des Übertragungsvorgangs (WAN-Link)

- Situation vor dem Versenden



- Situation nach $t_{tx,1} + t_{prop}$



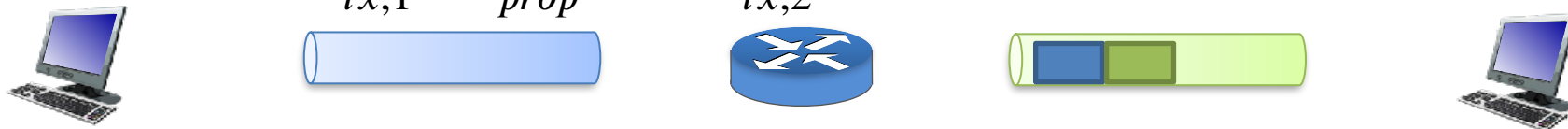
- Situation nach $2 \cdot t_{tx,1} + t_{prop}$



- Situation nach $t_{tx,1} + t_{prop} + t_{tx,2}$



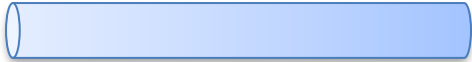
- Situation nach $t_{tx,1} + t_{prop} + 2 \cdot t_{tx,2}$



- Situation nach $t_{tx,1} + t_{prop} + 2 \cdot t_{tx,2} + t_{prop}$



Dauer der Übertragung eines Paket über einen LAN-Link

- Link 
 - Kapazität C
 - Physikalische Länge l
 - Ausbreitungsgeschwindigkeit: v

- Paket: 

- Größe L

- Wie lange dauert die Übertragung eines Paket über den Link?

- Übertragungsverzögerung: $t_{tx} = \frac{L}{C} = \frac{10 \text{ kbit}}{100 \text{ Mbps}} = \frac{10e3 \text{ bit}}{100e6 \text{ bps}} = 1 \text{ ms}$

- Ausbreitungsverzögerung: $t_{prop} = \frac{l}{v} = \frac{200 \text{ m}}{200000 \text{ km/s}} = \frac{2e2 \text{ m}}{2e8 \text{ m/s}} = 1 \mu\text{s}$

- Gesamtverzögerung: $t_{link} = t_{tx} + t_{prop} = 1 \text{ ms} + 1 \mu\text{s} = 1,001 \text{ ms}$

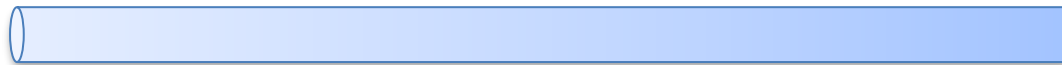
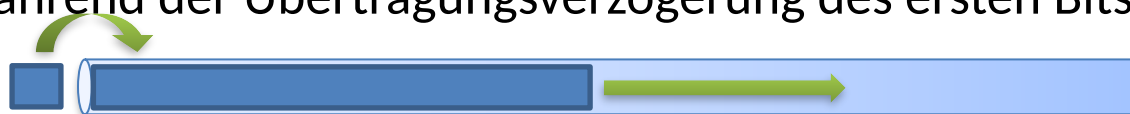


Illustration des Übertragungsvorgangs (LAN-Link)

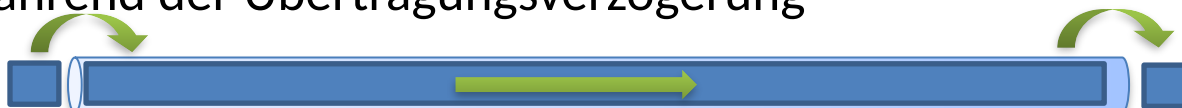
- Situation vor dem Versenden



- Situation während der Übertragungsverzögerung des ersten Bits



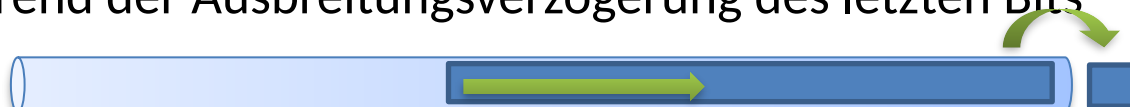
- Situation während der Übertragungsverzögerung



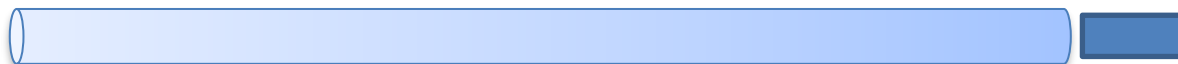
- Situation nach der Übertragungsverzögerung



- Situation während der Ausbreitungsverzögerung des letzten Bits



- Situation nach der Übertragungs- und Ausbreitungsverzögerung



Übertragung von zwei Paketen über zwei LAN Links

- 2 Pakete: 
 - Größe $L = 10kbit$

- Link 1: 

$$C = 100 \text{ Mbps}$$

$$l = 200 \text{ m}$$

$$v = 200000 \text{ km/s}$$

$$t_{tx,1} = 1 \text{ ms}$$

- Link 2: 

$$C = 10 \text{ Mbps}$$

$$l = 200 \text{ m}$$

$$v = 200000 \text{ km/s}$$

$$t_{tx,2} = 10 \text{ ms}$$

- Wie lange dauert die Übertragung von zwei Paketen?

- Antwort 1: $t_{e2e} = t_{tx,1} + t_{tx,2} + 2 \cdot t_{prop}$

- Antwort 2: $t_{e2e} = 2 \cdot t_{tx,1} + t_{tx,2} + 2 \cdot t_{prop}$

- Antwort 3: $t_{e2e} = t_{tx,1} + 2 \cdot t_{tx,2} + 2 \cdot t_{prop}$

- Antwort 4: $t_{e2e} = 2 \cdot t_{tx,1} + 2 \cdot t_{tx,2} + 2 \cdot t_{prop}$



$$t_{e2e} = 21,002 \text{ ms}$$



Illustration des Übertragungsvorgangs (LAN-Link)

- Situation vor dem Versenden



- Situation nach $t_{tx,1} + t_{prop}$



- Situation nach $2 \cdot t_{tx,1} + t_{prop}$



- Situation nach $t_{tx,1} + t_{prop} + t_{tx,2}$



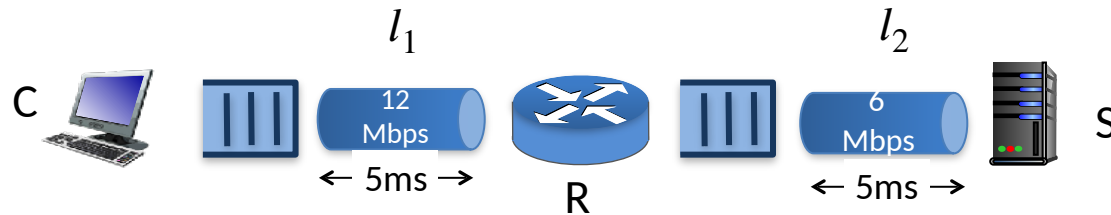
- Situation nach $t_{tx,1} + t_{prop} + 2 \cdot t_{tx,2}$



- Situation nach $t_{tx,1} + t_{prop} + 2 \cdot t_{tx,2} + t_{prop}$



Beispiel: Mehrere Pakete über eine Übertragungsstrecke



1 Paket
(1500 Bytes)



Ende-zu-Ende-
Verzögerung?

5 Pakete
(je 1500 Bytes)



Ende-zu-Ende-
Verzögerung?

n Pakete
(je 1500 Bytes)



Ende-zu-Ende-
Verzögerung?

l : Link

C_l : Linkkapazität

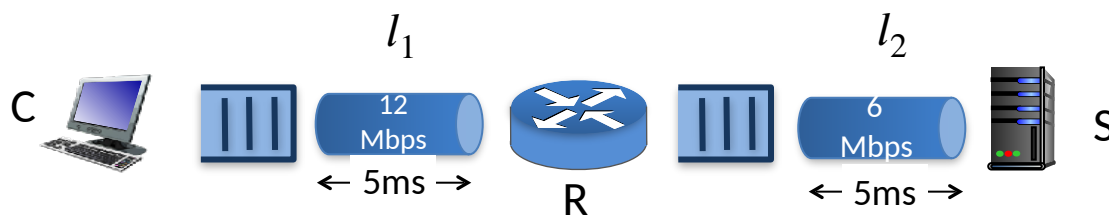
d_l : Linkdelay ($= t_{prop}$)

L : Paketgröße

$T_l(k)$: Übertragungsdauer für k Pakete über den Link l

$T_{E2E}(k)$: Ende-zu-Ende Übertragungsdauer für k Pakete

Beispiel: Übertragung von einem Paket



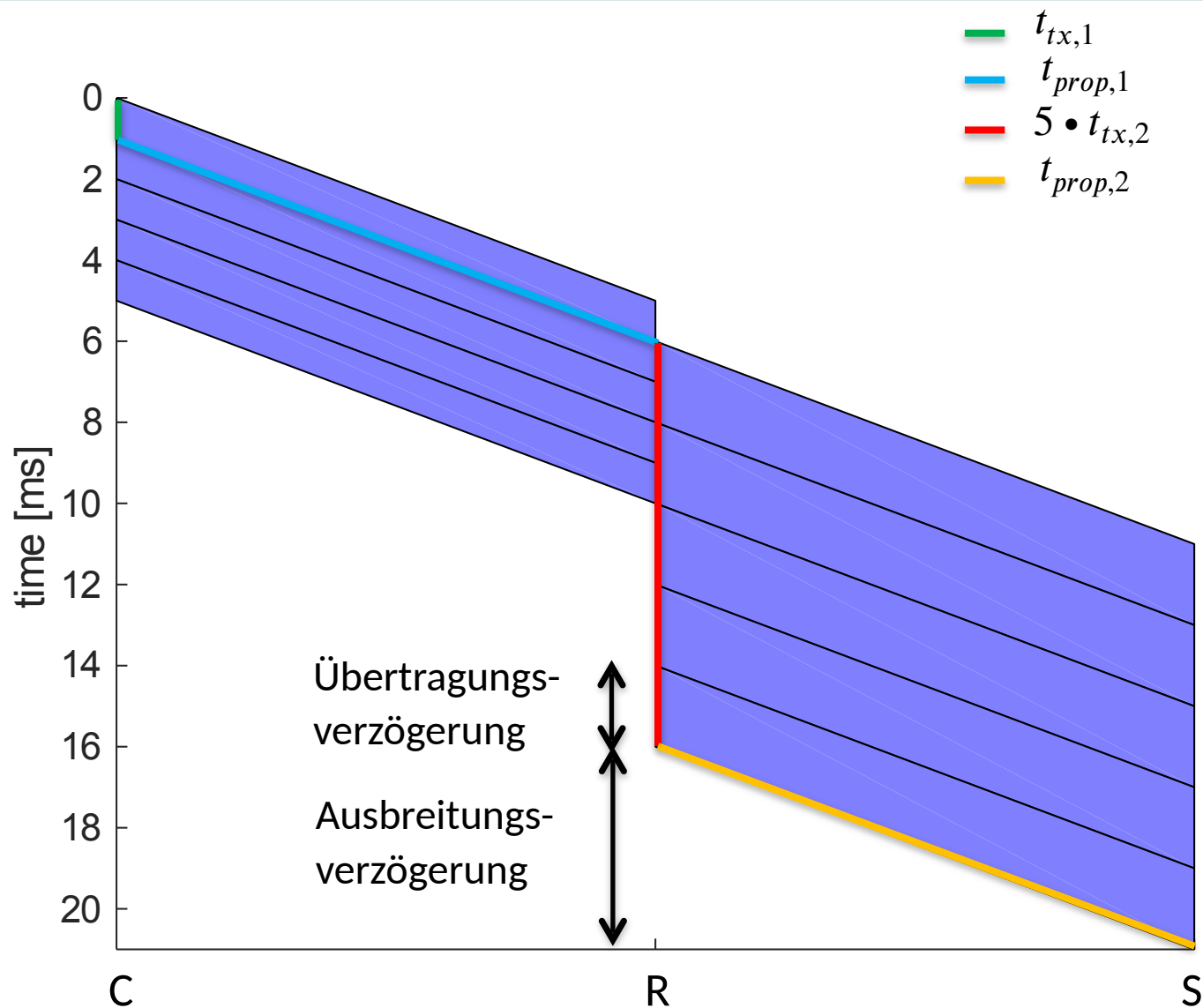
1 Paket
(1500 Bytes)



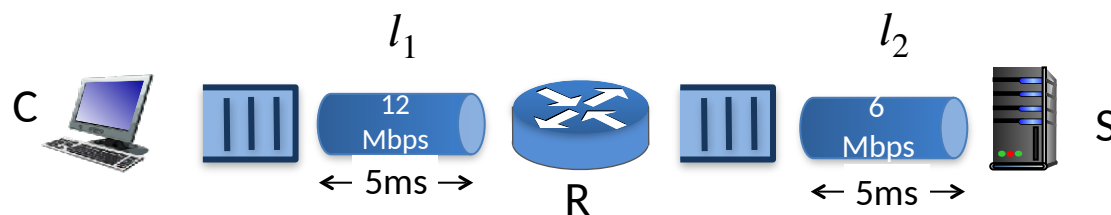
Ende-zu-Ende-
Verzögerung?

$$T_{E2E}(1) = T_{l_1}(1) + T_l(1) = \frac{12e3bit}{12Mbps} + 5ms + \frac{12e3bit}{6Mbps} + 5ms = 1ms + 5ms + 2ms + 5ms = 13ms$$

Beispiel: Übertragung von 5 Paketen



Beispiel: Übertragung von einem Block aus 5 Paketen



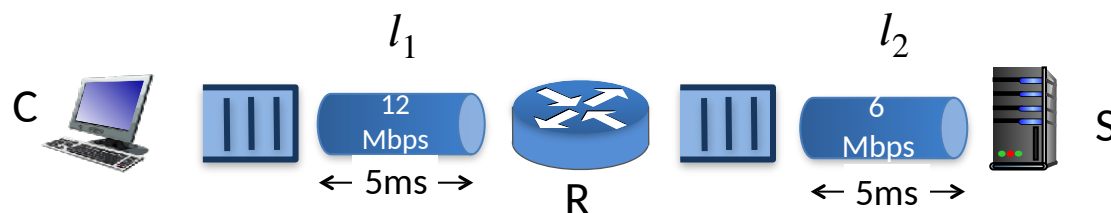
5 Pakete
(1500 Bytes)

Ende-zu-Ende-
Verzögerung?

$$T_{E2E}(5) = T_{l_1}(1) + 5 \cdot \frac{L}{C_{l_2}} + d_{l_2} = 1ms + 5ms + 10ms + 5ms = 21ms$$

1. Übertragung Paket 1 auf Link l_1 (**1ms**)
2. Übertragung aller Pakete über den Bottleneck-Link (**5 · 2ms=10ms**)
 - da der Bottleneck-Link die größte Übertragungsdauer hat, ist das zweite Datenpaket schon am Bottleneck, bevor das erste Paket fertig übertragen ist
3. Ausbreitungsverzögerung von Link l_2 (**5ms**)

Beispiel



5 Pakete
(1500 Bytes)

Ende-zu-Ende-
Verzögerung?

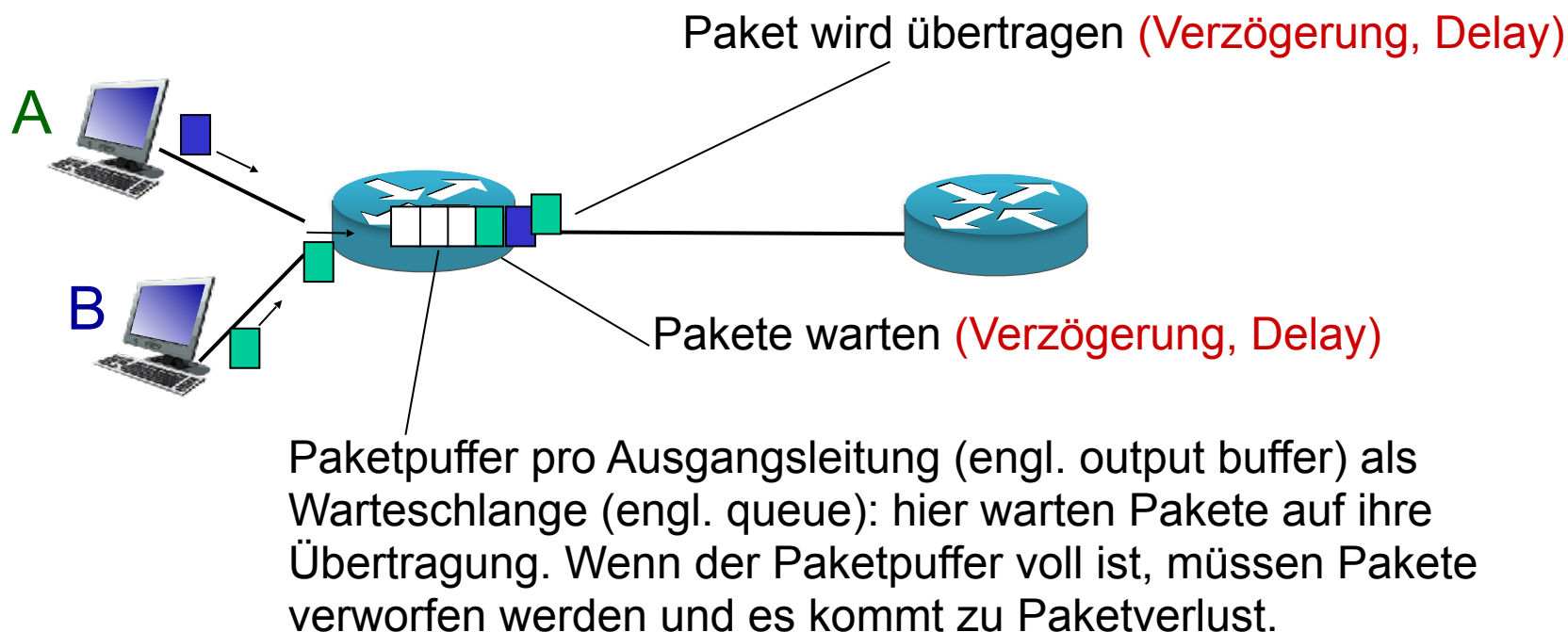
- Allgemein:

- Ende-zu-Ende Verzögerung von n Paketen ist die Ende-zu-Ende-Verzögerung für ein Paket plus die $(n - 1)$ -fache Übertragungsverzögerung eines Pakets auf dem Bottleneck-Link, d.h. dem Link mit kleinster Bandbreite

$$T_{E2E}(n) = T_{E2E}(1) + (n - 1) \frac{L}{\min(C_l)} = 13ms + 4 \cdot 2ms = 21ms$$

Puffern von Paketen im Router

Wenn Pakete schneller ankommen, als sie weitergeleitet werden können, warten sie im Paketpuffer der Ausgangsleitung.

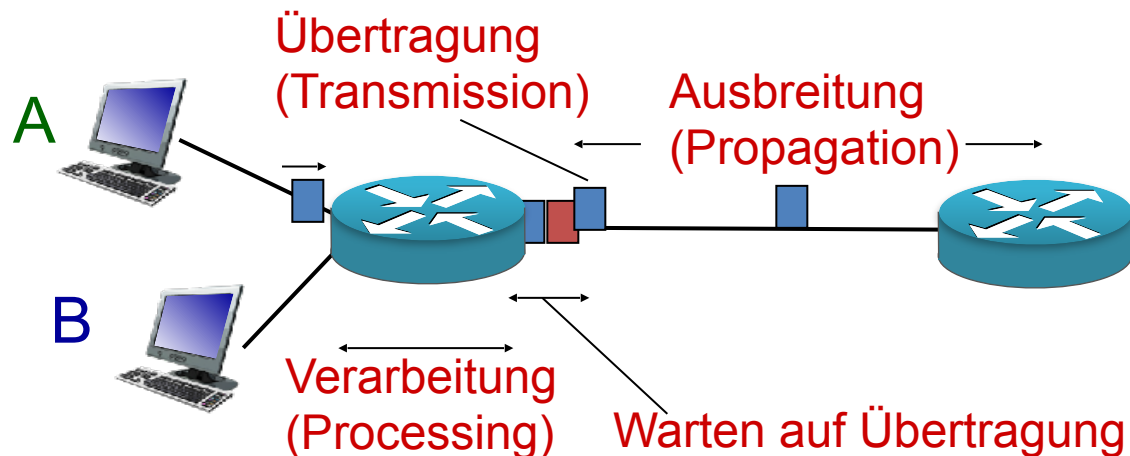


Speichern von Paketen in einem Puffer

- Wenn die Ankunftsrate der Pakete für eine Ausgangsleitung temporär die Übertragungsrate dieser Ausgangsleitung übersteigt, werden Pakete im **Puffer** der Ausgangsleitung (engl. **output buffer**) zwischengespeichert und warten darauf abgearbeitet d.h. weitergeschickt zu werden.
- Dadurch entsteht eine **Wartezeit** im Router. Die Wartezeit beträgt eine Übertragungsverzögerung pro Paket, das vorher in der **Warteschlange** eingereiht ist. Bei einer **FIFO** (First In First Out) Queue werden die Pakete in der Reihenfolge ihres Eintreffens weitergeleitet. Es gibt aber auch andere Varianten, in denen bestimmte Pakete priorisiert (Priority Queueing) werden oder jeder Verkehrsfluss die gleiche Datenrate erhält (Round Robin). Die unterschiedlichen Abarbeitungsstrategien werden als **Scheduling** bezeichnet.

- Die Puffer im Router sind begrenzt. Wenn die Ankunftsrate längerfristig die Übertragungsrate übersteigt, läuft der Puffer voll und Pakete gehen verloren. Die **Packet Drop Policy** bestimmt, wie Pakete verworfen werden
 - ankommendes Paket verwerfen
 - vorderstes Paket verwerfen oder Pakete nach einer bestimmten Wartezeit verwerfen
 - die Idee diesen Strategien ist, dass Pakete oftmals nach einer zu langen Übertragungsdauer nutzlos sind, da sie zu spät ankommen
 - Beispiel: Sprachübertragung
 - zufällig Pakete in Abhängigkeit der Warteschlangenlänge verwerfen (Random Early Discard)
 - gut für TCP

Verzögerungen in einem Knoten



$$t_{node} = t_{proc} + t_{queue} + t_{tx} + t_{prop}$$

Die Paketverzögerung in einem Knoten ist die Zeit vom Eintreffen an einem Knoten bis zur Ankunft am nächsten Knoten und beinhaltet 4 Komponenten:

- **Verarbeitungsverzögerung (Processing Delay):**
 - die Zeit zum Verarbeiten des Pakets im Knoten
 - $< 1\text{ms}$ und daher meist vernachlässigbar
- **Wartezeit (Waiting Time):** die Zeit, die das Paket wartet, bis es zum nächsten Knoten übertragen werden kann
 - abhängig von der Last/Auslastung des Links
- **Übertragungsverzögerung (Transmission Delay):** die Zeit zur Übertragung der Bits eines Pakets auf eine Leitung
 - Paketgröße/Kapazität
- **Ausbreitungsverzögerung (Propagation Delay):** die Zeit die ein Bit von einem Ende einer Leitung zum anderen braucht
 - Link-Länge/Ausbreitungsgeschwindigkeit

- Parameter DSL Downlink:
 - Übertragungsrate: 16 Mbps
 - Länge der DSL Strecke (last mile): 1600m
 - Ausbreitungsgeschwindigkeit: 200000km/s
 - Paketgröße: 1500 Bytes
 - Warteschlangenlänge bei Eintreffen des Pakets: 10
- Last-unabhängige Größen:
 - Ausbreitungsverzögerung “last mile”?
 - Übertragungsverzögerung eines Pakets?
 - Gesamt-Verzögerung (ohne Warte- und Verarbeitungsverzögerung)?
 - Wie viele Bits passen auf die DSL-Leitung?
 - Wie viele Pakete passen auf die DSL Leitung?
 - Physikalische Länge
 - eines Bits?
 - eines Pakets?
- Last-abhängige Größen
 - Wartezeit?
 - Gesamtverzögerung mit Wartezeit aber ohne Verarbeitungsverzögerung?

- Parameter DSL Downlink:
 - Übertragungsrate: 16 Mbps
 - Länge der DSL Strecke (last mile): 1600m
 - Ausbreitungsgeschwindigkeit: 200000km/s
 - Paketgröße: 1500 Bytes
- Last-unabhängige Größen:
 - Ausbreitungsverzögerung “last mile”: $1600\text{m}/2e8\text{m/s}=8\mu\text{s}$
 - Übertragungsverzögerung eines Pakets: $12e3\text{b}/16e6\text{bps}=750\mu\text{s}$
 - Gesamt-Verzögerung (ohne Warte- und Verarbeitungsverzögerung): $758\mu\text{s}$
 - Wie viele Bits passen auf die DSL-Leitung?
 - Dauer einer Bitübertragung: $750\mu\text{s}/12000=62,5\text{ns}$ ($=1/16e6\text{bps}$)
 - Anzahl Bits auf der Leitung: $8\mu\text{s}/62,5\text{ns}=128$ bits
 - Wie viele Pakete passen auf die DSL Leitung?
 $128\text{bits}/12000\text{bits}\approx 0,01$ (ein Hundertstel Paket)
 - Physikalische Länge
 - eines Bits: $62,5\text{ns}\cdot 2e8\text{m/s}=12,5\text{m}$
 - eines Pakets: $750\mu\text{s}\cdot 2e8\text{m/s}=150\text{km}$
- Last-abhängige Größen
 - Warteverzögerung bei 10 Paketen in der Warteschlange? $10\cdot 750\mu\text{s}=7,5\text{ms}$
 - Gesamtverzögerung mit Warteverzögerung aber ohne Verarbeitungsverzögerung?
 - $7,5\text{ms}+750\mu\text{s}+8\mu\text{s}=8,258\text{ ms}$

2.1 Grundlagen

2.2 Protokolle und Dienste

2.3 Sockets

2.4 Grundlagen der Datenübertragung

2.4.1 Paketvermittlung

2.4.2 Verzögerungen

2.4.3 Paketverlust

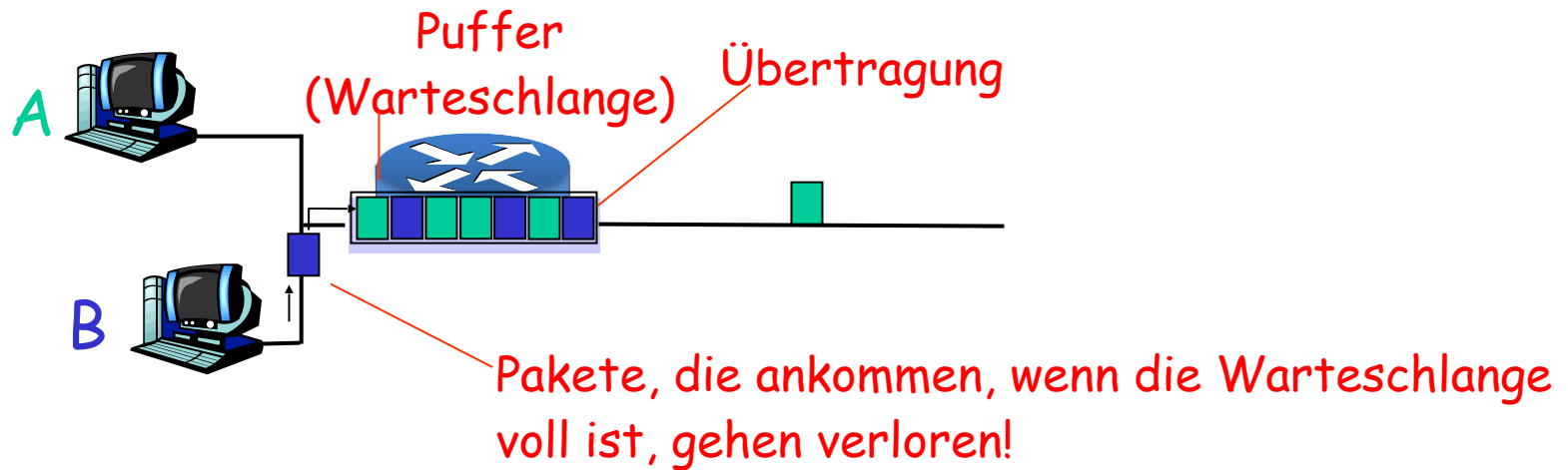
2.4.4 Durchsatz

2.4.5 Messungen im Internet

2.5 Aufbau des Internets

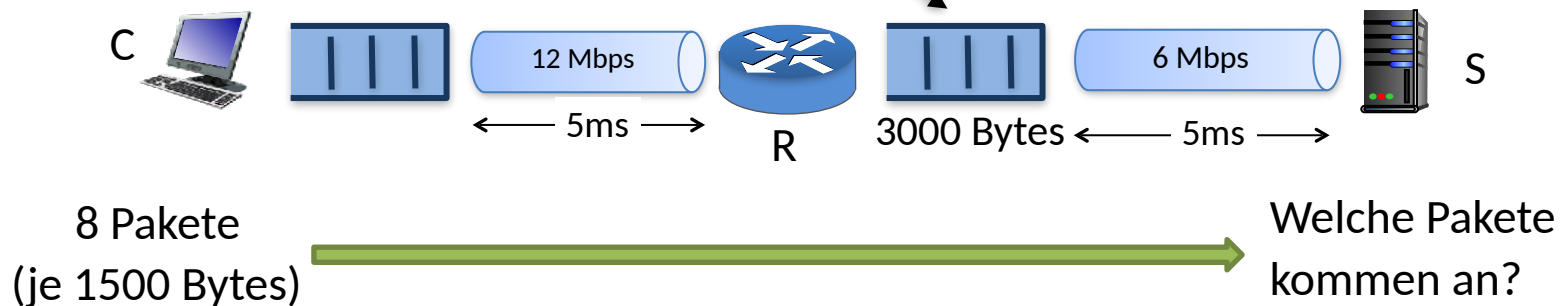
2.6 Zusammenfassung

- Warteschlange vor einer Ausgangsleitung hat endlich viele Pufferplätze
 - wenn alle Pufferplätze belegt sind, werden neu ankommende Pakete verworfen: Paketverlust
 - verlorene Pakete können vom vorangegangenen Knoten oder vom Sender erneut übertragen werden – oder auch gar nicht!



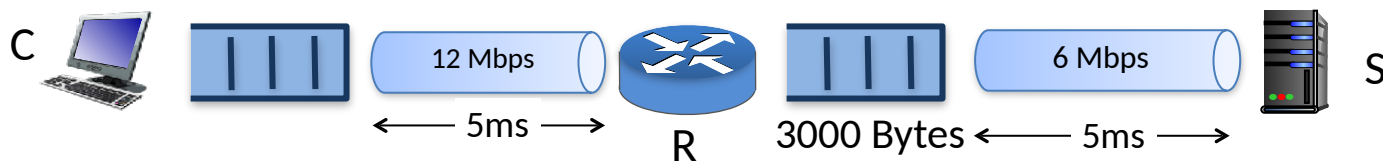
Beispiel: Pakete im Buffer vor Link 2

Zwischenankunftszeit: 1ms
Übertragungsverzögerung: 2ms
Puffer: 2 Pakete



- Ereignis-orientierte Simulation für Link 2
 - Ereignisse: Paketankünfte (An) und Paketabgänge (Dn)
 - Simulation startet mit erster Paketankunft (A1) an Link 2
 - Zustand: Pakete in Warteschlange

Beispiel: Pakete im Buffer vor Link 2



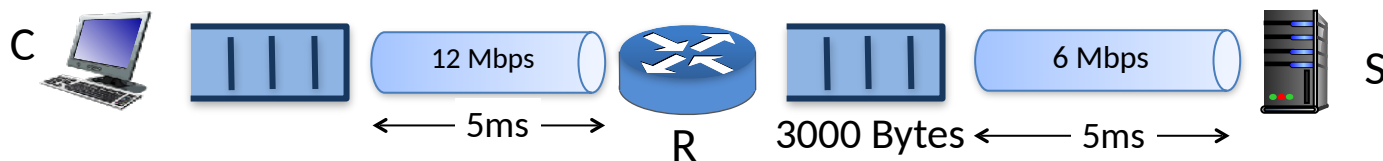
Zwischenankunftszeit: 1ms

Übertragungsverzögerung: 2ms

Puffer: 2 Pakete

Zeit	Ereignisse	Pakete im Buffer (kurz nach Ereignis)	übertragenes Paket (kurz nach Ankunft)	Ende der Übertragungsverzögerung (Paket vollständig auf dem Link)

Beispiel: Pakete im Buffer vor Link 2



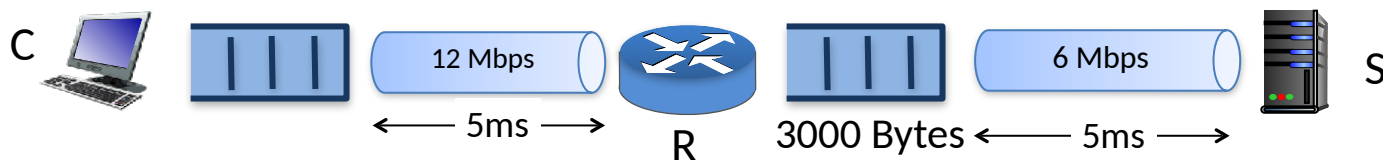
Zwischenankunftszeit: 1ms

Übertragungsverzögerung: 2ms

Puffer: 2 Pakete

Zeit	Ereignisse	Pakete im Buffer (kurz nach Ereignis)	übertragenes Paket (kurz nach Ankunft)	Ende der Übertragungsverzögerung (Paket vollständig auf dem Link)
0ms	A1		P1	2ms
1ms	A2	P2		
2ms	D1		P2	4ms
	A3	P3		
3ms	A4	P4,P3		
4ms	D2	P4	P3	6ms
	A5	P5,P4		
5ms	A6 (dropped)			
6ms	D3	P5	P4	8ms
	A7	P6,P5		
7ms	A8 (dropped)			

Beispiel: Pakete im Buffer vor Link 2



Zwischenankunftszeit: 1ms

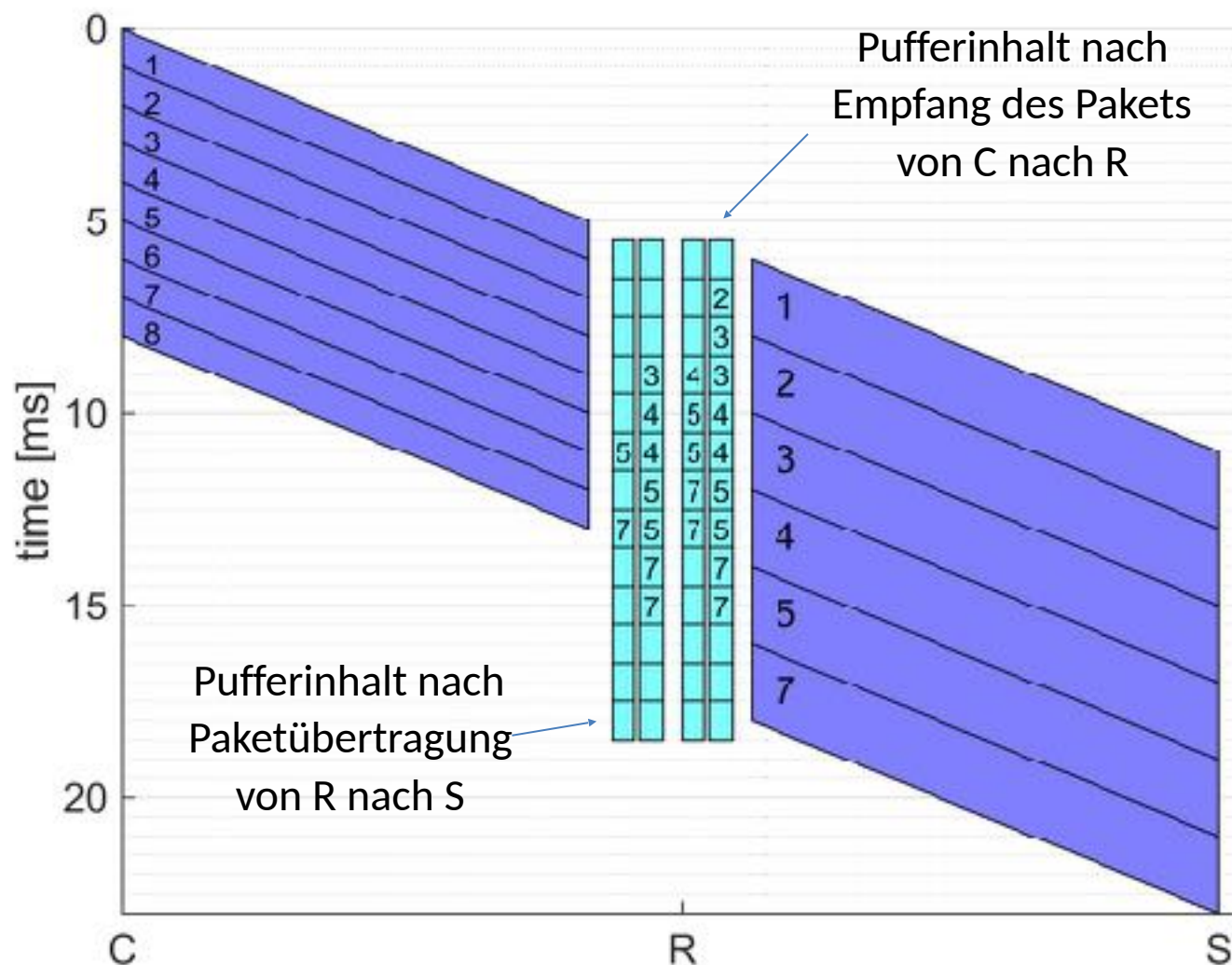
Übertragungsverzögerung: 2ms

Puffer: 2 Pakete

Zeit	Ereignisse	Pakete im Buffer nach Ereignis
0ms	A1	
1ms	A2	P1
2ms	D1	
	A3	P2
3ms	A4	P3, P4
4ms	D2	P3
	A5	P4, P5
5ms	A6 (dropped)	P5
6ms	D3	P5
	A7	P6, P5
7ms	A8 (dropped)	

- Ab dem sechsten Paket geht jedes zweite Paket verloren!!!
- Jedes zweite Paket erhöht den Puffer → P2, P4 in Puffer, P6, P8, ... gehen verloren.
- Ergebnis nur abhängig von Verhältnis Zwischenankunftszeit zu Übertragungsverzögerung und Puffergröße-
- Analyse ist nicht immer so einfach

Darstellung der Übertragung mit begrenztem Puffer

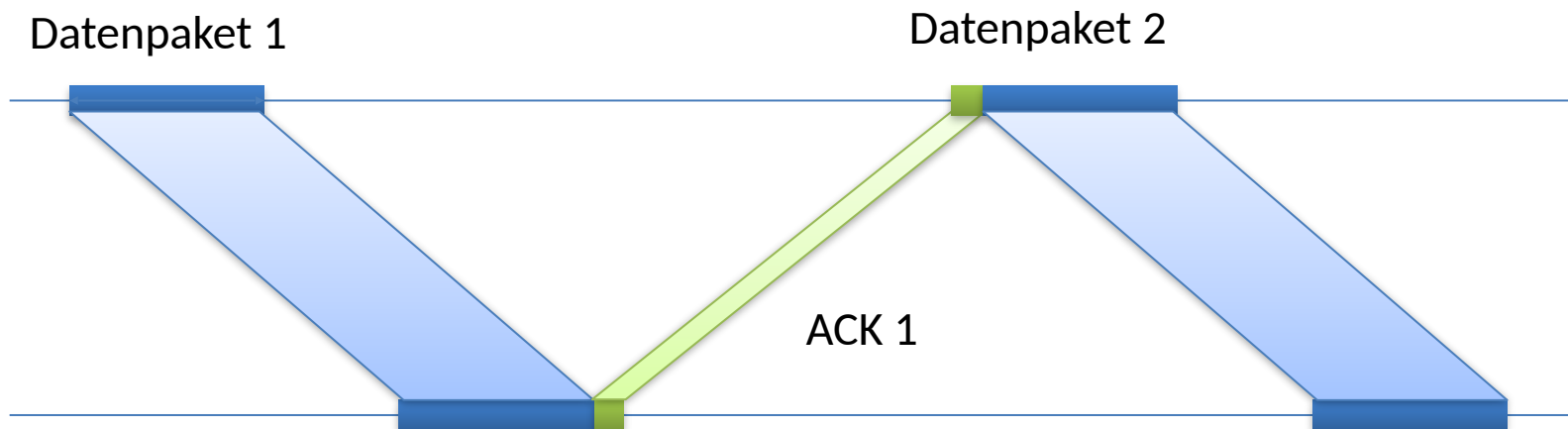


Annahme: Bei gleichzeitigem Empfang eines Pakets und Fertigstellung der Übertragung eines Pakets, wird erst das nächste Paket nach S übertragen, so dass ein Platz im Puffer frei wird.

- Ursachen für Paketverluste sind
 - überlaufende Puffer durch Überlast
 - Übertragungsfehler
- Paketverluste werden am Sender durch **ausbleibende Bestätigungen** erkannt
 - alle Pakete enthalten im **Header** eine **Sequenznummer**
 - der Empfänger bestätigt ein korrekt empfangenes Paket mit einem **Acknowledgement** (ACK), das die Sequenznummer des Pakets enthält
 - **selektive** ACKs: Bestätigung für ein spezielles Paket
 - **kumulative** ACKs: Bestätigung für alle Pakete bis inklusive der gesendeten Paketnummer
 - erhält der Sender nach einiger Zeit kein ACK für ein Paket, wird eine **Timeout** ausgelöst und das Paket als verloren gewertet
 - das Paket wird dann noch einmal übertragen, d.h. eine wiederholte Übertragung durchgeführt (engl. **Retransmission**)
 - ein einfaches Protokoll dafür ist das Send-and-Wait-Protokoll
 - auch Stop-and-Wait-Protokoll

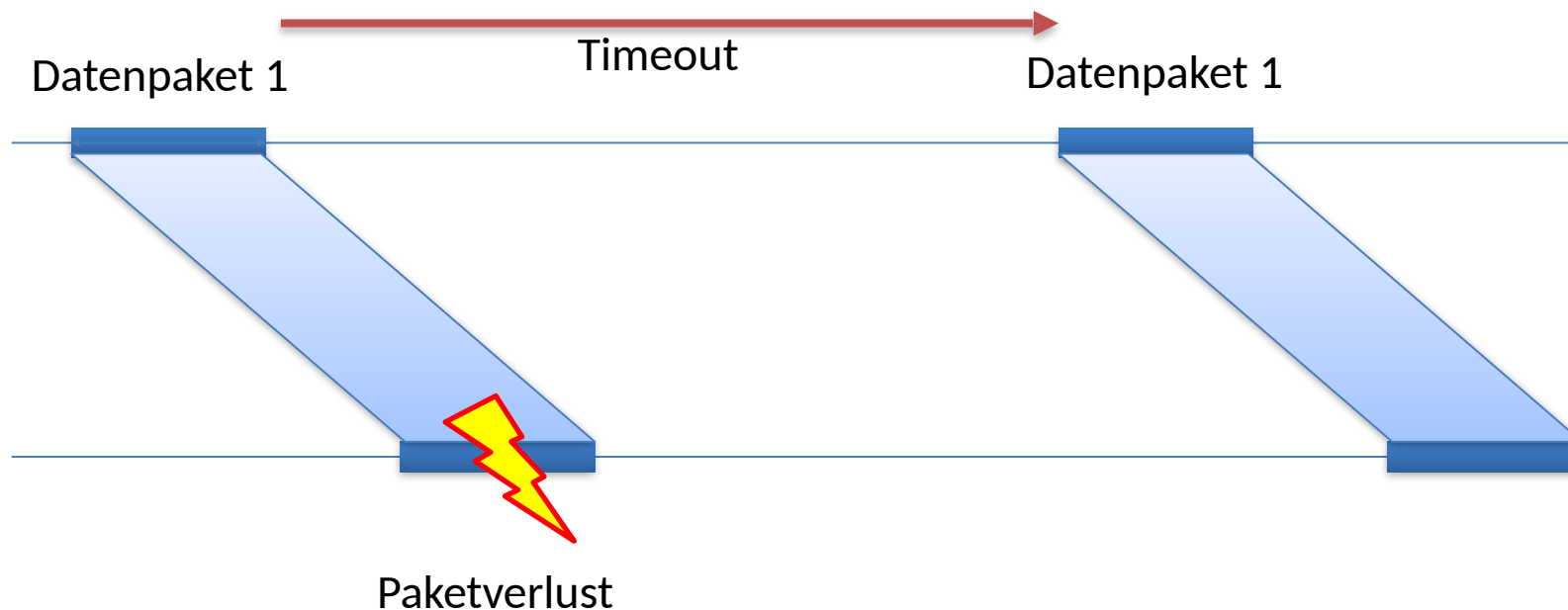
Wiederholungsaufforderung: Send-and-Wait

- Sender überträgt ein Paket und wartet auf eine Bestätigung
- nach korrektem Empfang eines Datenpakets sendet der Empfänger eine Bestätigung (Acknowledgement, ACK)
- nach Erhalt der Bestätigung überträgt der Sender das nächste Datenpaket
- falls nach einiger Zeit keine Bestätigung eintrifft (Timeout), wiederholt der Sender die Übertragung des Datenpakets

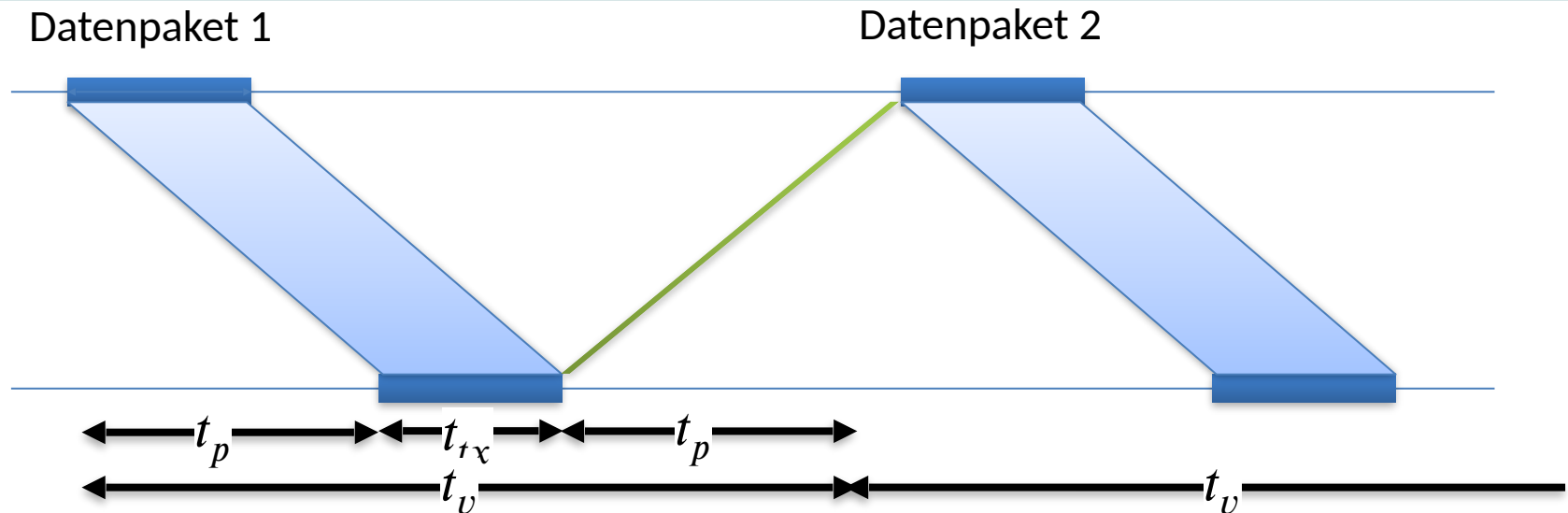


Wiederholungsaufforderung: Send-and-Wait

- Sender überträgt ein Paket und wartet auf eine Bestätigung
- nach korrektem Empfang eines Datenpakets sendet der Empfänger eine Bestätigung (Acknowledgement, ACK)
- nach Erhalt der Bestätigung überträgt der Sender das nächste Datenpaket
- falls nach einiger Zeit keine Bestätigung eintrifft (Timeout), wiederholt der Sender die Übertragung des Datenpakets

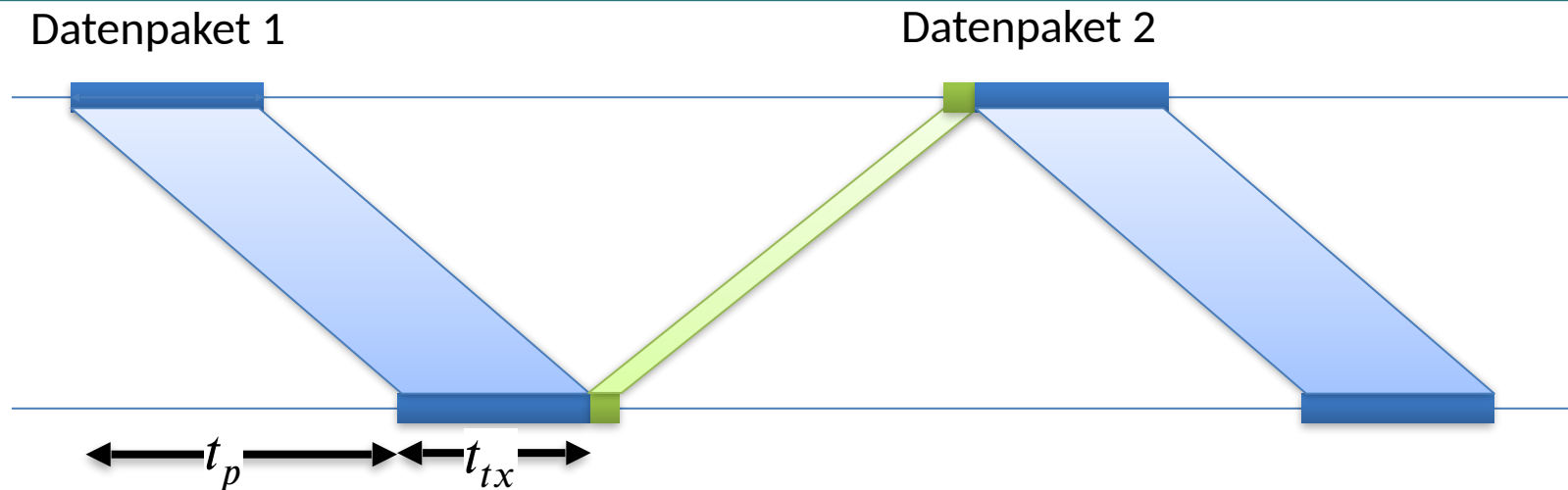


Send-and-Wait: Durchsatz und Auslastung



- Der **Durchsatz** D (engl. throughput) bezeichnet das erfolgreich übertragene Datenvolumen pro Zeit
- Abschätzung des maximalen Durchsatzes über die **virtuelle Übertragungszeit**, d.h. die Zeit, die eine Paket die Leitung virtuell belegt
 - „Virtuelle“ Übertragungszeit für ein Datenpaket (ohne Fehler): $t_v = t_{tx} + 2t_p$
 - Maximaler Durchsatz: $D_{max} = \frac{1}{t_v} = \frac{1}{t_{tx} + 2t_p}$ in Paketen / s
- Die **Auslastung** ρ einer Leitung bezeichnet den Anteil der Zeit, zu dem über die Leitung effektiv Daten übertragen werden.
 - Auslastung: $\rho = \frac{t_{tx}}{t_{tx} + 2t_p}$

Send-and-Wait: Beispiele



- Beispiele:
 - Paketgröße: 1 kbit
 - Übertragungskapazität: 64 kbps
 - Ausbreitungsverzögerung
 - LAN: 0,1 ms
 - Satellit: 250 ms
 - Wie groß ist der maximale Durchsatz? Wie groß ist die Auslastung?
 - Ist das einfache Send-and-Wait-Protokoll für beide Strecken geeignet?

Send-and-Wait: Beispiele

- Übertragungsverzögerung $t_{tx} = \frac{1 \text{ kbit}}{64 \text{ kbps}} \approx 0,016 \text{ s} = 16 \text{ ms}$
- LAN:
 - Durchsatz: $D_{max} = \frac{1}{t_{tx} + 2t_p} = \frac{1}{16,2 \text{ ms}} \approx 62 \frac{\text{Pakete}}{\text{s}}$
 - Auslastung: $\rho = \frac{t_{tx}}{t_{tx} + 2t_p} = \frac{16 \text{ ms}}{16,2 \text{ ms}} \approx 0,98 = 98\%$
- Satellit:
 - Durchsatz: $D_{max} = \frac{1}{t_{tx} + 2t_p} = \frac{1}{516 \text{ ms}} \approx 2 \frac{\text{Pakete}}{\text{s}}$
 - Auslastung: $\rho = \frac{t_{tx}}{t_{tx} + 2t_p} = \frac{16 \text{ ms}}{516 \text{ ms}} \approx 0,03 = 3\%$
- Send-and-Wait ist für Übertragungsstrecken mit hoher Ausbreitungsverzögerung oder großer Übertragungskapazität ungeeignet
 - andere Verfahren zur Datenflusssteuerung in Kapitel 4 Transportschicht

2.1 Grundlagen

2.2 Protokolle und Dienste

2.3 Sockets

2.4 Grundlagen der Datenübertragung

2.4.1 Paketvermittlung

2.4.2 Verzögerungen

2.4.3 Paketverlust

2.4.4 Durchsatz

2.2.4.1 Ende-zu-Ende-Durchsatz

2.2.4.2 Max-Min-Fairness

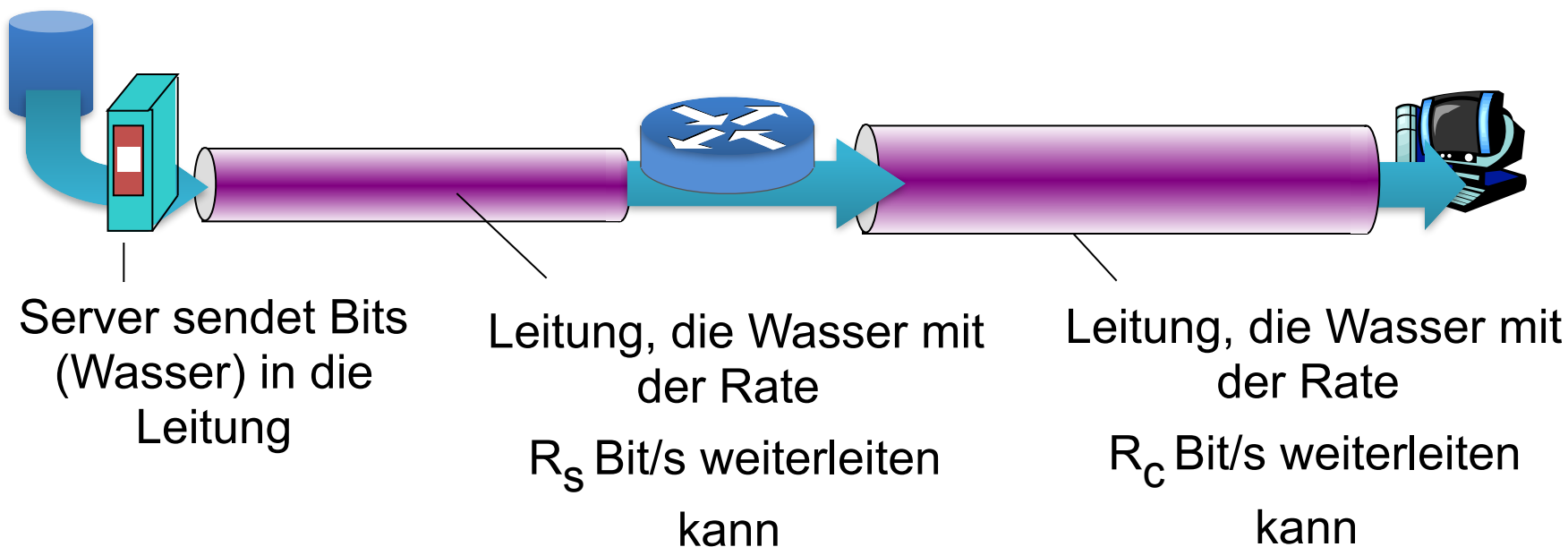
2.4.5 Messungen im Internet

2.5 Aufbau des Internets

2.6 Zusammenfassung

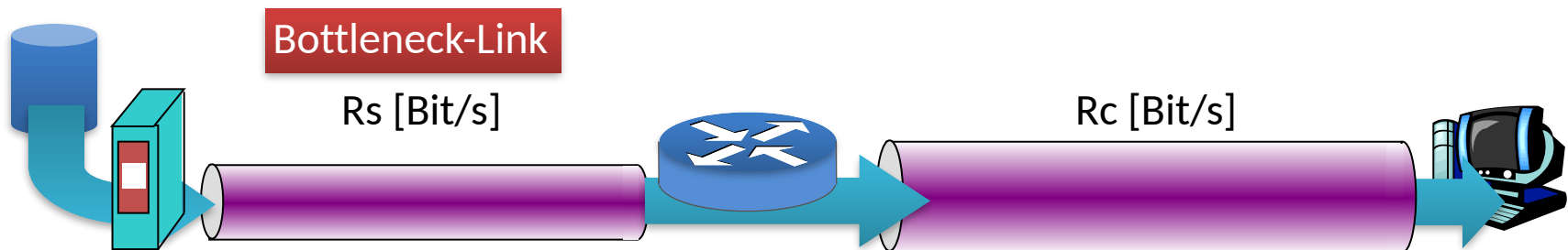
Durchsatz

- Durchsatz eines Links: Daten, die über einen Link transportiert werden. Der Durchsatz eines Links ist immer kleiner als die Kapazität des Links und der Durchsatz ist auch immer kleiner als der ankommende Verkehr
- Ende-zu-Ende-Durchsatz eines Verkehrsflusses: Rate (Bit/Zeiteinheit), mit der Daten zwischen Sender und Empfänger ausgetauscht werden.
 - Analogie: Wasserleitung

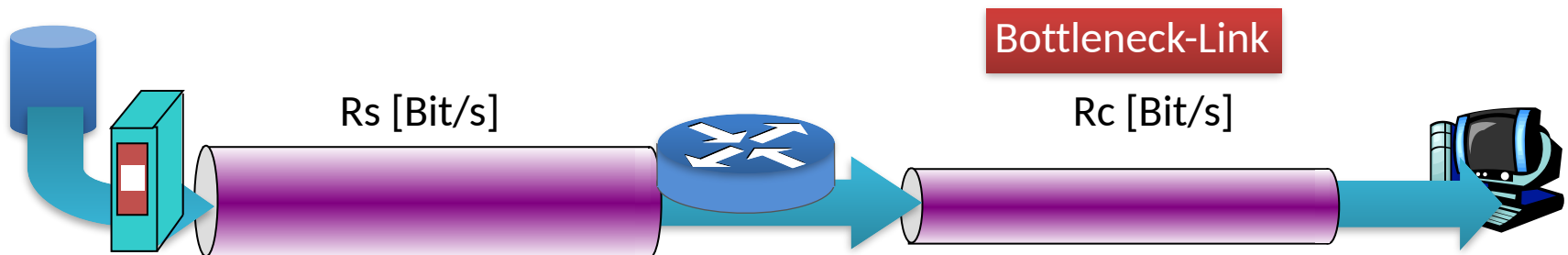


Durchsatz

- Die Leitung auf dem Ende-zu-Ende-Pfad, welche den Ende-zu-Ende-Durchsatz begrenzt, bildet den Engpass und wird als Flaschenhals (Bottleneck) bezeichnet. Der Bottleneck-Link bestimmt den Ende-zu-Ende Durchsatz.
- $R_s < R_c$: Was ist der durchschnittliche Ende-zu-Ende-Durchsatz?

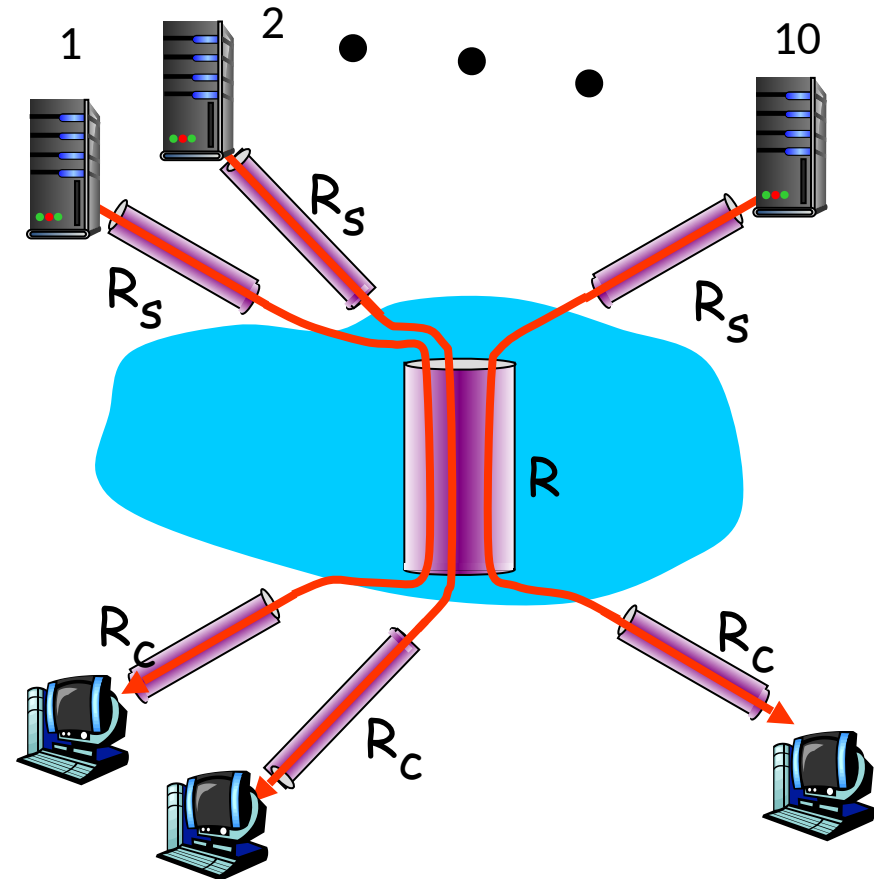


- $R_s > R_c$: Was ist der durchschnittliche Ende-zu-Ende-Durchsatz?



Durchsatz im Internet

- Durchsatz für Ende-zu-Ende-Verbindungen: $\min(R_c, R_s, R/10)$
- In der Realität: Häufig sind R_c oder R_s die Engpässe
- Anmerkung: Ein faires Aufteilen der Bottleneck-Bandbreite trifft nur zu, wenn es sich um Verbindungen handelt, die in jedweder Hinsicht identisch sind (Protokoll, Ausbreitungsverzögerung, Paketgröße, Charakteristik des erzeugten Verkehrs, etc.)



10 Verbindungen teilen sich fair
die Rate R des Bottleneck- Links
im Backbone

- Wir unterscheiden zwei Arten von „fairen“ Verkehrsflüssen
 - adaptive **saturierte Verkehrsflüsse**: die Quelle hat unbegrenzt Daten zu senden und sendet diese mit der Bottleneckrate
 - Verkehrsflüsse mit **fester Datenrate**: die Quelle sendet Pakete mit einer bestimmten Rate
 - ist die Datenrate kleiner als die Bottleneckrate, dann kann der Fluss seine faire Bottleneckrate nicht voll ausschöpfen und der ungenutzte Anteil wird fair unter den anderen Flüssen aufgeteilt
 - ist die Datenrate größer als die Bottleneckrate, dann geht ein Teil der Pakete verloren

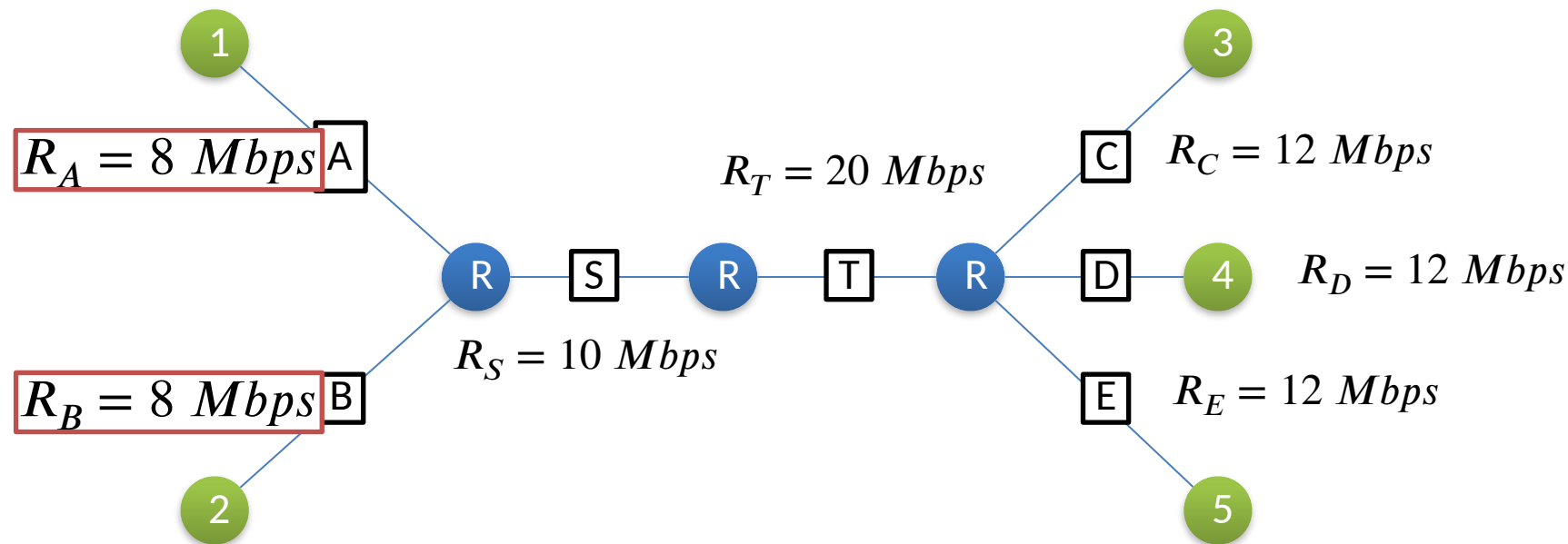
- Der Ende-zu-Ende-Durchsatz D eines Flusses v ist das Minimum aus der Datenrate S_v , mit der die Quelle sendet und der Bottleneckdatenrate B_v
- Die Bottleneckdatenrate B_v ist die kleinste Rate pro Verkehrsfluss R_l für einen Link l , der auf dem Pfad P_v des Verkehrsflusses liegt

$$B_v = \min_{l \in P_v} R_l = \min_{l \in P_v} \frac{C_l}{N_l}$$

- dabei ist C_l die Kapazität des Links und N_l die Anzahl der Flüsse, die durch diesen Link laufen
- diese Berechnung ist nur korrekt, wenn für alle diese N_l Flüsse Link l der Bottlenecklink ist und die Datenrate S_v mindestens so groß wie die berechnete Bottleneckdatenrate ist

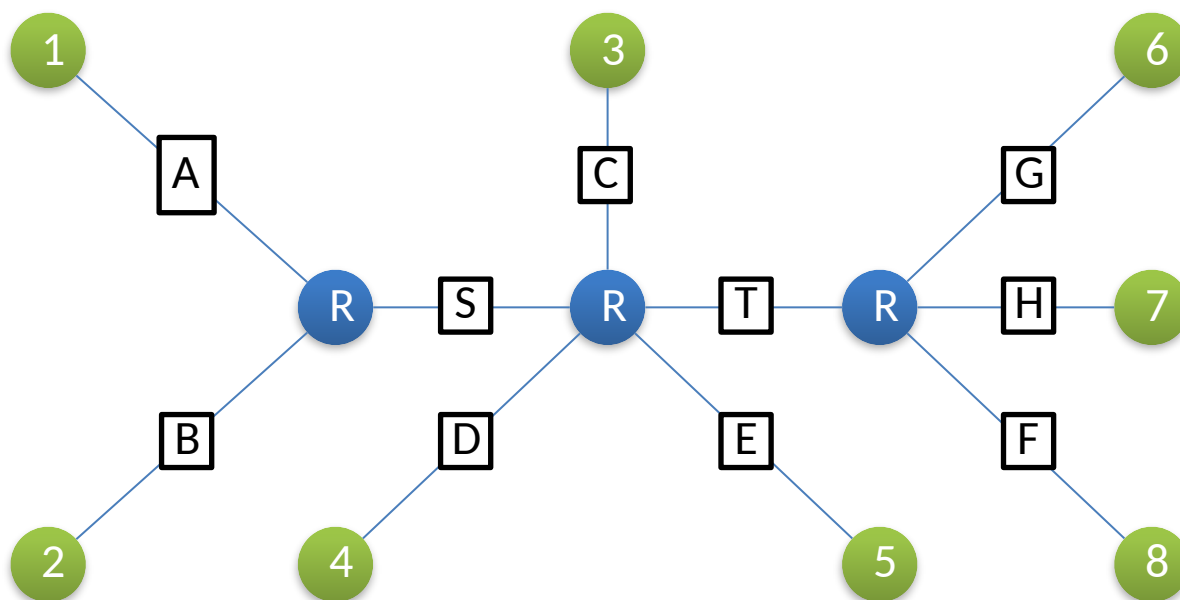
Beispiel: Ende-zu-Ende Durchsatz

- 6 Verkehrsflüsse:
 - Hosts 1 und 2 kommunizieren jeweils mit Hosts 3, 4 und 5
- 7 Links:
 - Links A-E mit jeweils 24 Mbps
 - Links S mit 60 Mbps
 - Link T mit 120 Mbps
- 3 Router, die nur Verkehr weiterleiten
- Wie groß sind die Ende-zu-Ende Durchsätze der Flüsse?



Beispiel: Ende-zu-Ende Durchsatz

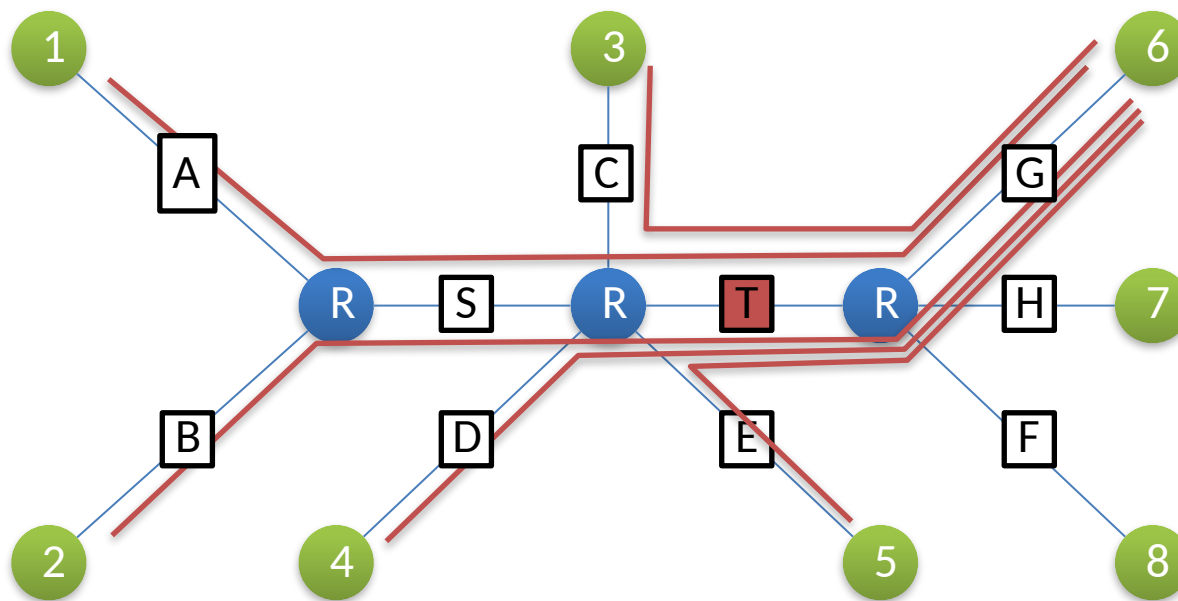
- 8 Hosts, die alle miteinander kommunizieren
 - es gibt 21 saturierte Verkehrsflüsse
- 10 Links A-H sowie S und T mit jeweils 100 Mbps
- 3 Router, die nur Verkehr weiterleiten
- Wie groß sind die Ende-zu-Ende Durchsätze der Flüsse?



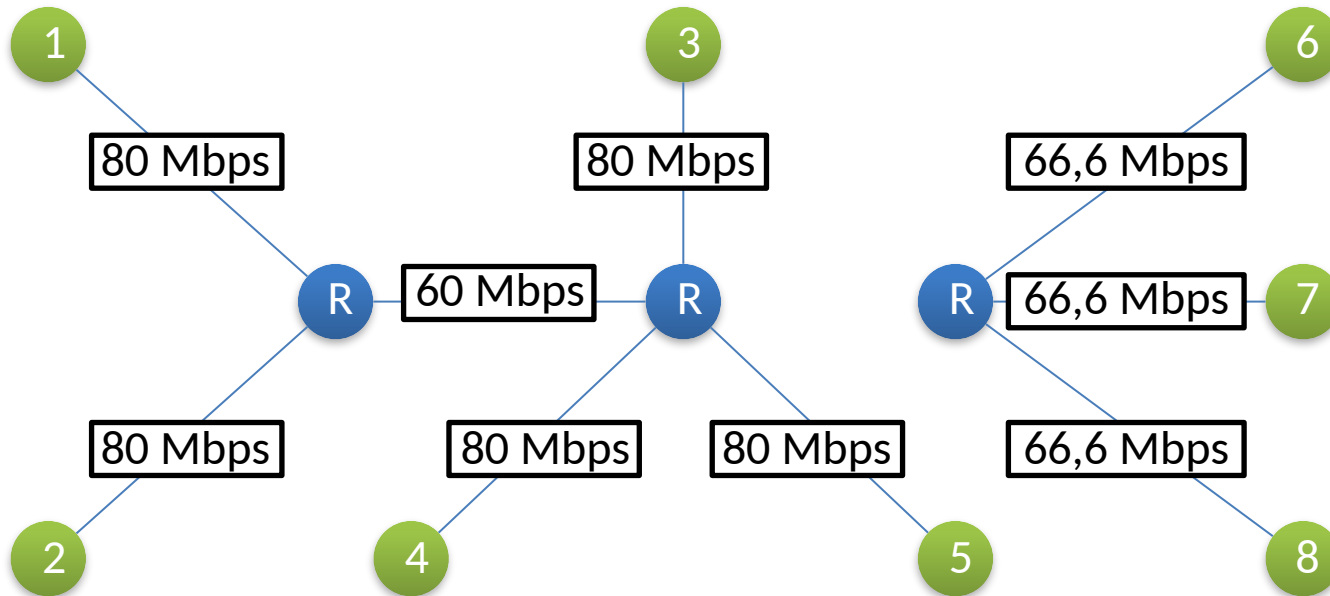
- In einem Netz mit zahlreichen Ende-zu-Ende Flüssen, können die Flüsse verschiedene Bottlenecks haben. In diesem Fall wird der Bottleneck nach dem Prinzip der Max-Min-Fairness bestimmt
 - die Kapazität wird so verteilt, dass der minimale Ende-zu-Ende-Durchsatz maximiert wird
- Algorithmus:
 - (1) bestimme für alle Links l die Anzahl N_l von Flüssen durch den Link
 - (2) bestimme für jeden Link die Rate pro Fluss $R_l = C_l / N_l$
 - (3) bestimme den Bottleneck-Link l^* mit minimaler Rate pro Fluss
 - (4) für alle Flüsse v durch den Bottleneck-Link l^*
 - setze den Ende-zu-Ende Durchsatz auf $D_v = R_{l^*}$
 - für alle Links $l \in P_v$, die von dem Fluss v durchlaufen werden, setze
 - die verbleibende Kapazität auf $C_l = C_l - R_{l^*}$
 - die verbleibenden Flüsse auf $N_l = N_l - 1$
 - (5) brich ab, wenn für alle Flüsse der Ende-zu-Ende-Durchsatz bestimmt wurde, sonst weiter mit (2)

Link															
A															
B															
C															
D															
E															
F															
G															
H															
S															
T															

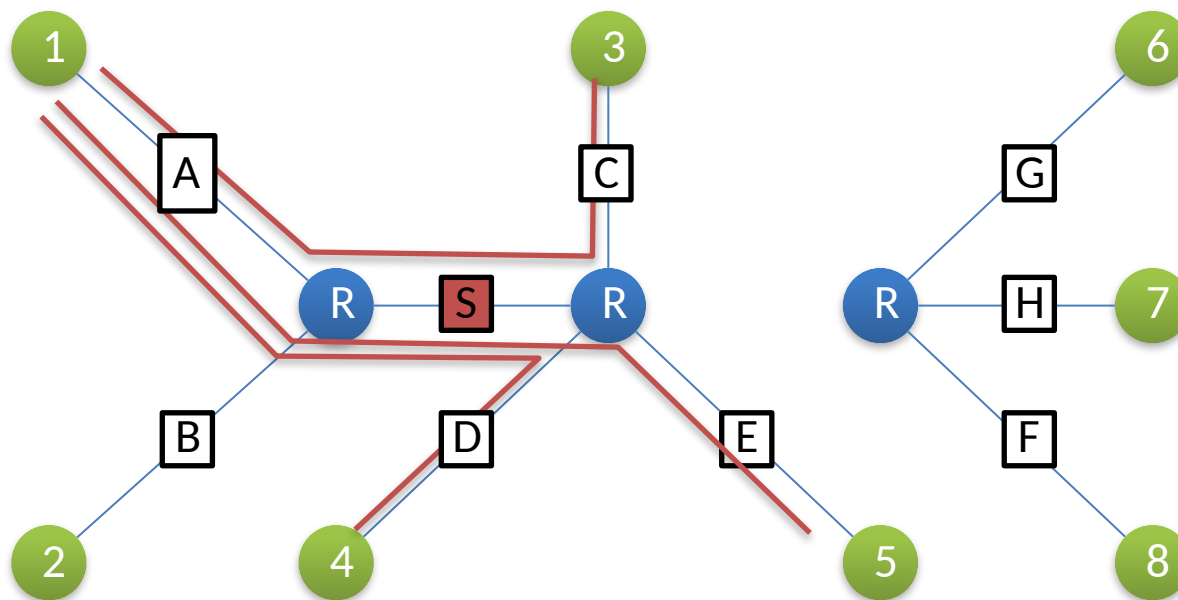
- Bottleneck: T
- 15 Flüsse durch T: $\{1,2,3,4,5\} \rightarrow \{6,7,8\}$
- Durchsatz (Ende-zu-Ende Datenrate) pro Fluss: $6, \bar{6}$ Mbps
- Exemplarisch: 5 Flüsse $\{1,2,3,4,5\} \rightarrow 6$ durch T



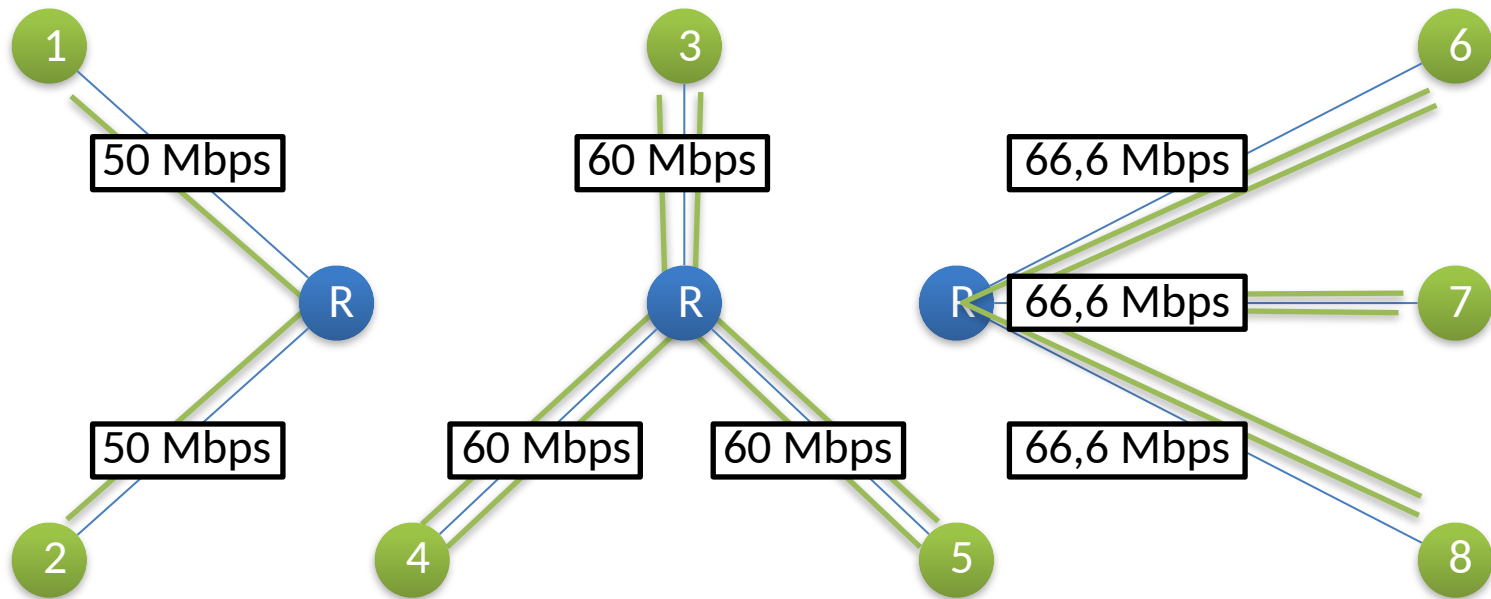
- Netz nach Schritt 1 ohne Link T und ohne Flüsse durch Link T
- Verbleibende Flüsse:
 - 12 Flüsse zwischen Knoten 1-5
 - 3 Flüsse zwischen Knoten 6-8
- Verbleibende Kapazitäten



- Bottleneck: S
- 6 Flüsse durch S: $\{1,2\} \rightarrow \{3,4,5\}$
- Durchsatz (Ende-zu-Ende Datenrate) pro Fluss: 10 Mbps
- Exemplarisch: 3 Flüsse $1 \rightarrow \{3,4,5\}$ durch S



- Netz nach Schritt 2 ohne Links T und S und ohne Flüsse durch Links T oder S
- Verbleibende Flüsse:
 - 1 Fluss zwischen Knoten 1 und 2
 - 3 Flüsse zwischen Knoten 3-4
 - 3 Flüsse zwischen Knoten 6-8
- Verbleibende Flüsse und Links sowie deren Kapazitäten



2.1 Grundlagen

2.2 Protokolle und Dienste

2.3 Sockets

2.4 Grundlagen der Datenübertragung

2.4.1 Paketvermittlung

2.4.2 Verzögerungen

2.4.3 Paketverlust

2.4.4 Durchsatz

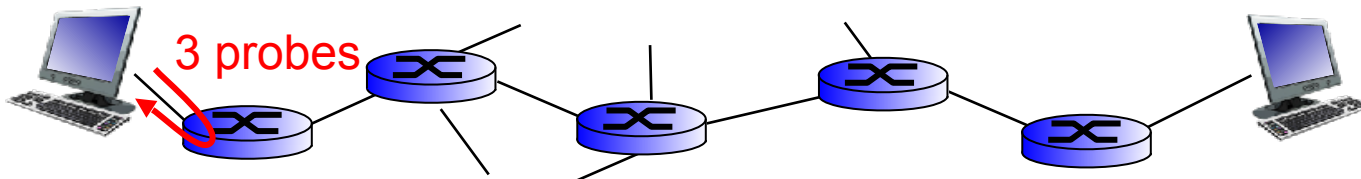
2.4.5 Messungen im Internet

2.5 Aufbau des Internets

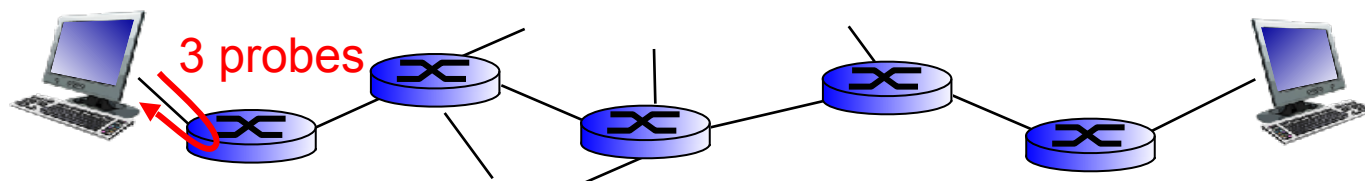
2.6 Zusammenfassung

Delay-Messungen im Internet

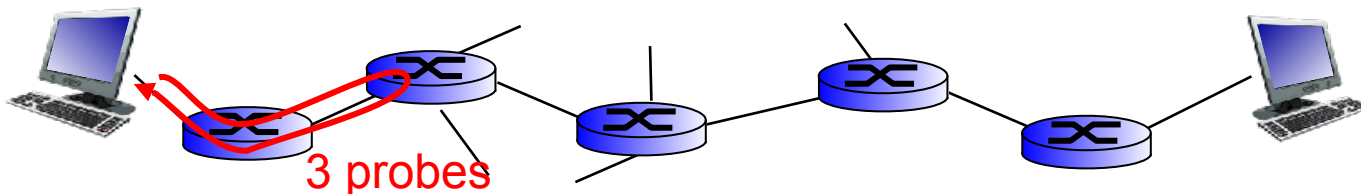
- Wie können wir messen, welche Verzögerung wir zu einem Host im Internet haben und wo die Verzögerungen auftreten?
- **Ping** schickt ein Paket zu einem Rechner, der mit einem Paket antwortet. Wenn das Paket eintrifft, kann die Round-Trip-Time (RTT) oder auch Ping-Dauer bestimmt werden. Diese Pakete sind im Protokoll ICMP (Internet Control Message Protocol) festgelegt.
- **TraceRoute** ermittelt die RTTs zu allen Routern auf dem Ende-zu-Ende Pfad. Der Sender schickt drei Ping-Pakete an den Zielrechner mit aufsteigender TTL (Time-To-Live). Erhält ein Router ein Ping-Paket mit TTL=1, so schickt er das Ping-Paket nicht weiter sondern antwortet direkt dem Sender.



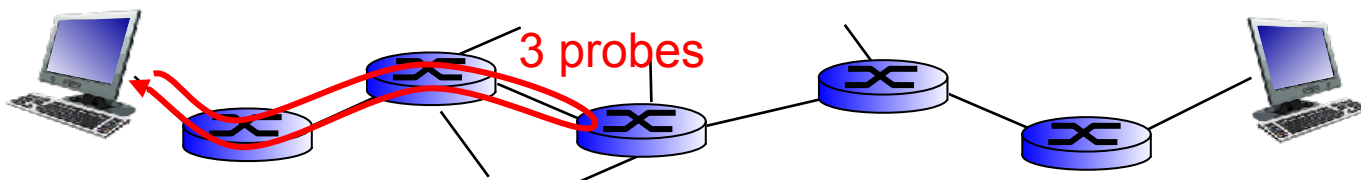
- Schritt 1: senden 3 Ping-Pakete (Probes) mit TTL=1; 1. Router antwortet



- Schritt 2: senden 3 Ping-Pakete (Probes) mit TTL=2; 2. Router antwortet



- Schritt 3: senden 3 Ping-Pakete (Probes) mit TTL=3; 3. Router antwortet



TraceRoute zu Uni Würzburg

```
C:\Users\dstaehle>tracert www.uni-wuerzburg.de

Routenverfolgung zu 132.187.1.114 [132.187.1.114] über maximal 30 Abschnitte:

  1      7 ms     9 ms     8 ms     konstanz-p3-1-10ge-0-1-0-1.belwue.net [129.143.47.77]
  2      5 ms     8 ms     9 ms     konstanz-bib-1-10ge-0-0-0-0.belwue.net [129.143.58.54]
  3      4 ms     9 ms     9 ms     tuebingen-nw3-1-10ge-0-1-0-1.belwue.net [129.143.57.185]
  4      5 ms     8 ms     9 ms     stuttgart-nw3-1-10ge-0-1-0-1.belwue.net [129.143.57.89]
  5     15 ms    18 ms    15 ms     de-bell-1-0-7-0-14.x-wm.dfn.de [188.1.233.229]
  6     12 ms    16 ms    16 ms     x-wue1-1-2-2.x-wm.dfn.de [188.1.145.157]
  7      *      *      *      Zeitüberschreitung der Anforderung.
  8      *      *      *      Zeitüberschreitung der Anforderung.
  9      *      *      *      Zeitüberschreitung der Anforderung.
 10      *      *      *      Zeitüberschreitung der Anforderung.
```



TraceRoute cs.umass.edu (Kurose)

```
C:\Users\dstaehle>tracert cs.umass.edu
```

```
Routenverfolgung zu cs.umass.edu [128.119.240.93] über maximal 30 Abschnitte:
```

Stufe	IP-Adresse	Routername	AS-Name	IP-Adresse			
1	5	NE	9	NE	konstanz-e3-1-10ge-0-1-0-1.belwue.net [129.143.47.77]		
2	4	NE	9	NE	tuebingen-was-1-10ge-0-1-0-0.belwue.net [129.143.59.73]		
3	4	NE	9	NE	stuttgart-al30-1-10ge-0-2-0-2.belwue.net [129.143.57.85]		
4	9	NE	9	NE	stuttgart-nw3-1-10ge-0-0-0-0.belwue.net [129.143.57.2]		
5	15	NE	15	NE	ce-bell-100-7-0-14.x-win.dfn.de [188.1.233.229]		
6	20	NE	23	NE	24	NE	ce-tubi-100-1-0-0-7.x-win.dfn.de [188.1.144.185]
7	29	NE	33	NE	34	NE	dfn.ix1.asn.nl.giant.net [62.40.112.145]
8	107	NE	112	NE	108	NE	xe-0-3-0.102.etc.newy32adv.net.internet2.edu [198.71.45.236]
9	109	NE	112	NE	113	NE	mux300gw1-v1-110-mux-i2.mux.uy [192.5.89.221]
10	110	NE	114	NE	114	NE	mux300gw1-peer-mux-umass-192-5-89-102.mux.uy [192.5.89.102]
11	111	NE	115	NE	115	NE	core1-et-xe-0-0-0.gw.umass.edu [192.80.83.101]
12	111	NE	114	NE	115	NE	lync-et-106-8-pu-10.gw.umass.edu [128.119.0.233]
13	115	NE	114	NE	115	NE	128.119.3.32
14	123	NE	127	NE	126	NE	luki.us.umass.edu [128.119.240.93]

```
Ablaufverfolgung beendet.
```



Wie finde ich mehr über die Route raus?

- Es gibt Seiten im Internet, die das AS (Autonomous System) siehe Kapitel Routing) zu einer IP-Adresse verraten und auch zu welchem ISPs dieses AS gehört
 - Beispiel: <https://www.ultratools.com/tools/asnInfo>
- ISPs bieten im Internet den Dienst an, Ping oder TraceRoute von ihren Routern auszuführen
 - Beispiel: lg.he.net
- ISPs bieten oft Karten mit ihrer Netztopologie an
 - Hurricane Electric: <http://he.net/HurricaneElectricNetworkMap.pdf>
 - BelWue: <http://www.belwue.de/typo3temp/pics/6e95b882e5.png>
 - DFN: https://www.dfn.de/fileadmin/1Dienstleistungen/XWIN/Netzentwurf_20140110_Seite1.pdf

- Durchsatz-Messungen werden durchgeführt, in dem versucht wird, zwischen einem Sender und einem Empfänger möglichst viel Verkehr zu senden
 - UDP-Verkehr mit fester Datenrate:
 - geeignet um die minimale Link-Kapazität zu ermitteln
 - verdrängt langfristig TCP-Verkehr auf dem Bottleneck-Link
 - großes Datenvolumen über TCP:
 - geeignet, um die Bottleneck-Link Datenrate für eine TCP Verbindung zu ermitteln
 - die Kapazität des Bottleneck abzüglich des UDP Verkehrs wird ansatzweise fair unter (langlebigen) TCP Verbindungen aufgeteilt
- Tools:
 - iPerf: einfaches Command-Line Tool
 - kann als Client und Server installiert werden und TCP- und UDP-Verkehr generieren
 - es gibt iPerf-Server im Internet
 - aktuelle Version ist iPerf3
 - GUI (jPerf) existiert für iPerf2
 - DSL-Speed-Testseiten:
 - Transfer von Datenblöcken unterschiedlicher Größe über TCP zwischen PC und Server

2.1 Grundlagen

2.2 Protokolle und Dienste

2.3 Sockets

2.4 Grundlagen der Datenübertragung

2.5 Aufbau des Internets

2.6 Zusammenfassung