

## 2.1 Terminologie

## 2.2 Grundlagen der Datenübertragung

### 2.2.1 Datenübertragung in Paketen

2.2.1.1 Übertragung eines Pakets über einen Link

2.2.1.2 Übertragung eines Packet-Bursts über eine Übertragungsstrecke

2.2.1.3 Paketvermittlung

2.2.1.4 Verzögerungen in einem Netzknoten

2.2.2 Paketverluste

2.2.3 Durchsatz

2.2.4 Messungen im Internet

## 2.3 Protokolle und Dienste

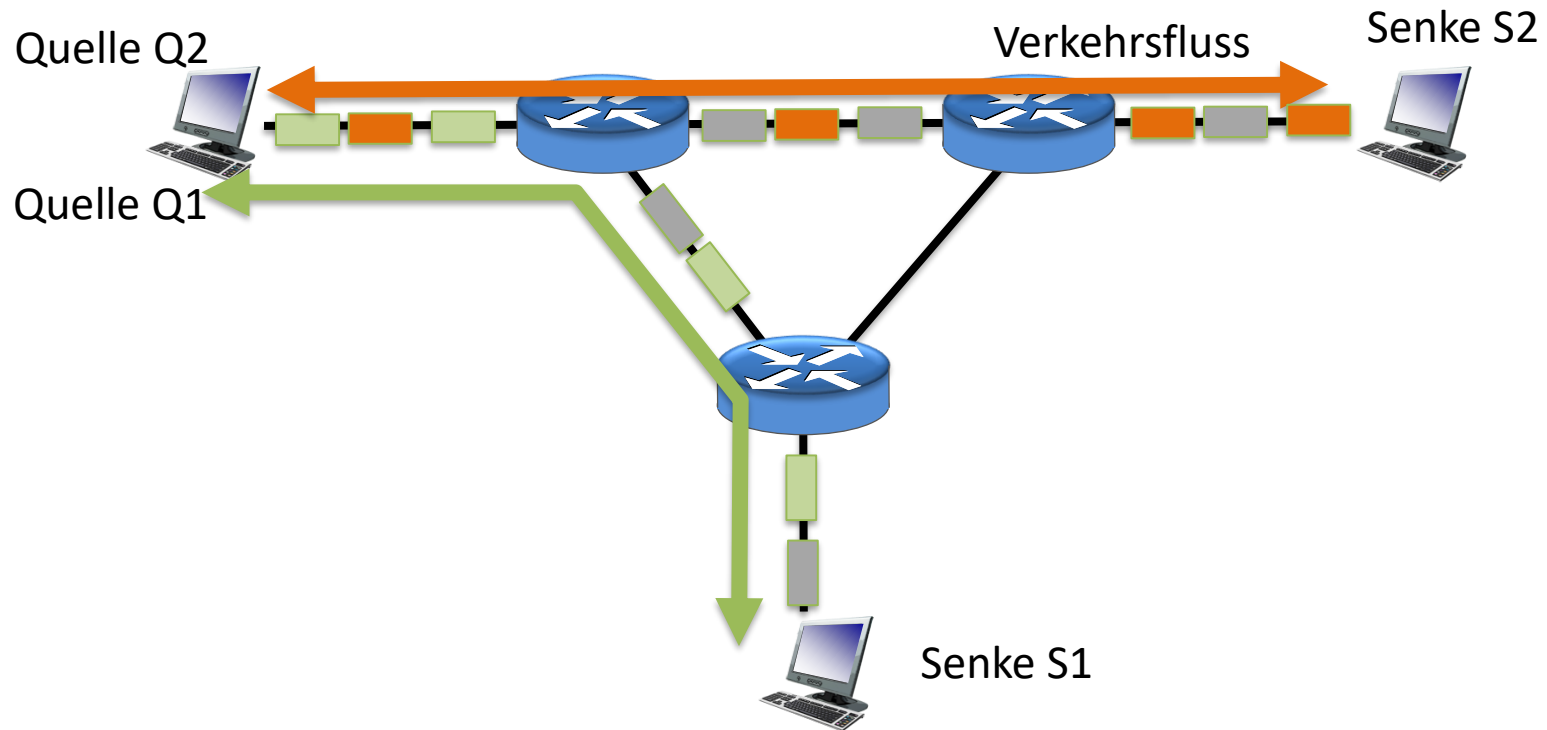
## 2.4 Sockets

## 2.5 Aufbau des Internets

## 2.6 Zusammenfassung

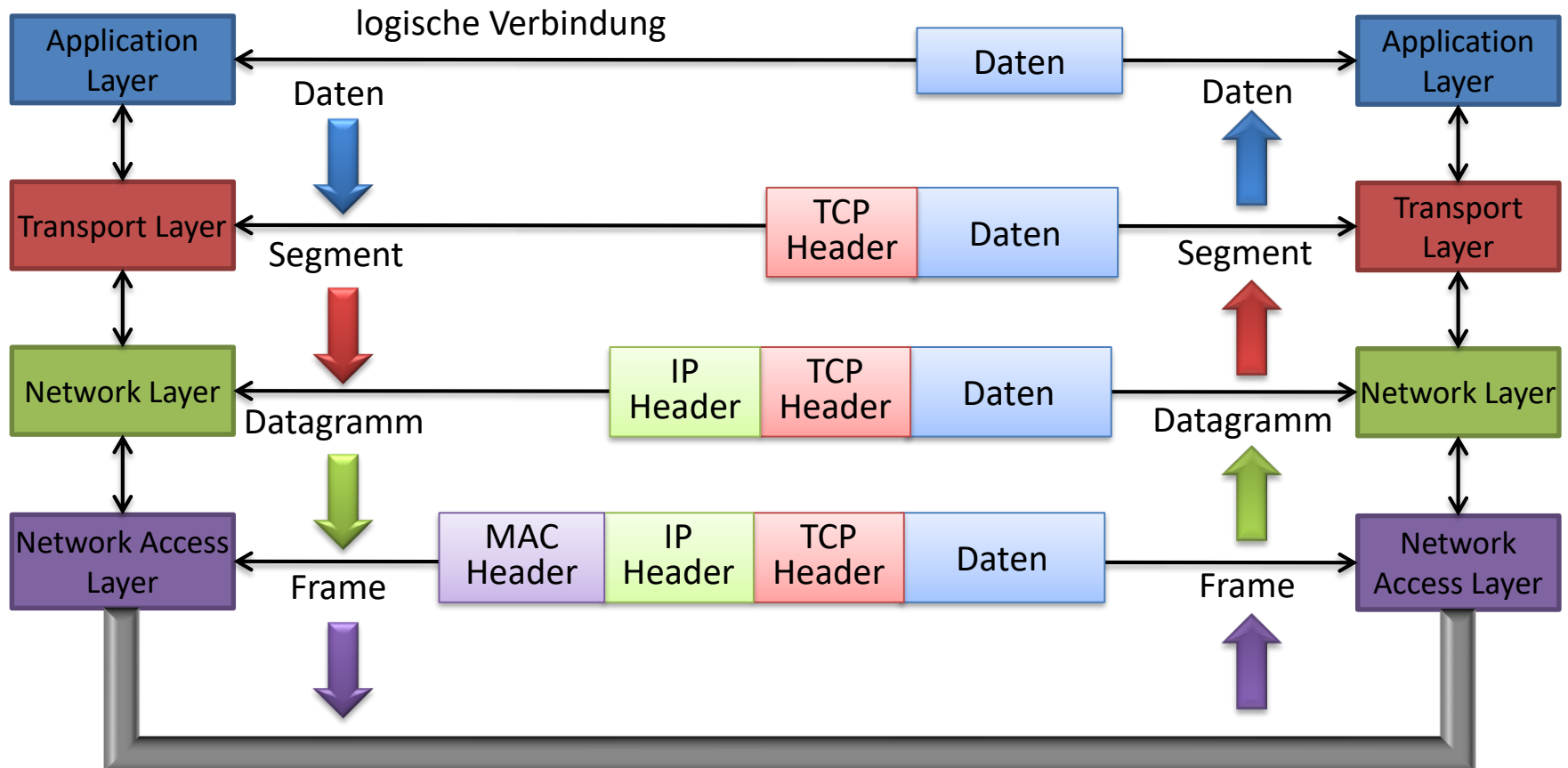
# Übertragung in Paketen

- Die Übertragung von Daten zwischen Quelle und Senke findet in Form von Paketen (engl. packet) statt
  - große Datenmengen werden in kleinere Pakete aufgeteilt
  - Pakete können auf unterschiedlichen Protokollschichten verschiedene Namen haben (Frames, Datagramme, Segmente)



# Kapselung der Pakete

- Jede Protokollschicht betrachtet das Paket der höheren Schichten als Payload und fügt seinen Header vorne an, bevor das Paket der darunterliegenden Schicht übergeben wird. Das Paket wird gekapselt.
- Ein Paket auf der physikalischen Leitung enthält Header von allen Protokollen.



- WireShark ist ein Werkzeug, das Datenpakete aufzeichnet und analysiert
- In der ersten Übung lernen Sie WireShark kennen
  - Sie können insbesondere den Aufbau eines IP Pakets verstehen und die Header der verschiedenen Protokolle analysieren
  - Sie gewinnen eine Übersicht der Kommunikationspartner, zu denen Sie Pakete schicken und von denen Sie Pakete empfangen

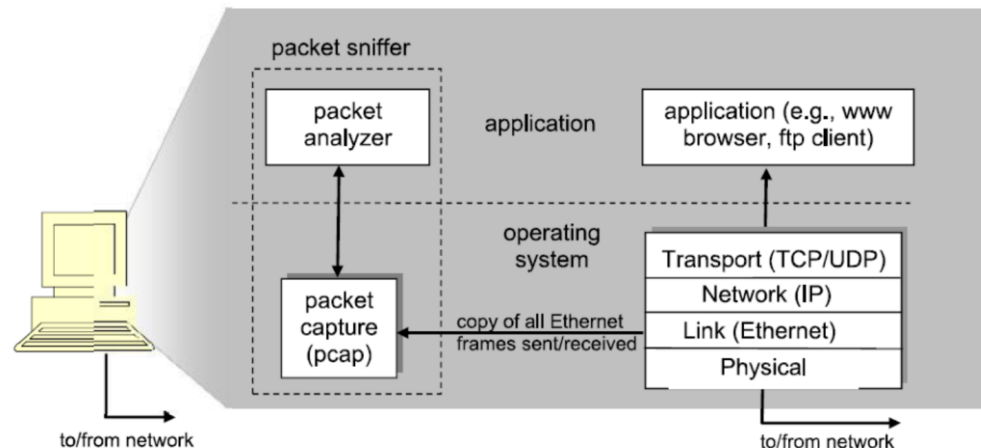


Figure 1: Packet sniffer structure

## 2.1 Terminologie

## 2.2 Grundlagen der Datenübertragung

### 2.2.1 Datenübertragung in Paketen

#### 2.2.1.1 Übertragung eines Pakets über einen Link

#### 2.2.1.2 Paketvermittlung

#### 2.2.1.3 Verzögerungen in einem Netzknoten

#### 2.2.1.4 Übertragung eines Packet-Bursts über eine Übertragungsstrecke

### 2.2.2 Paketverluste

### 2.2.3 Durchsatz

### 2.2.4 Messungen im Internet

## 2.3 Protokolle und Dienste

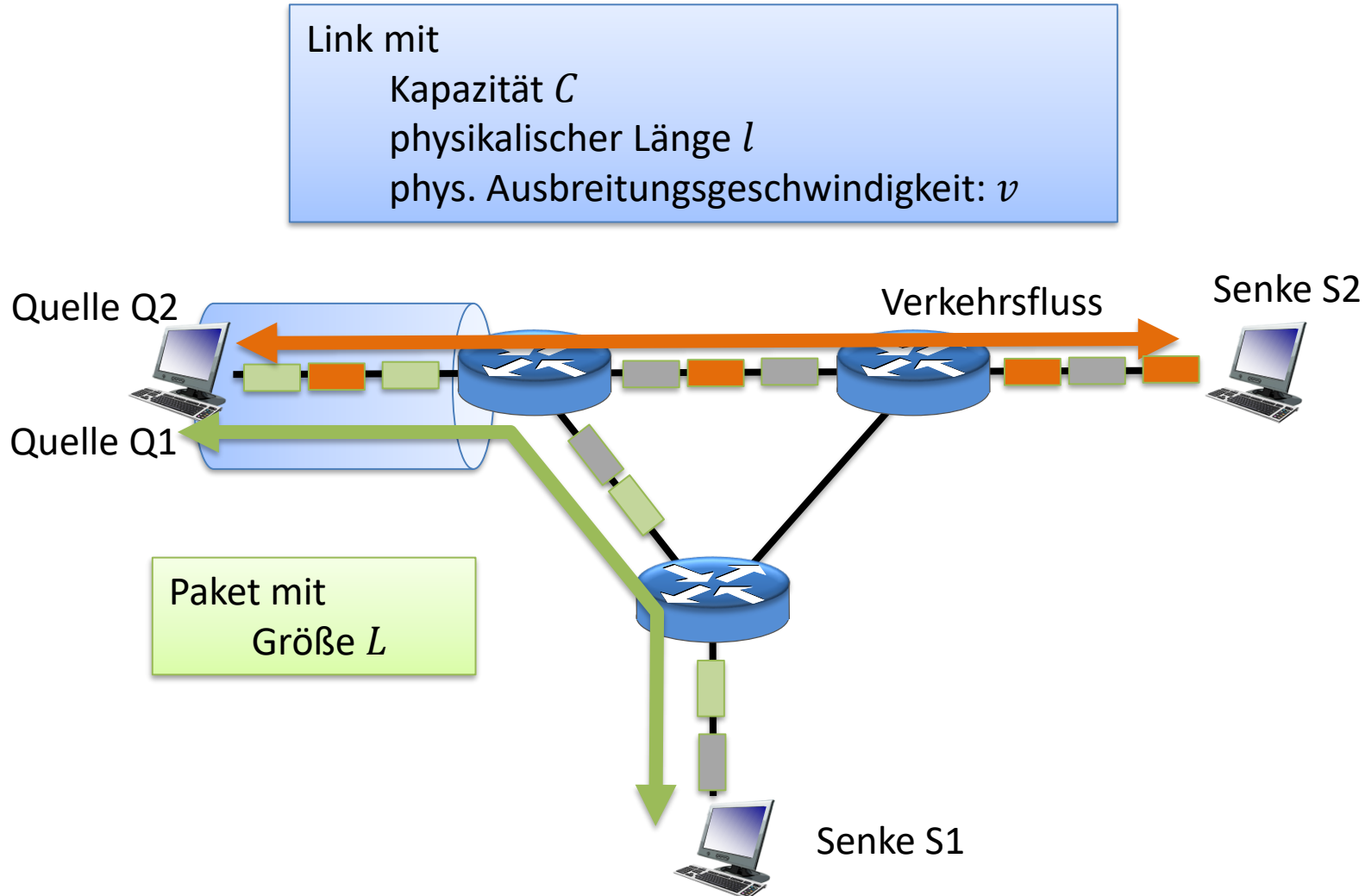
## 2.4 Sockets

## 2.5 Aufbau des Internets

## 2.6 Zusammenfassung

# Paketübertragung über einen Link

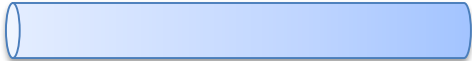

- Wie lange dauert die Übertragung eines Pakets über einen Link?

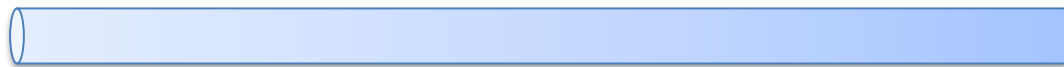


# Übertragung eines Pakets über einen Link

- Charakteristische Eigenschaften eines Links:
  - Übertragungskapazität  $C$  [bps]
  - physikalische Länge  $l$  [m]
  - physikalische Ausbreitungsgeschwindigkeit  $v$  [m/s]
- Abgeleitete Eigenschaften:
  - Bitübertragungsdauer:  $t_{bit} = \frac{1b}{C}$  [s]
  - Ausbreitungsverzögerung:  $t_{prop} = \frac{l}{v}$  [s]
  - Anzahl Bits auf Leitung:  $n_{bit} = C \cdot t_{prop}$  [b]
- Übertragung eines Pakets:
  - Paketgröße:  $L$  [b]
  - Übertragungsverzögerung:  $t_{tx} = L \cdot t_{bit} = \frac{L}{C}$  [s]
  - Anzahl Pakete auf der Leitung:  $n_p = \frac{n_{bit}}{L} = \frac{C \cdot t_{prop}}{L} = \frac{t_{prop}}{t_{tx}}$
  - Gesamtverzögerung:  $t_{link} = t_{tx} + t_{prop}$

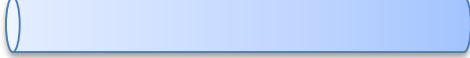
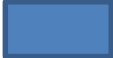
# Dauer der Übertragung eines Paket über einen WAN-Link

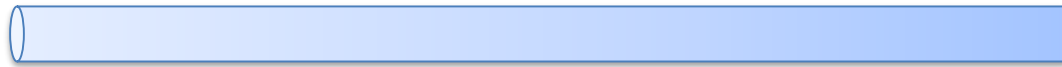
- Link 
  - Kapazität  $C$
  - Physikalische Länge  $l$
  - Ausbreitungsgeschwindigkeit:  $v$
- Paket: 
  - Größe  $L$
- Wie lange dauert die Übertragung eines Paket über den Link?
  - Übertragungsverzögerung?
  - Ausbreitungsverzögerung ?
  - Gesamtverzögerung ?
- Beispiel
  - $C = 10 \text{ Gbps}$
  - $l = 200 \text{ km}$  (WAN: >10km)
  - $v = 200000 \text{ km/s}$
  - $L = 10 \text{ kbit}$





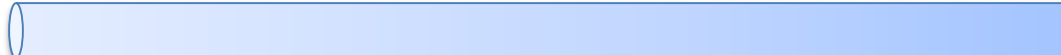
# Dauer der Übertragung eines Paket über einen WAN-Link

- Link 
  - Kapazität  $C$
  - Physikalische Länge  $l$
  - Ausbreitungsgeschwindigkeit:  $v$
- Beispiel
  - $C = 10 \text{ Gbps}$
  - $l = 200 \text{ km}$
  - $v = 200000 \text{ km/s}$
  - $L = 10 \text{ kbit}$
- Paket: 
  - Größe  $L$
- Wie lange dauert die Übertragung eines Paket über den Link?
  - Übertragungsverzögerung:  $t_{tx} = \frac{L}{C} = \frac{10 \text{ kbit}}{10 \text{ Gbps}} = \frac{10e3 \text{ bit}}{10e9 \text{ bps}} = 1 \mu\text{s}$
  - 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:  $t_{link} = t_{tx} + t_{prop} = 1 \mu\text{s} + 1 \text{ ms} = 1,001 \text{ ms}$

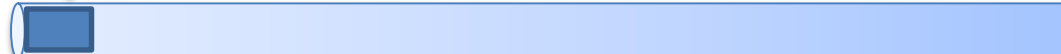


# Illustration des Übertragungsvorgangs (WAN-Link)

- Situation vor dem Versenden



- Situation während der Übertragungsverzögerung



- Situation nach der Übertragungsverzögerung



- Situation während der Ausbreitungsverzögerung



- Situation nach der Ausbreitungsverzögerung



- Eine Animation zur Übertragung eines Paket über einen Link mit verschiedenen Datenraten, Linklängen und Paketgrößen finden Sie hier:

[https://media.pearsoncmg.com/ph/esm/ecs\\_kurose\\_compnetwork\\_8/cw/content/interactiveanimations/transmission-vs-propagation-delay/transmission-propagation-delay-ch1/index.html](https://media.pearsoncmg.com/ph/esm/ecs_kurose_compnetwork_8/cw/content/interactiveanimations/transmission-vs-propagation-delay/transmission-propagation-delay-ch1/index.html)

## 2.1 Terminologie

## 2.2 Grundlagen der Datenübertragung

### 2.2.1 Datenübertragung in Paketen

2.2.1.1 Übertragung eines Pakets über einen Link

#### 2.2.1.2 Paketvermittlung

2.2.1.3 Verzögerungen in einem Netzknoten

2.2.1.4 Übertragung eines Packet-Bursts über eine Übertragungsstrecke

2.2.2 Paketverluste

2.2.3 Durchsatz

2.2.4 Messungen im Internet

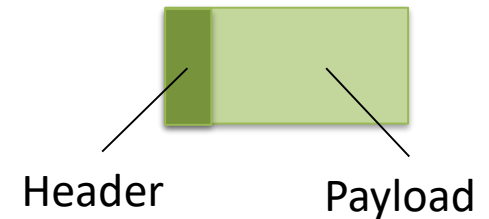
## 2.3 Protokolle und Dienste

## 2.4 Sockets

## 2.5 Aufbau des Internets

## 2.6 Zusammenfassung

- Pakete bestehen aus Payload und Header
  - die Payload beinhaltet die eigentlichen Daten
  - der Header beinhaltet Informationen für
    - den Empfänger
    - für Verkehrsknoten, die das Paket weiterleiten
- Paketvermittlung
  - Paketvermittlung bedeutet, dass im Header eines Pakets die Zieladresse steht
  - ein Netzknoten entscheidet aufgrund der Zieladresse im Paket, zu welchem nächsten Netzknoten das Paket weitergeleitet wird



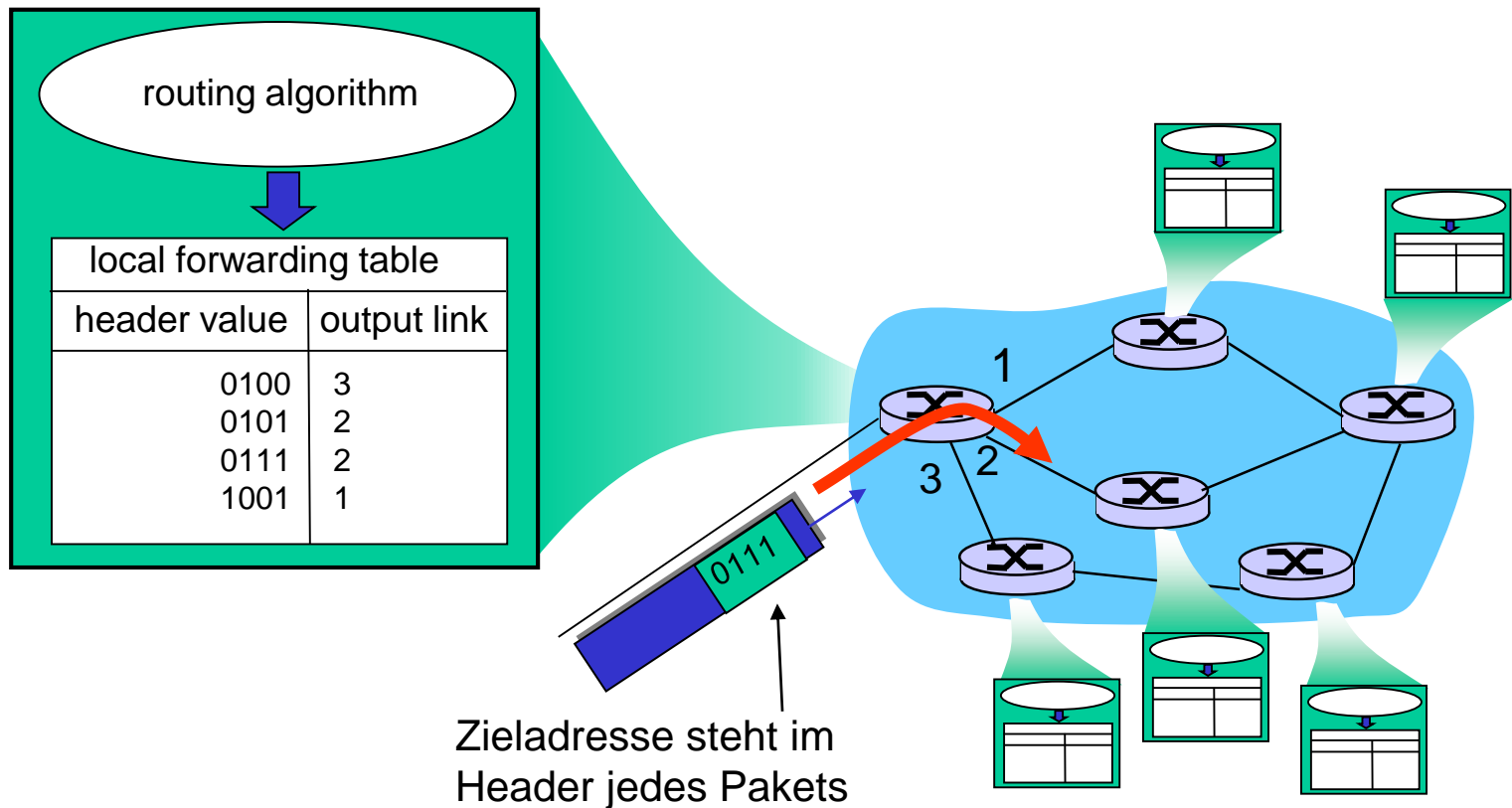
# Routing und Forwarding: Hauptfunktionen des Internets

## 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

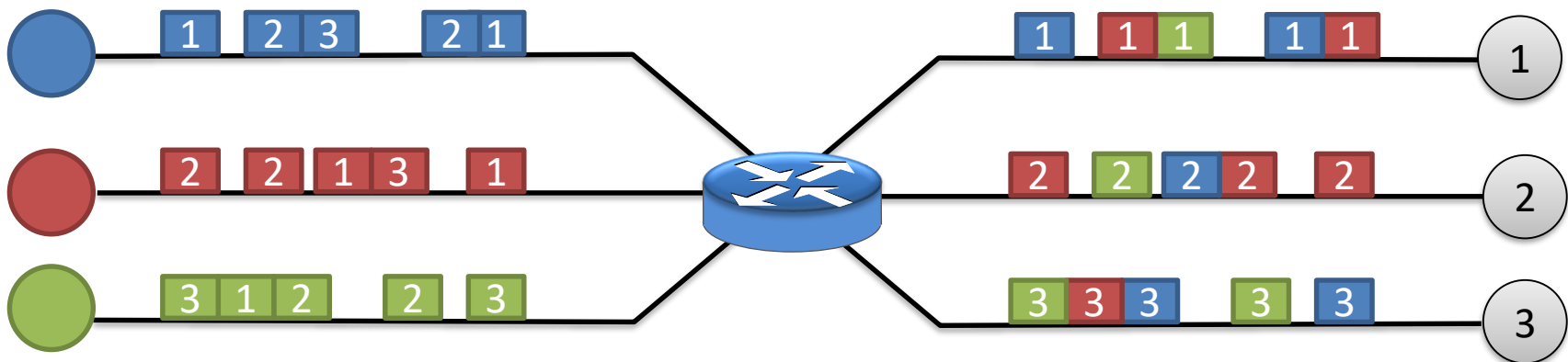
## Forwarding:

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



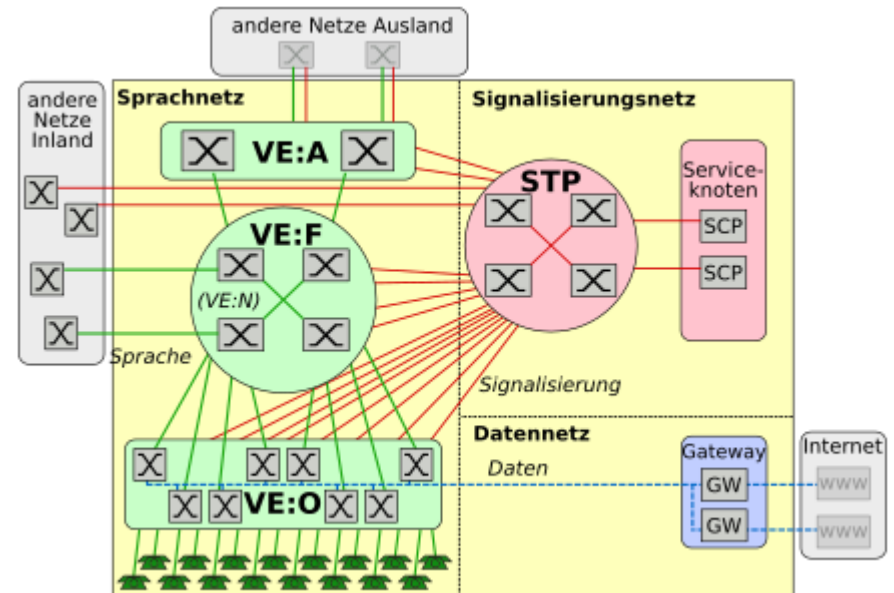
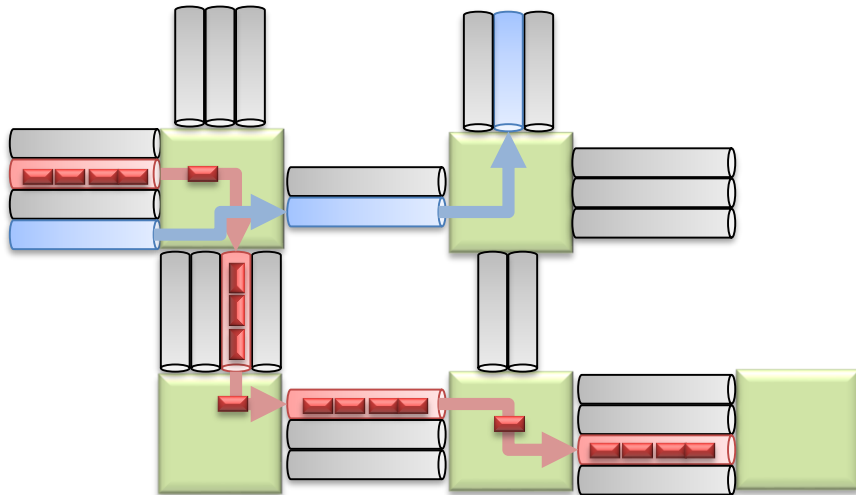
# Paketvermittlung – Statistischer Multiplex

- 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
  - wird auch als statistischer Multiplex bezeichnet
  - 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



# Leitungsvermittlung (Circuit Switching)

- bei der Leitungsvermittlung leitet ein Netzknoten alle Pakete, die auf einer Leitung ankommen auf einer anderen Leitung weiter
- die Pakete haben keine Zieladresse, d.h. es muss vor dem Senden des ersten Paket eine Ende-zu-Ende-Verbindung aufgebaut werden, die in jedem Netzknoten der Eingangsleitung eine Ausgangsleitung zuweist
- Leitungsvermittlung wird in der Festnetztelefonie eingesetzt (sofern der Transport nicht über IP stattfindet), die Verbindung wird in Vermittlungsstellen (switching center) durchgeschaltet





## 2.1 Terminologie

## 2.2 Grundlagen der Datenübertragung

### 2.2.1 Datenübertragung in Paketen

2.2.1.1 Übertragung eines Pakets über einen Link

2.2.1.2 Paketvermittlung

#### 2.2.1.3 Verzögerungen in einem Netzknoten

2.2.1.4 Übertragung eines Packet-Bursts über eine Übertragungsstrecke

2.2.2 Paketverluste

2.2.3 Durchsatz

2.2.4 Messungen im Internet

## 2.3 Protokolle und Dienste

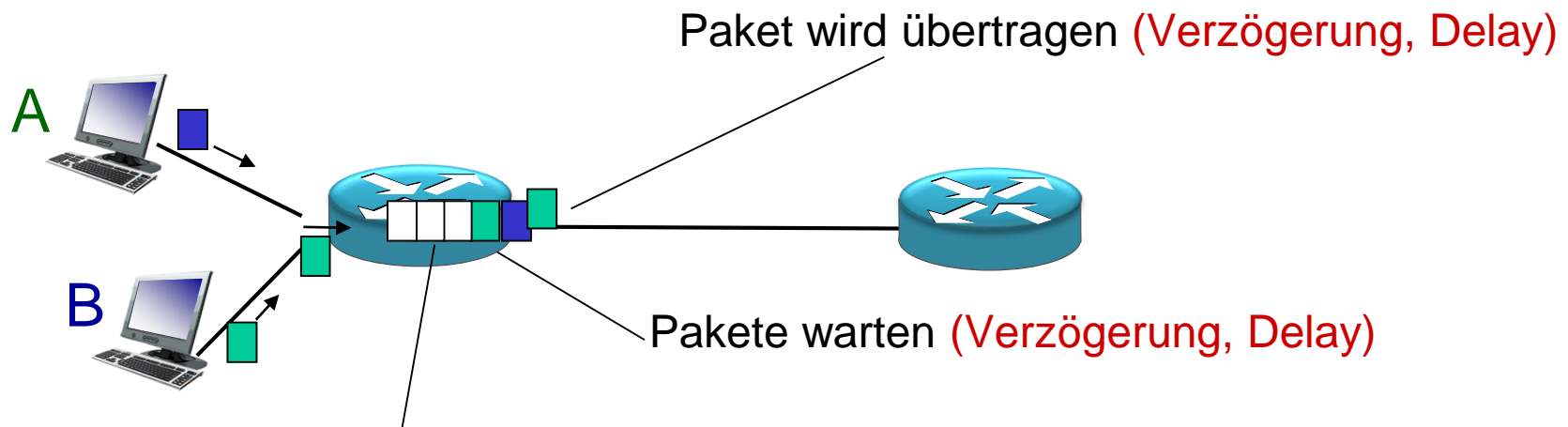
## 2.4 Sockets

## 2.5 Aufbau des Internets

## 2.6 Zusammenfassung

# Puffern von Paketen im Router

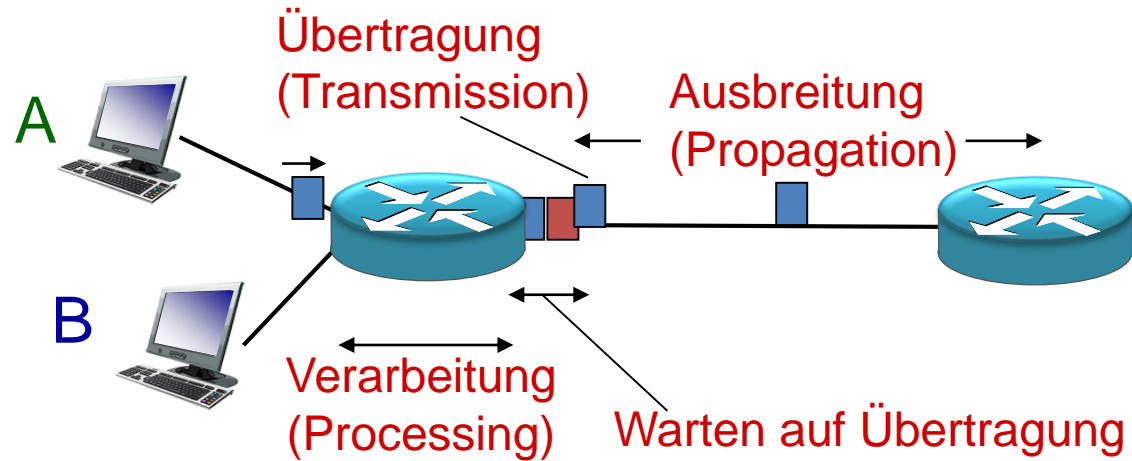
Wenn Pakete schneller ankommen, als sie weitergeleitet werden können, warten sie im Paketpuffer der Ausgangsleitung.



Paketpuffer pro Ausgangsleitung (engl. output buffer) als Warteschlange (engl. queue): hier warten Pakete auf ihre Übertragung. Wenn der Paketpuffer voll ist, müssen Pakete verworfen werden und es kommt zu Paketverlust.

- 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 (Queue)** 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 dieser 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



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

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 VDSL Downlink:
  - Übertragungsrate:  $C = 100 \text{ Mbps}$
  - Länge der VDSL Strecke:  $l = 100 \text{ m}$
  - Ausbreitungsgeschwindigkeit:  $v = 200000 \text{ km/s}$
  - Paketgröße:  $L = 1500 \text{ Bytes}$
- Last-unabhängige Größen:
  - Ausbreitungsverzögerung “last mile”
  - Übertragungsverzögerung eines Pakets
  - Gesamt-Verzögerung (ohne Warte- und Verarbeitungsverzögerung)
  - Wie viele Bits passen auf die VDSL-Leitung?
  - Wie viele Pakete passen auf die VDSL Leitung?
  - Physikalische Länge
    - eines Bits?
    - eines Pakets?

- Parameter VDSL Downlink:
  - Übertragungsrate:  $C = 100 \text{ Mbps}$
  - Länge der VDSL Strecke (last mile):  $l = 100 \text{ m}$
  - Ausbreitungsgeschwindigkeit:  $v = 200000 \text{ km/s}$
  - Paketgröße:  $L = 1500 \text{ Bytes}$
- Last-unabhängige Größen:
  - Ausbreitungsverzögerung “last mile”:  $t_{prop} = \frac{l}{v} = 0,5 \mu\text{s}$
  - Übertragungsverzögerung eines Pakets:  $t_{tx} = \frac{L}{C} = \frac{12 \text{ kb}}{100 \text{ Mbps}} = 120 \mu\text{s}$
  - Gesamt-Verzögerung (ohne Warte- und Verarbeitungsverzögerung):  
$$t_{link} = t_{tx} + t_{prop} = 120,5 \mu\text{s}$$
  - Wie viele Bits passen auf die VDSL-Leitung?
    - Dauer einer Bitübertragung:  $t_{bit} = \frac{1 \text{ b}}{C} = \frac{1 \text{ b}}{100 \text{ Mbps}} = 10 \text{ ns}$
    - Anzahl Bits auf der Leitung:  $n_{bit} = C \cdot t_{prop} = 100 \text{ Mbps} \cdot 0,5 \mu\text{s} = 50 \text{ b}$
    - Anzahl Pakete auf der Leitung?  $n_p = \frac{n_{bit}}{L} = \frac{50 \text{ b}}{12 \text{ kb}} \approx 0,00416 \text{ (1/240 Paket)}$
    - Physikalische Länge
      - eines Bits:  $l_{bit} = t_{bit} \cdot v = 10 \text{ ns} \cdot 200000 \frac{\text{km}}{\text{s}} = 2 \text{ m}$
      - eines Pakets:  $l_p = L \cdot l_{bit} = 12000 \cdot 2 \text{ m} = 24 \text{ km}$

- Parameter VDSL Downlink:
  - Übertragungsrate:  $C = 100 \text{ Mbps}$
  - Länge der VDSL Strecke (last mile):  $l = 100 \text{ m}$
  - Ausbreitungsgeschwindigkeit:  $v = 200000 \text{ km/s}$
  - Paketgröße:  $L = 1500 \text{ Bytes}$
- Last-unabhängige Größen:
  - Ausbreitungsverzögerung “last mile”:  $t_{prop} = \frac{l}{v} = 0,5 \mu\text{s}$
  - Übertragungsverzögerung eines Pakets:  $t_{tx} = \frac{L}{C} = \frac{12 \text{ kb}}{100 \text{ Mbps}} = 120 \mu\text{s}$
  - Gesamt-Verzögerung (ohne Warte- und Verarbeitungsverzögerung):  
$$t_{link} = t_{tx} + t_{prop} = 120,5 \mu\text{s}$$
- Last-abhängige Größen
  - Warteschlangenlänge bei Eintreffen des Pakets:  $n_{queue} = 100$
  - Warteverzögerung bei 100 Paketen in der Warteschlange?
  - Gesamtverzögerung mit Warteverzögerung aber ohne Verarbeitungsverzögerung?



- Parameter VDSL Downlink:
  - Übertragungsrate:  $C = 100 \text{ Mbps}$
  - Länge der VDSL Strecke (last mile):  $l = 100 \text{ m}$
  - Ausbreitungsgeschwindigkeit:  $v = 200000 \text{ km/s}$
  - Paketgröße:  $L = 1500 \text{ Bytes}$
- Last-unabhängige Größen:
  - Ausbreitungsverzögerung “last mile”:  $t_{prop} = \frac{l}{v} = 0,5 \mu\text{s}$
  - Übertragungsverzögerung eines Pakets:  $t_{tx} = \frac{L}{C} = \frac{12 \text{ kb}}{100 \text{ Mbps}} = 120 \mu\text{s}$
  - Gesamt-Verzögerung (ohne Warte- und Verarbeitungsverzögerung):  
$$t_{link} = t_{tx} + t_{prop} = 120,5 \mu\text{s}$$
- Last-abhängige Größen
  - Warteschlangenlänge bei Eintreffen des Pakets:  $n_{queue} = 100$
  - Warteverzögerung in der Warteschlange?  
$$t_{queue} = n_{queue} \cdot t_{tx} = 100 \cdot 120 \mu\text{s} = 12 \text{ ms}$$
  - Gesamtverzögerung mit Warteverzögerung aber ohne Verarbeitungsverzögerung?  
$$t_{node} = t_{queue} + t_{tx} + t_{prop} = 12,1205 \text{ ms}$$

## 2.1 Terminologie

## 2.2 Grundlagen der Datenübertragung

### 2.2.1 Datenübertragung in Paketen

2.2.1.1 Übertragung eines Pakets über einen Link

2.2.1.2 Paketvermittlung

2.2.1.3 Verzögerungen in einem Netzknoten

**2.2.1.4 Übertragung eines Packet-Bursts über eine Übertragungsstrecke**

2.2.2 Paketverluste

2.2.3 Durchsatz

2.2.4 Messungen im Internet

## 2.3 Protokolle und Dienste

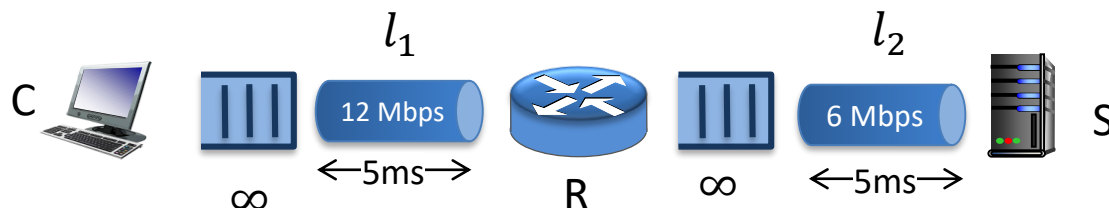
## 2.4 Sockets

## 2.5 Aufbau des Internets

## 2.6 Zusammenfassung

# Beispiel: Übertragung über eine Übertragungsstrecke

- Übertragungsstrecke mit zwei Links



1 Paket  
(1500 Bytes)

Ende-zu-Ende-  
Verzögerung?

- Übertragungsverzögerungen der Links

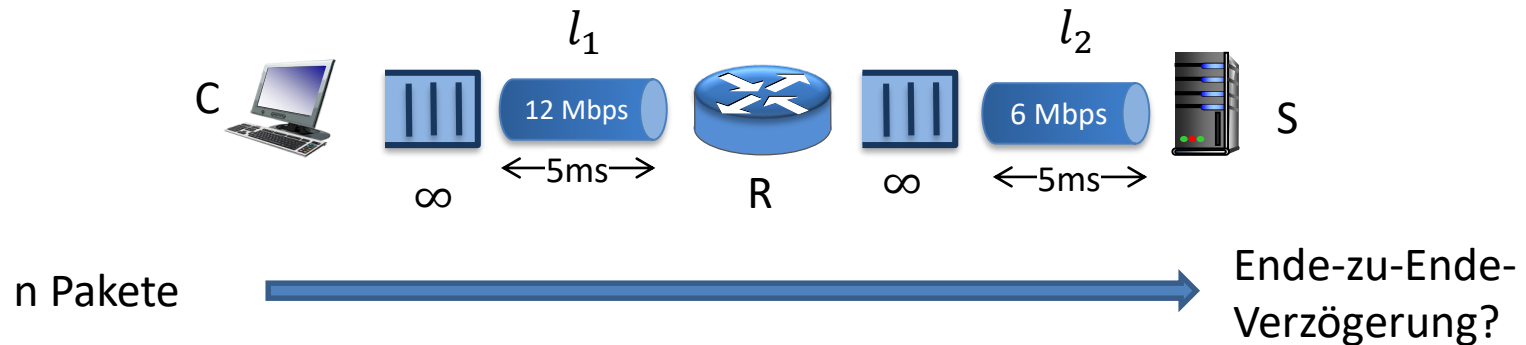
$$t_{tx,l_1} = \frac{12 \text{ kb}}{12 \text{ Mbps}} = 1 \text{ ms} \quad t_{tx,l_2} = 2 \cdot t_{tx,l_1} = 2 \text{ ms}$$

- Ende-zu-Ende-Verzögerung für ein Paket:

$$T_{E2E} = \sum_l t_l = \sum_l t_{tx,l} + t_{prop,l} = 1 \text{ ms} + 5 \text{ ms} + 2 \text{ ms} + 5 \text{ ms} = 13 \text{ ms}$$

# Beispiel: Mehrere Pakete über eine Übertragungsstrecke

- Übertragungsstrecke



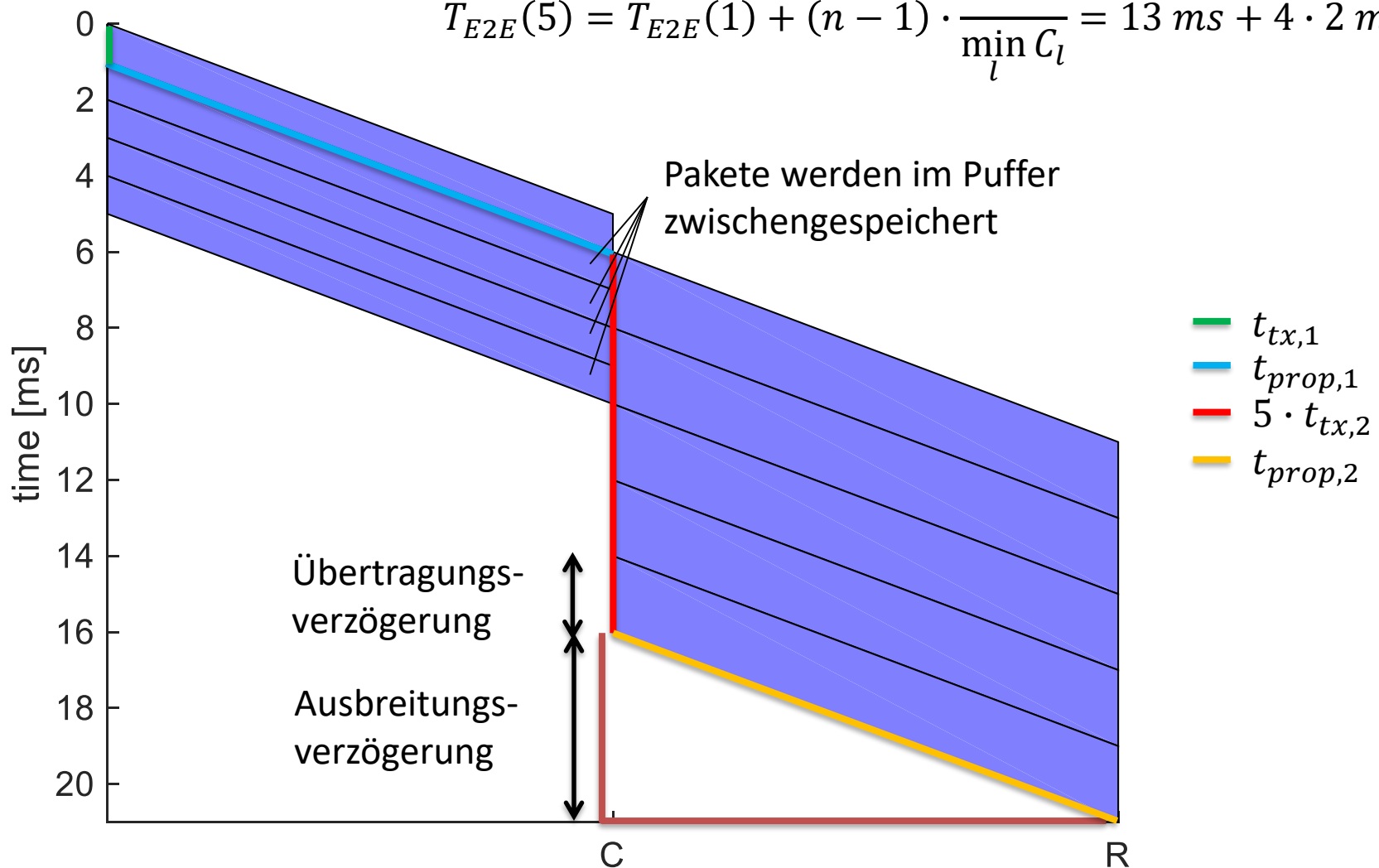
- Ende-zu-Ende-Verzögerung der Übertragungsstrecke für einen Packet-Burst (auch Packet-Train) aus  $n$  Paketen:

$$T_{E2E}(n) = T_{E2E}(1) + (n - 1) \cdot \frac{L}{\min_l C_l} = T_{E2E}(1) + (n - 1) \cdot \max_l t_{tx,l}$$
$$T_{E2E}(1) = T_{E2E}$$

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

# Darstellung der Übertragung der 5 Pakete

$$T_{E2E}(5) = T_{E2E}(1) + (n - 1) \cdot \frac{L}{\min_l C_l} = 13 \text{ ms} + 4 \cdot 2 \text{ ms}$$



## 2.1 Terminologie

## **2.2 Grundlagen der Datenübertragung**

### 2.2.1 Datenübertragung in Paketen

### **2.2.2 Paketverluste**

#### **2.2.2.1 Ursachen für Paketverluste**

#### 2.2.2.2 Umgang mit Paketverlusten

### 2.2.3 Durchsatz

### 2.2.4 Messungen im Internet

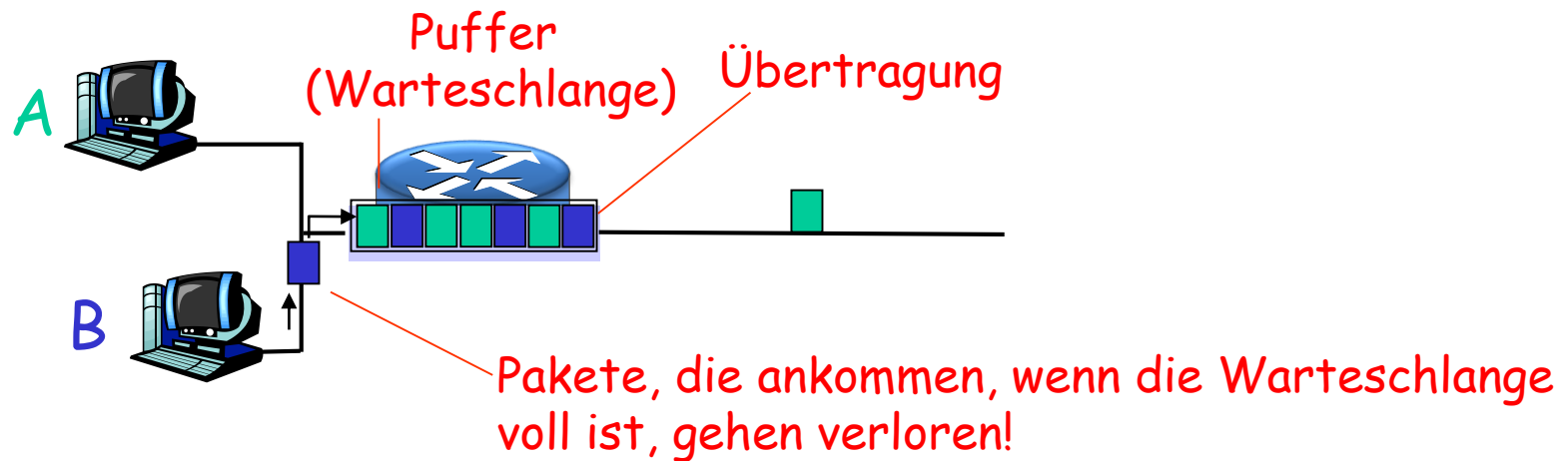
## 2.3 Protokolle und Dienste

## 2.4 Sockets

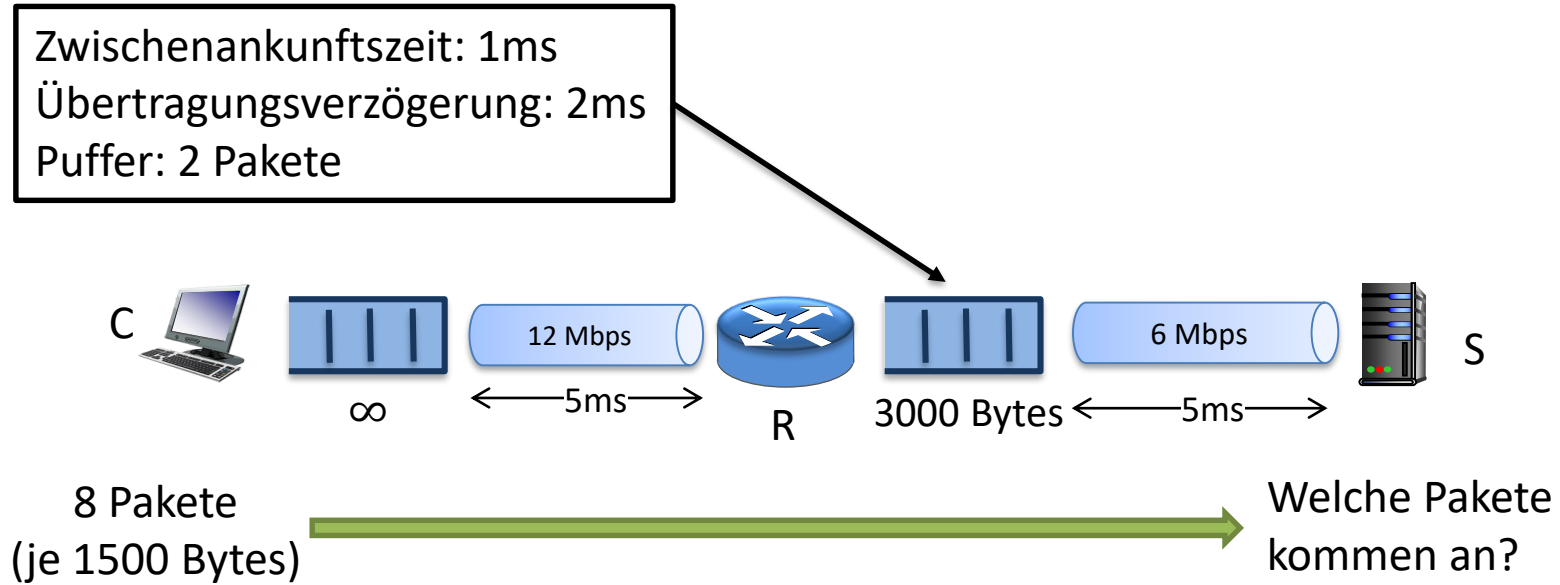
## 2.5 Aufbau des Internets

## 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!



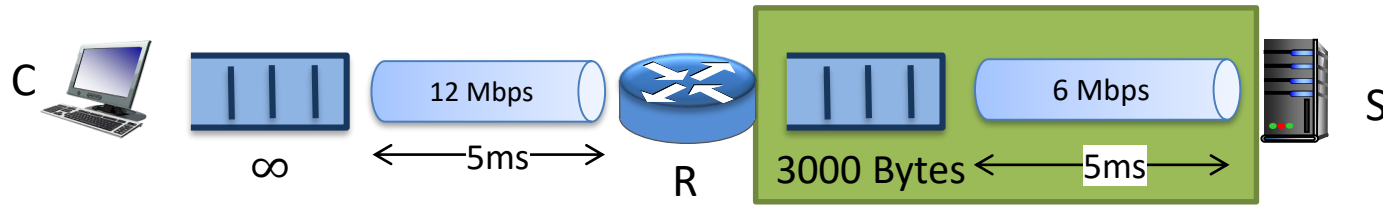
# Beispiel: Pakete im Buffer vor Link 2



- 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



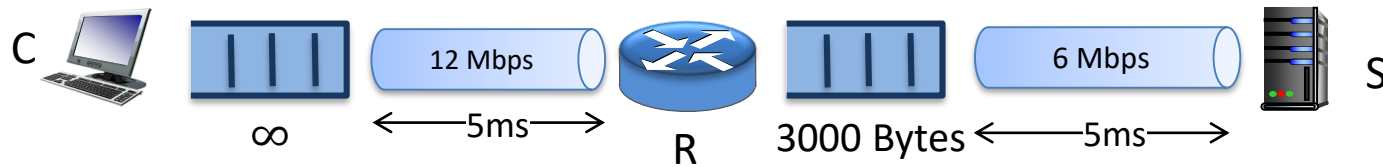
Zwischenankunftszeit: 1ms

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



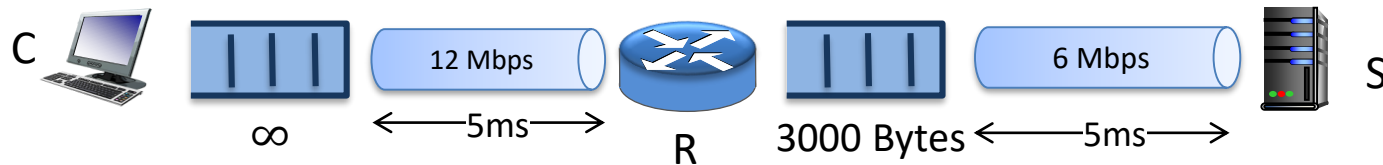
Zwischenankunftszeit: 1ms

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

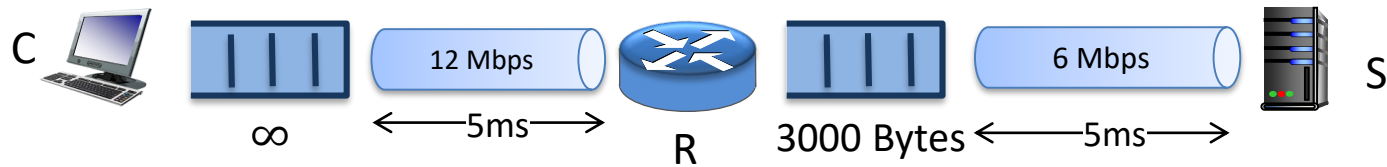
Übertragungsverzögerung: 2ms

Puffer: 2 Pakete

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

- Ab dem sechsten Paket geht jedes zweite Paket verloren!!!
- Puffer können nur bei kurzfristiger Überlast Paketverluste vermeiden; bei längere Überlast helfen auch große Puffer nicht
- Achtung: die maximale Warteverzögerung hängt von der Puffergröße ab; die Warteverzögerung sollte auch bei vollem Puffer einen Maximalwert nicht überschreiten.

# Beispiel: Pakete im Buffer vor Link 2 (kürzer)



Zwischenankunftszeit: 1ms

Übertragungsverzögerung: 2ms

Puffer: 2 Pakete

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

- Eine Animation zur Pufferung von Paketen finden Sie hier:  
[https://media.pearsoncmg.com/ph/esm/ecs\\_kurose\\_compnetwork\\_8/cw/content/interactiveanimations/queuing-loss-applet/index.html](https://media.pearsoncmg.com/ph/esm/ecs_kurose_compnetwork_8/cw/content/interactiveanimations/queuing-loss-applet/index.html)

## 2.1 Terminologie

## **2.2 Grundlagen der Datenübertragung**

### 2.2.1 Datenübertragung in Paketen

### **2.2.2 Paketverluste**

#### 2.2.2.1 Ursachen für Paketverluste

#### **2.2.2.2 Umgang mit Paketverlusten**

### 2.2.3 Durchsatz

### 2.2.4 Messungen im Internet

## 2.3 Protokolle und Dienste

## 2.4 Sockets

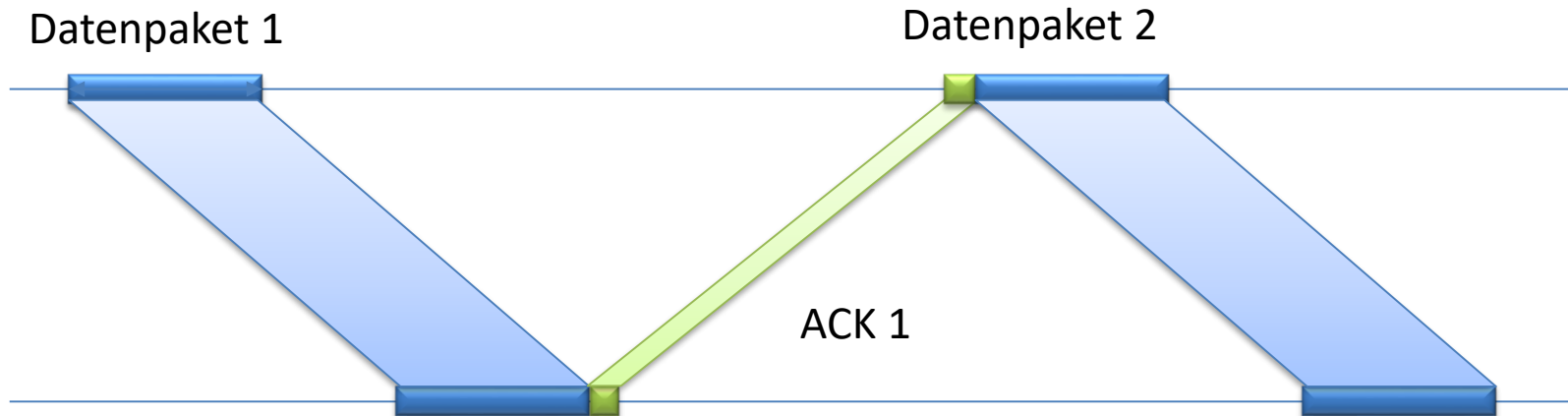
## 2.5 Aufbau des Internets

## 2.6 Zusammenfassung

- 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

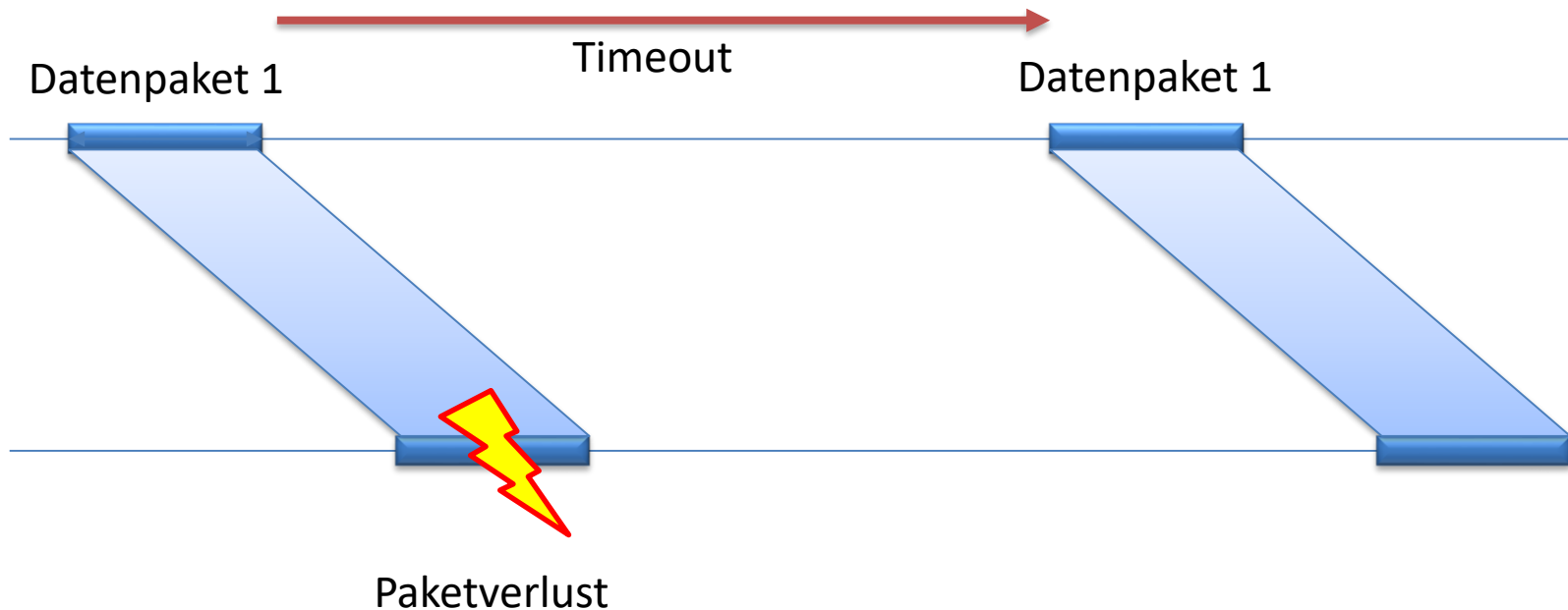
- Sender überträgt ein Paket und wartet auf eine Bestätigung
- nach korrektem Empfang eines Datenpakets sendet der Empfänger eine Bestätigung (Acknowledgement, ACK)
- nach Erhalt der Bestätigung überträgt der Sender das nächste Datenpaket
- falls nach einiger Zeit keine Bestätigung eintrifft (Timeout),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



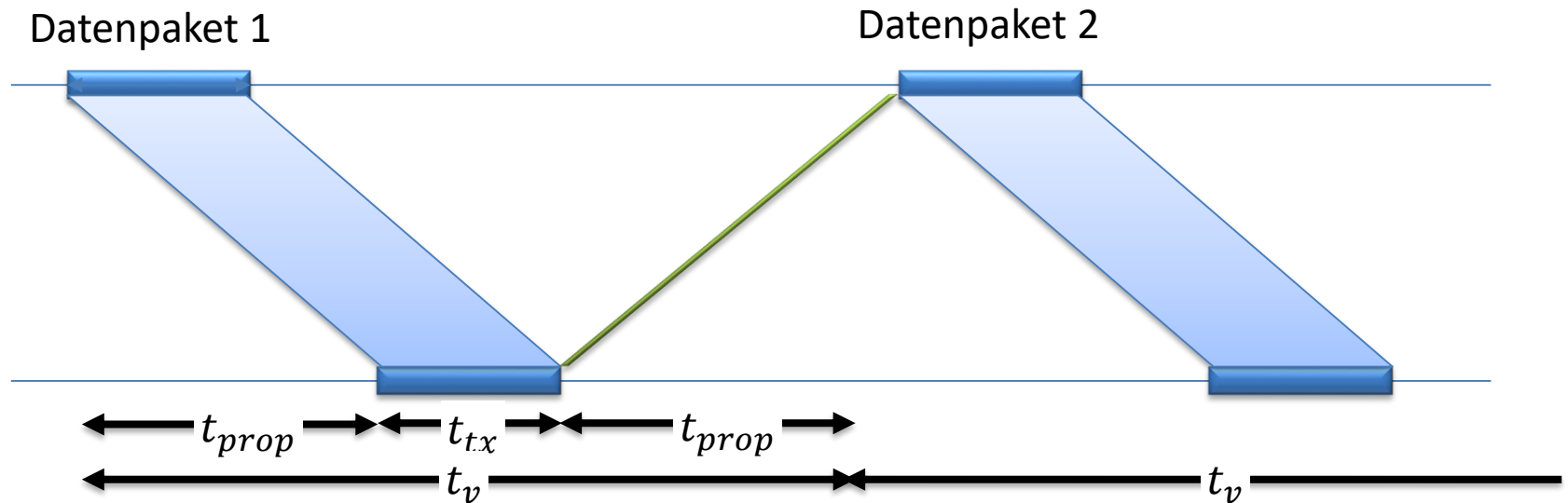


# Wiederholungsaufforderung: Send-and-Wait

- 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

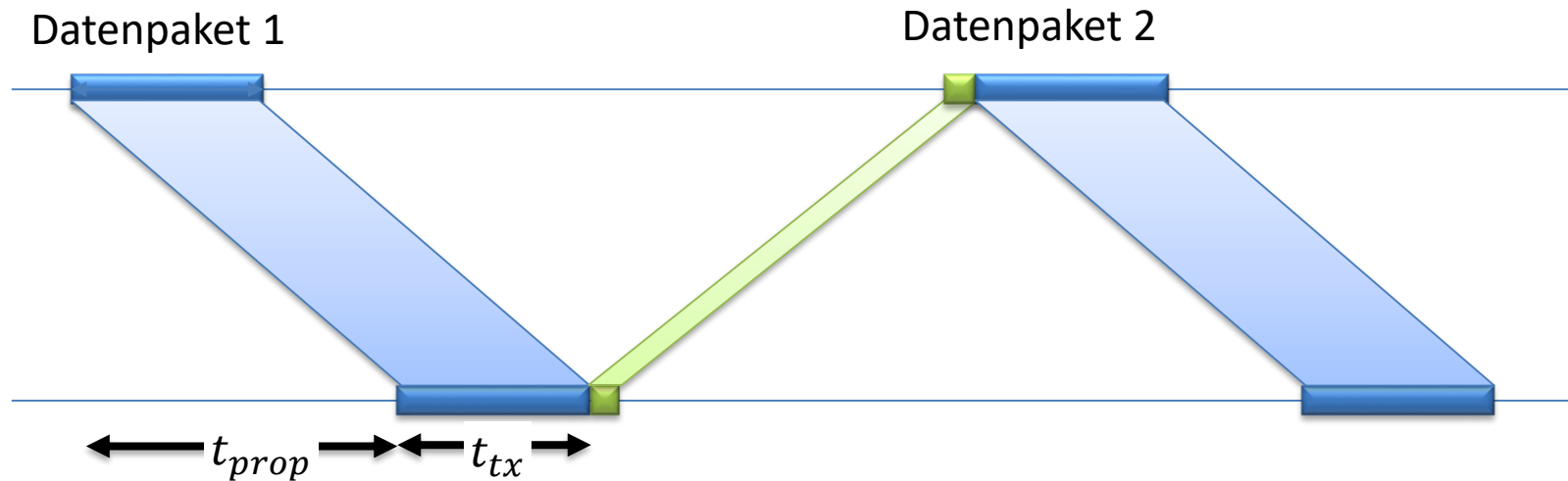


# Send-and-Wait: Durchsatz und Auslastung



- 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 ein Paket die Leitung virtuell belegt
  - „Virtuelle“ Übertragungszeit für ein Datenpaket (ohne Fehler):  $t_v = t_{tx} + 2 \cdot t_{prop}$
  - Maximaler Durchsatz:  $D_{max} = \frac{1}{t_v} = \frac{1}{t_{tx} + 2 \cdot t_{prop}}$  in Paketen / s
- Die **Auslastung**  $\rho$  einer Leitung bezeichnet den Anteil der Zeit, zu dem über die Leitung effektiv Daten übertragen werden.
  - Auslastung:  $\rho = \frac{t_{tx}}{t_{tx} + 2 \cdot t_{prop}}$

# Send-and-Wait: Beispiele



- Beispiele:
  - Paketgröße:  $L = 1\text{ kb}$
  - Übertragungskapazität:  $C = 1\text{ Mbps}$
  - Ausbreitungsverzögerung
    - Bluetooth:  $t_{prop} = 10\text{ }\mu\text{s}$
    - Satellit:  $t_{prop} = 250\text{ ms}$
  - 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 für Bluetooth und Satellit

$$t_{tx} = \frac{1 \text{ kbit}}{1 \text{ Mbps}} \approx 0,001 \text{ s} = 1 \text{ ms}$$

- Bluetooth:

- Durchsatz:  $D_{max} = \frac{1}{t_{tx} + 2 \cdot t_{prop}} = \frac{1}{1,02 \text{ ms}} \approx 98 \frac{\text{Pakete}}{\text{s}}$
- Auslastung:  $\rho = \frac{t_{tx}}{t_{tx} + 2 \cdot t_{prop}} = \frac{1 \text{ ms}}{1,02 \text{ ms}} \approx 0,98 = 98\%$

- Satellit:

- Durchsatz:  $D_{max} = \frac{1}{t_{tx} + 2 \cdot t_{prop}} = \frac{1}{501 \text{ ms}} \approx 2 \frac{\text{Pakete}}{\text{s}}$
- Auslastung:  $\rho = \frac{t_{tx}}{t_{tx} + 2 \cdot t_{prop}} = \frac{1 \text{ ms}}{501 \text{ ms}} \approx 0,002 = 0,2\%$

- Send-and-Wait ist für Übertragungsstrecken mit hoher Ausbreitungsverzögerung oder großer Übertragungskapazität ungeeignet
  - andere Verfahren zur Datenflusssteuerung in Kapitel 4 Transportschicht

## 2.1 Terminologie

## **2.2 Grundlagen der Datenübertragung**

### 2.2.1 Datenübertragung in Paketen

### 2.2.2 Paketverluste

### **2.2.3 Durchsatz**

#### **2.2.3.1 Ende-zu-Ende-Durchsatz**

#### 2.2.3.2 Max-Min-Fairness

### 2.2.4 Messungen im Internet

## 2.3 Protokolle und Dienste

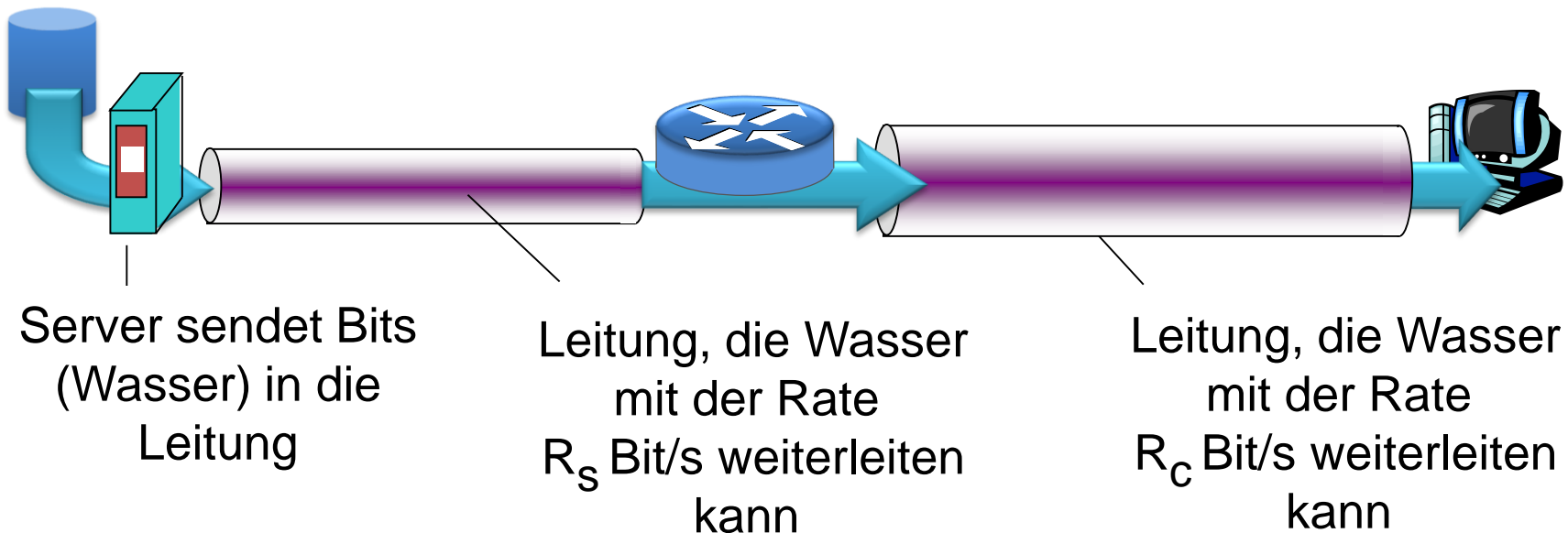
## 2.4 Sockets

## 2.5 Aufbau des Internets

## 2.6 Zusammenfassung

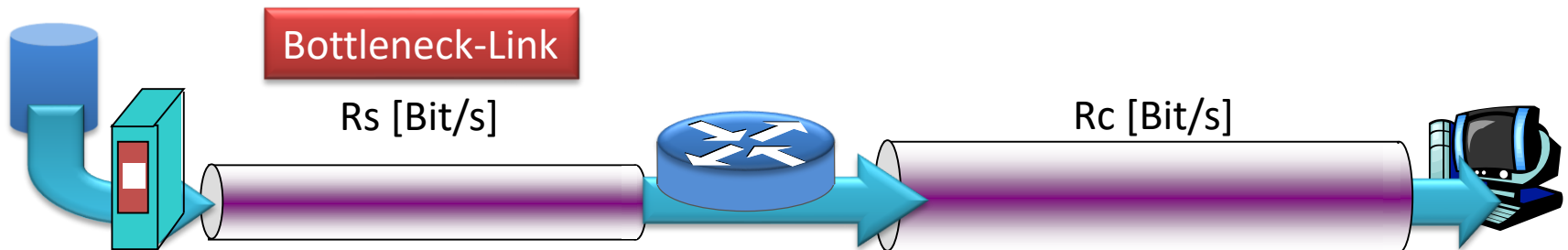
# Durchsatz

- Definition: Der **Durchsatz** eines Links entspricht der Menge an Daten, die pro Zeiteinheit erfolgreich über einen Link transportiert werden. Die Einheit des Durchsatzes ist Bit pro Sekunde [bps].
- Der Durchsatz eines Links ist immer kleiner als die Kapazität des Links und der Durchsatz ist auch immer kleiner als der ankommende Verkehr
- Definition: Der **Ende-zu-Ende-Durchsatz** eines Verkehrsflusses ist die Rate (Bit/Zeiteinheit), mit der Daten zwischen Sender und Empfänger ausgetauscht werden.
  - Analogie: Wasserleitung

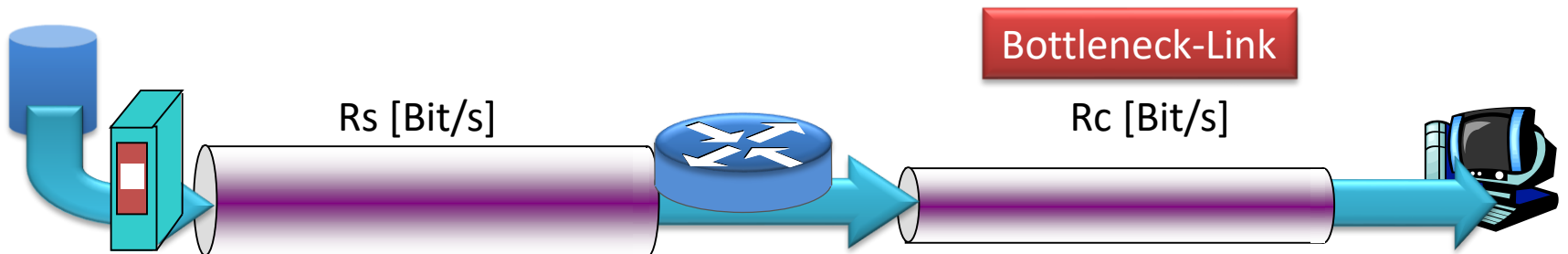


# Durchsatz

- 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?

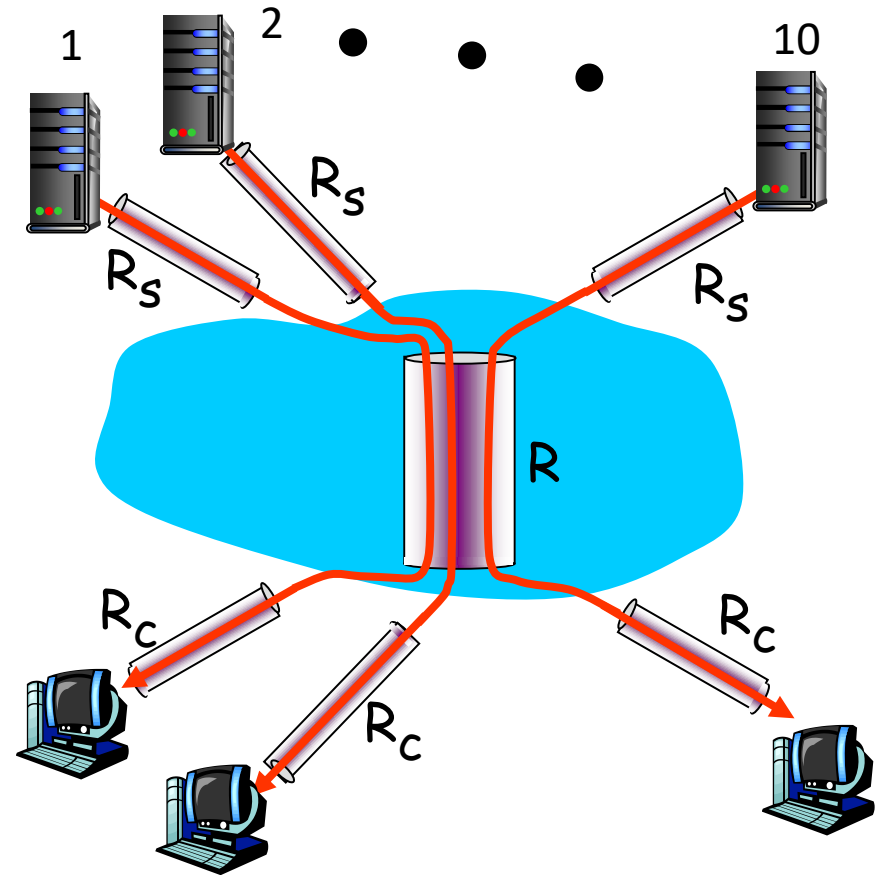


- $R_s > R_c$ : Was ist der durchschnittliche Ende-zu-Ende-Durchsatz?



# 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

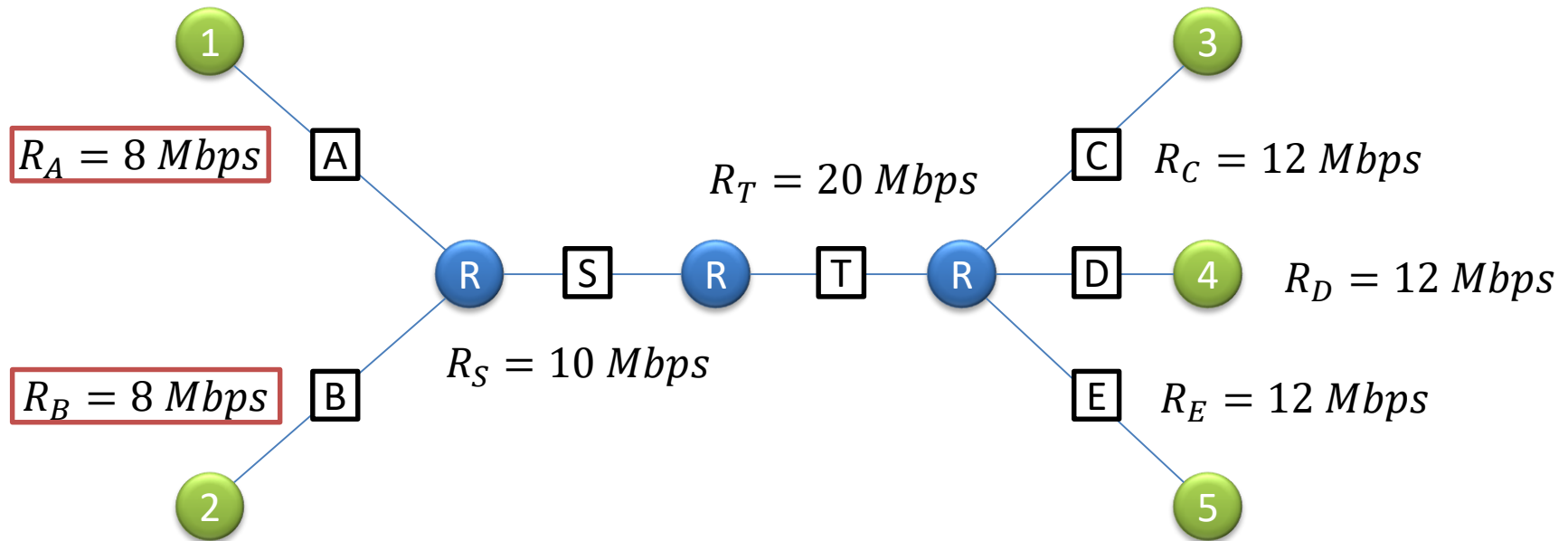
- Der Ende-zu-Ende-Durchsatz  $D$  eines Flusses  $v$  ist das Minimum aus der Datenrate  $S_v$ , mit der die Quelle sendet und der Bottleneckdatenrate  $B_v$
- Die Bottleneckdatenrate  $B_v$  ist die kleinste Rate pro Verkehrsfluss  $R_l$  für einen Link  $l$ , der auf dem Pfad  $P_v$  des Verkehrsflusses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



## 2.1 Terminologie

## **2.2 Grundlagen der Datenübertragung**

### 2.2.1 Datenübertragung in Paketen

### 2.2.2 Paketverluste

### **2.2.3 Durchsatz**

#### 2.2.3.1 Ende-zu-Ende-Durchsatz

#### **2.2.3.2 Max-Min-Fairness**

### 2.2.4 Messungen im Internet

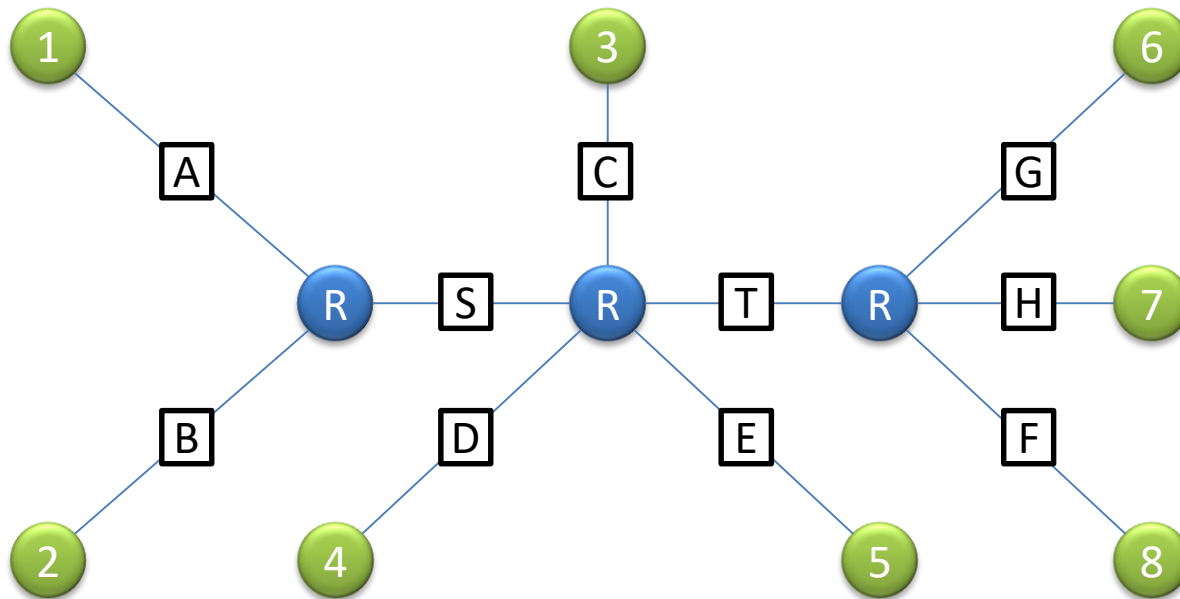
## 2.3 Protokolle und Dienste

## 2.4 Sockets

## 2.5 Aufbau des Internets

## 2.6 Zusammenfassung

- 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?



- Realistische Netze sind nicht zwangsweise baumartig aufgebaut sondern können auch Schleifen aufweisen.
- In einem solchen Netz können die Flüsse verschiedene Bottlenecks haben und die Bestimmung der Bottlenecks ist nicht mehr trivial. 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
  - zunächst wird der Link mit der kleinsten Rate pro Fluss bestimmt und allen Flüssen durch diesen Link wird diese Rate final zugewiesen
  - Flüsse, die nicht durch diesen Link gehen, können eine höhere Rate erhalten und um diese zu bestimmen wird der nächste Bottleneck-Link betrachtet
- Max-Min-Fairness ist ein theoretisches Konzept. In einen idealen Netz mit idealen Flüssen, würde die Datenflusssteuerung von TCP für einen max-min fairen Durchsatz sorgen.

## 2.1 Terminologie

## **2.2 Grundlagen der Datenübertragung**

### 2.2.1 Datenübertragung in Paketen

### 2.2.2 Paketverluste

### 2.2.3 Durchsatz

### **2.2.4 Messungen im Internet**

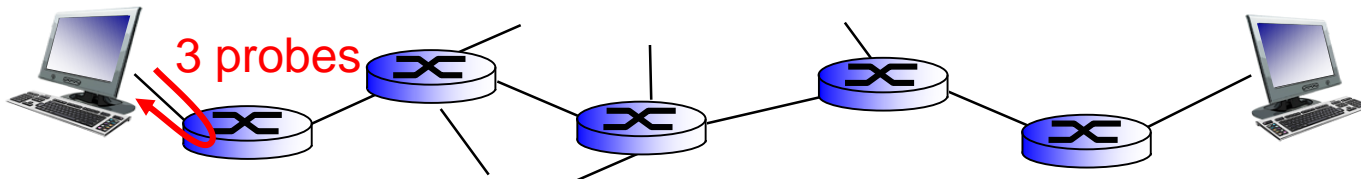
## 2.3 Protokolle und Dienste

## 2.4 Sockets

## 2.5 Aufbau des Internets

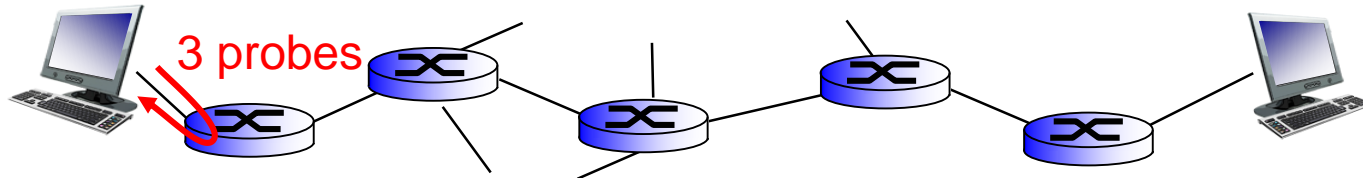
## 2.6 Zusammenfassung

- 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.

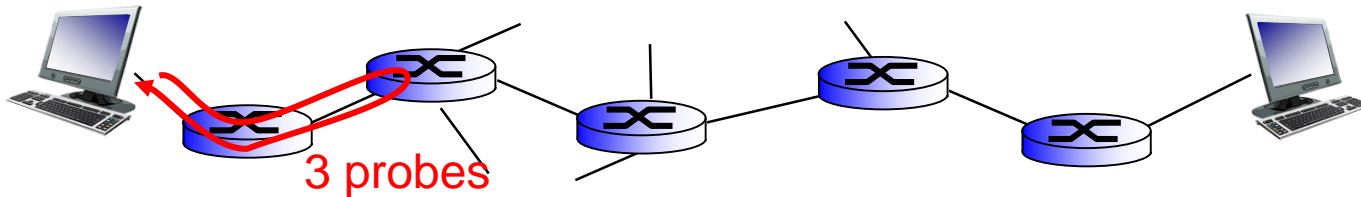




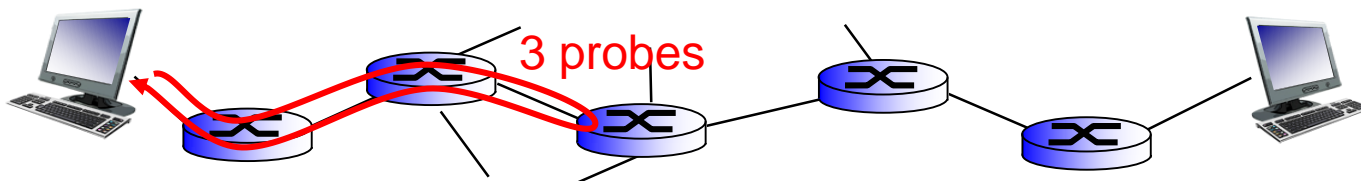
- Schritt 1: senden 3 Ping-Pakete (Probes) mit TTL=1; 1. Router antwortet



- 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 cs.umass.edu (Kurose)

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

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

Hop	RTT1	RTT2	RTT3	Router	IP
1	5 ms	9 ms	9 ms	konstanz-rz-1-10ge-0-1-0-1	belwue.net [129.143.47.77]
2	4 ms	9 ms	9 ms	tuebingen-wae-1-10ge-0-1-0-0	belwue.net [129.143.59.73]
3	4 ms	9 ms	9 ms	stuttgart-al30-1-10ge-0-2-0-2	belwue.net [129.143.57.85]
4	9 ms	9 ms	9 ms	stuttgart-nwz-1-10ge-0-0-0-0	belwue.net [129.143.57.2]
5	15 ms	15 ms	13 ms	cr-erl1-te0-7-0-14	x-win.dfn.de [188.1.233.229]
6	20 ms	23 ms	24 ms	cr-tub1-hundredgige0-1-0-0-7	x-win.dfn.de [188.1.144.185]
7	29 ms	33 ms	34 ms	dfn.mx1.ams.nl.geant.net	[62.40.112.145]
8	107 ms	112 ms	108 ms	xe-0-3-0.102.rtr.newy32	aaa.net.internet2.edu [198.71.45.236]
9	109 ms	112 ms	113 ms	nox300gw1-vl-110-nox-i2	nox.org [192.5.89.221]
10	110 ms	114 ms	114 ms	nox300gw1-peer-nox-umass-192-5-89-102	nox.org [192.5.89.102]
11	111 ms	115 ms	115 ms	core1-rt-xe-0-0-0.gw.umass.edu	[192.80.83.101]
12	111 ms	114 ms	115 ms	lgrc-rt-106-8-po-10.gw.umass.edu	[128.119.3.32]
13	115 ms	114 ms	115 ms		
14	123 ms	127 ms	126 ms	loki.cs.umass.edu	[128.119.240.93]

Ablaufverfolgung beendet.



- 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://dnschecker.org/asn-whois-lookup.php>
- 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: <https://www.belwue.de/netz/topologie.html>

- 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
    - JPerf ist eine GUI zu iPerf
  - DSL-Speed-Testseiten:
    - Transfer von Datenblöcken unterschiedlicher Größe über TCP zwischen PC und Server
      - Ookla: <https://www.speedtest.net/de>