Willkommen bei Kommunikations- und Netztechnik!

– *Von Kupferkabel, Glasfaser und Mikrowelle*
über Telefon, Ethernet und TCP
zu E-Mail, Webserver und REST.

–

–

Heute: **Wege von A nach B über N oder M** in einem Netz.
Adressen!

Zusammenfassung MAC I

- ALOHA, pure vs. slotted
  - Warum ist slotted ALOHA doppelt so effizienz?
- CSMA: sendet sobald frei; wenn Konflikt warte eine zufällige Zeit
- Länge der Contention Periode $= 2\ \tau$ erklären können
- Beispiel für Kollisionsfreie Protokolle (Bitmap, Token, Countdown)
- Adaptive Tree Walk Protocol
- Switches: cut through vs. store and forward (Längenbyte!)
- Backward Learning in Switches
- Spanning Tree für Switches

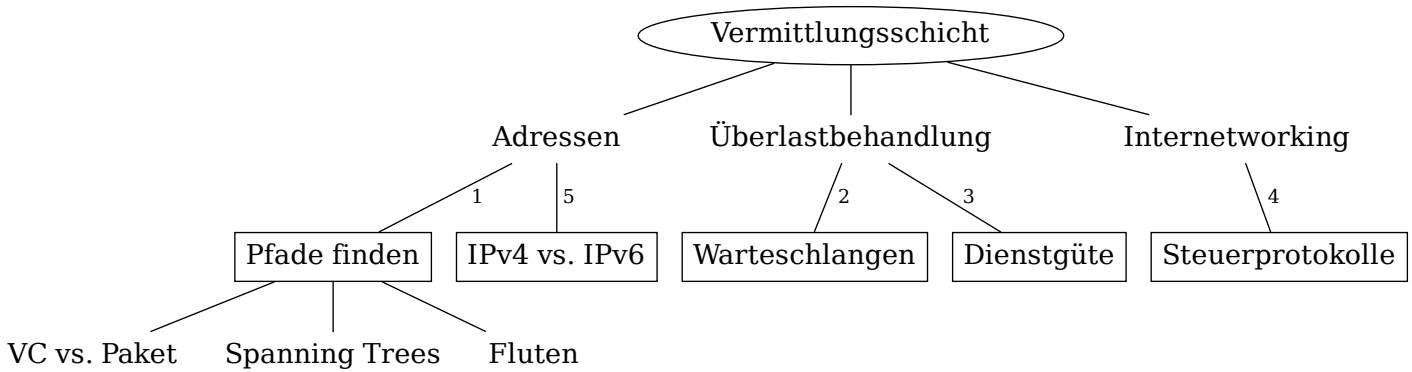
Ablauf heute

- Konzepte
  - Routing-Algorithmen
  - Überlastüberwachung
  - Dienstgüte
  - Internetworking

--- PAUSE ---

- IPv4 und IPv6

Struktur unserer Behandlung der Vermittlungsschicht

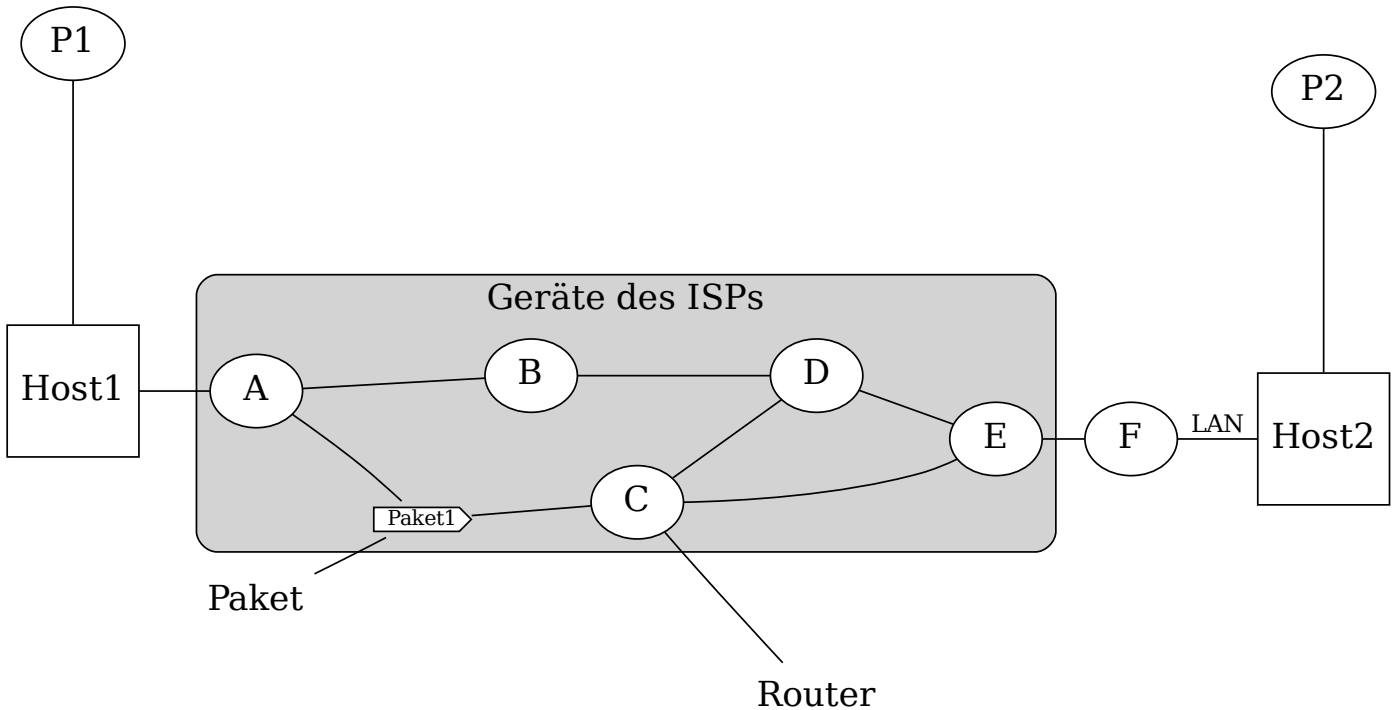


An Flipchart oder Tafel skizzieren.

Ziele heute I

- Sie verstehen Routing mit Quell-Senken Bäumen (spanning trees)
- Sie kennen Unterschiede zwischen VC (virtuelle Verbindung) vs. Paket
- Sie verstehen Fluten und kennen Optimierungen
- Sie kennen die Funktionsweise von Routing-Tabellen
- Sie verstehen Warteschlangen
- Sie kennen Methoden, um Übertragungen zu Drosseln
- Sie kennen die Unterschiede zwischen IPv4 und IPv6

Paketvermittlung: Store-and-Forward



ISP: Internet Service Provider.

Dienste der Vermittlungsschicht für die Transportschicht

Anforderungen:

- Unabhängig von Router-Technologie
- Transportschicht von Routern abgeschirmt
  - Art, Anzahl, Topologie
- Netzaddressen einheitlich nummeriert

Viel Freiheit $\Rightarrow$ 2 Lager!

Dienst-Arten

Internet: Nur Pakete

- SEND PACKET und RECEIVE PACKET
- Robust
- Ende-zu-Ende
- 40 Jahre Erfahrung mit Rechnernetzen

Telcos: Verbindungen

- Dienstgüte
- Interaktiver Echtzeitverkehr
- Dienste:
  - VLAN
  - MPLS (MultiProtocol Label Switching),
- 100 Jahre Erfahrung mit Telefonnetzen

Implementierung

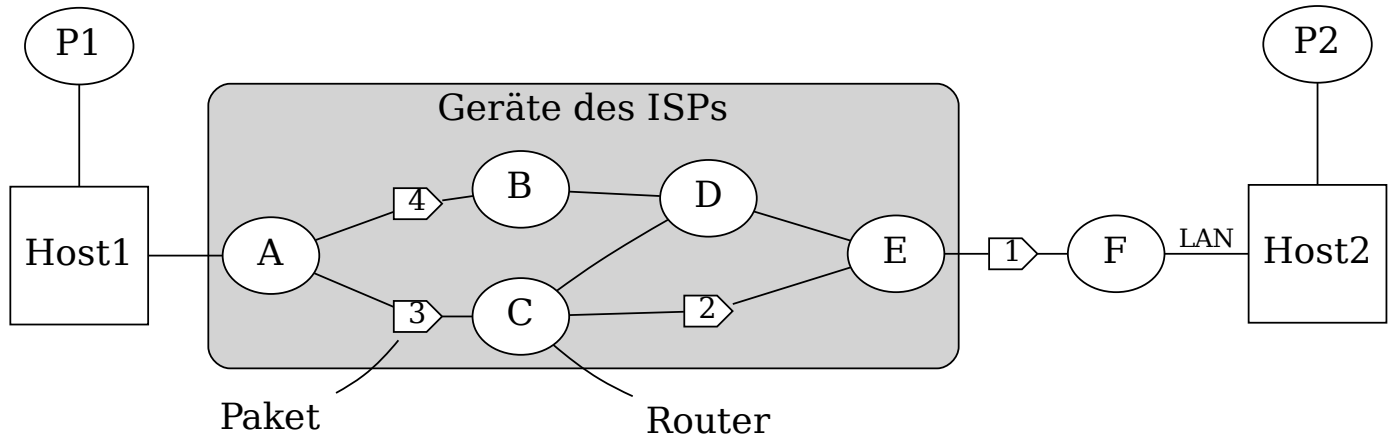
Paketorientiert

- Einzelne Pakete (Datagramme, vgl. Telegramm)
- Unabhängig voneinander
- Kein Einrichtungs-Aufwand
- Jedes Paket die volle Adresse

Verbindungsorientiert

- Pfad von Quelle zu Ziel einrichten
- Virtuelle Verbindung (VC: Virtual Circuit)
- Pakete enthalten VC-Nummer

Datagrammnetz



A (anfang)

A	-
B	B
C	C
D	B
E	C
F	C

A (später)

A	-
B	B
C	C
D	B
E	**B**
F	**B**

Tabelle C

A	A
B	A
C	-
D	E
E	E
F	E

Tabelle E

A	C
B	D
C	C
D	D
E	-
F	F

Verbindungsorientiertes Netz

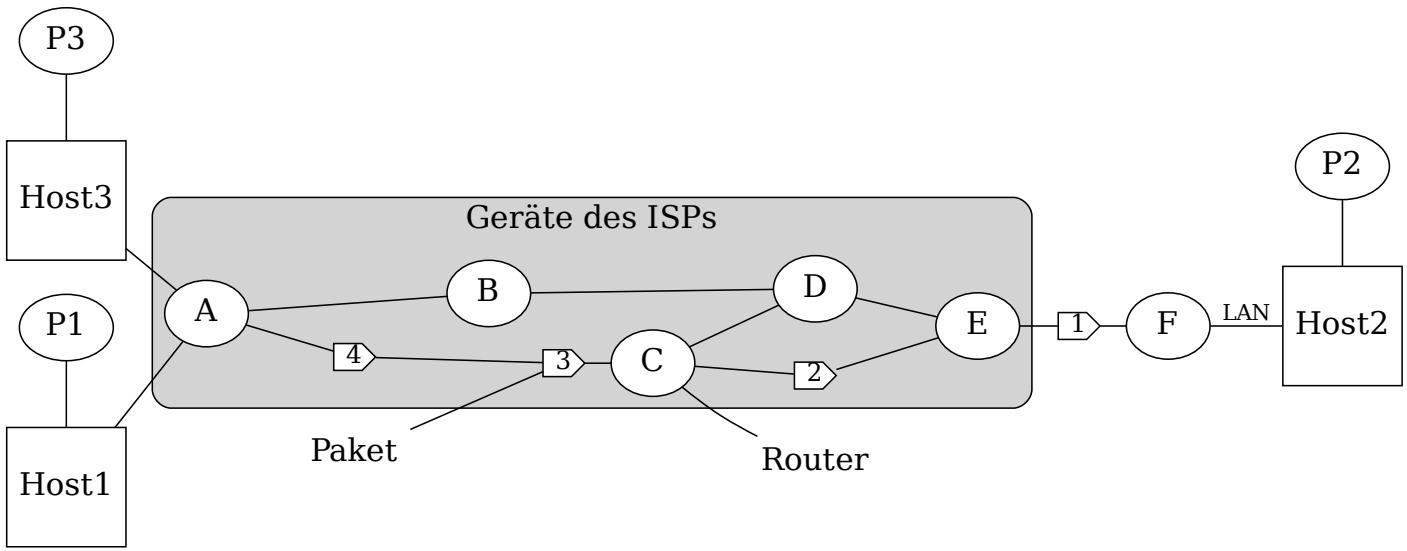


Tabelle von A

H1	1	→	C	1
H3	1	→	C	2*

Tabelle von C

A	1	→	E	1
A	2	→	E	2

Tabelle von E

C	1	→	F	1
C	2	→	F	2

*: $H3_1 \rightarrow C_2$: Label Switching. MPLS: 20 Bit. ISPs nach Datenmenge / Güte.

Vergleich VC-Netze vs. Datagrammnetze

Kriterium	**Datagramm**	**VC**
Aufbau	-	Erforderlich
Adressierung	Ziel- und Quell-Adresse*	VC-Nummer
Zustand	-	Ein Eintrag pro VC
Routing	unabhängig	alle gleich
Router-Ausfall	Paketverlust	Verbindungsverlust
Güte / Last	schwierig	Resourcen reservieren

*: IPv6-Adresse braucht 128 Bit (IPv4 32) — VC-Label bei MPLS nur 20 Bit.

Zusammenfassung

Pakete	**Verbindungen**
Alle unabhängig	Gleiche Route
brauchen weniger Zustand	brauchen weniger Bandbreite
Robuster	können Dienstgüte reservieren

ONLINE-PAUSE

Routing-Algorithmen

„Welche Ausgabeleitung für welches Paket?“

- Datagramm: Für jedes Paket neu entscheiden
- Verbindung: Session-Routing

Wünschenswerte Eigenschaften

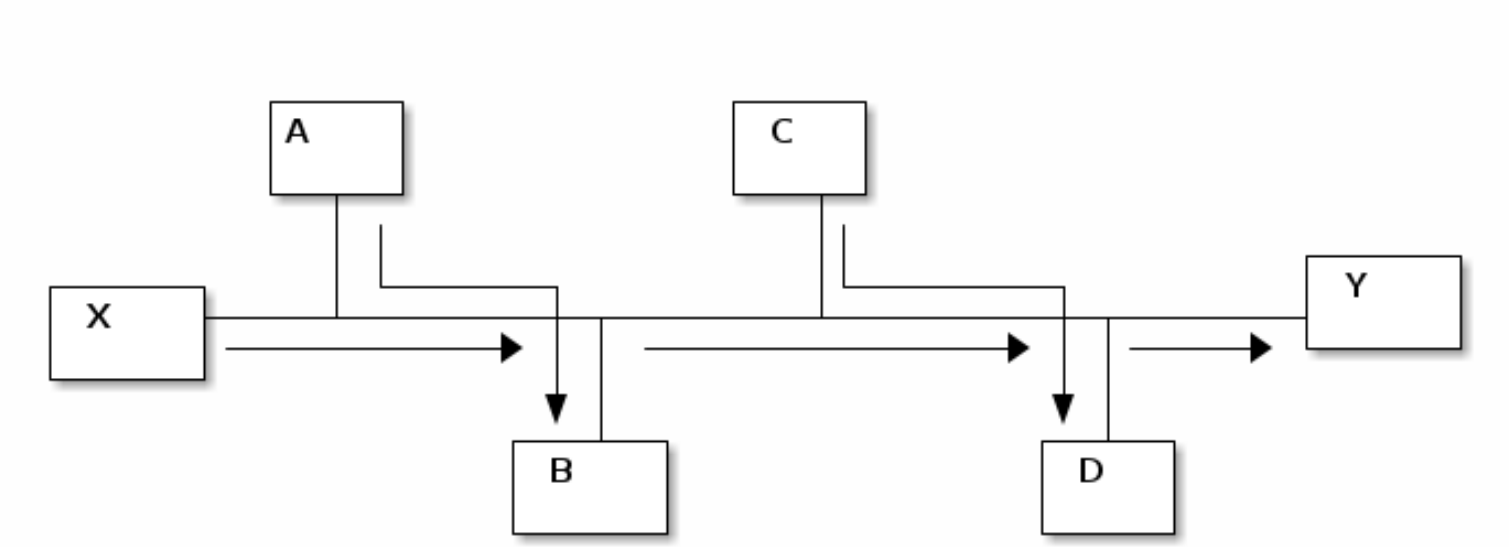
korrekt, einfach klar

robust Ausfälle verkraften

stabil konvergiert in endlicher Zeit

fair, effizient oft im Widerspruch

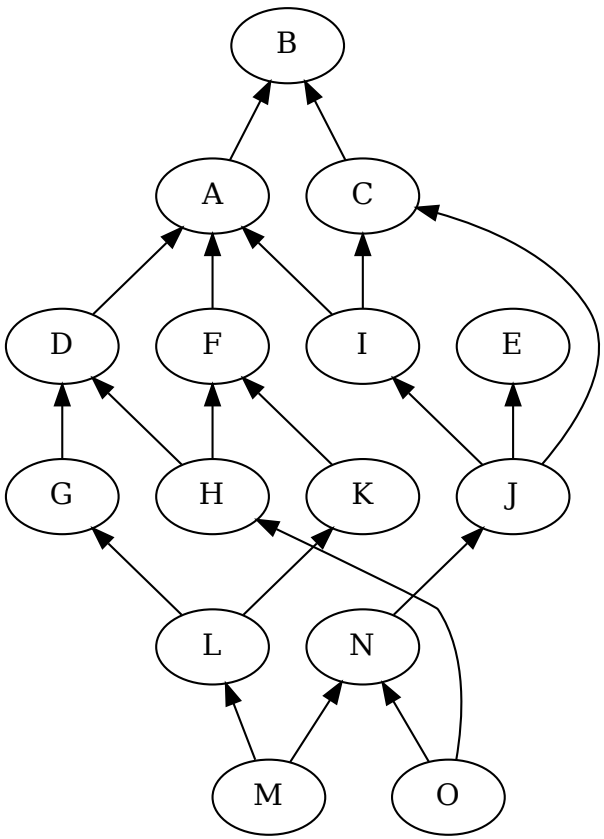
Fairness vs. Effizienz

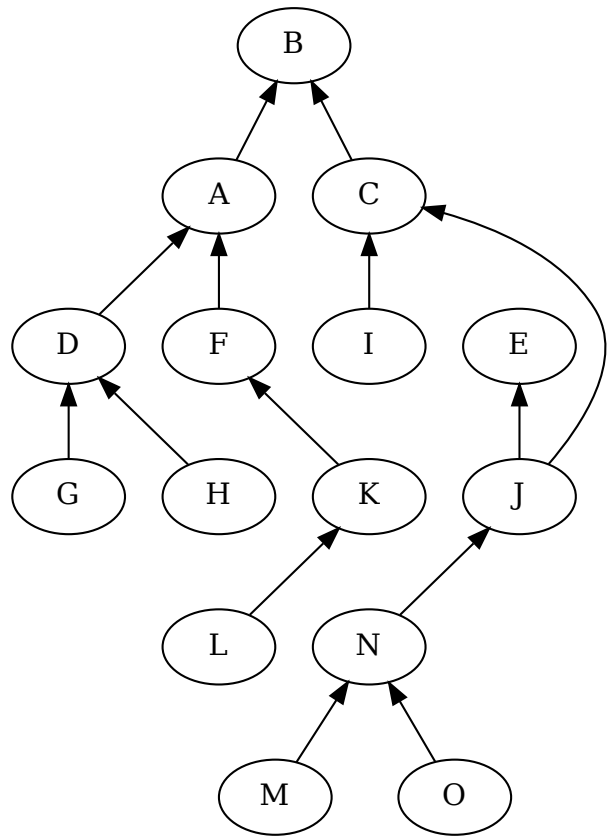


- Maximale Bandbreite?
- Maximale Fairness?
- Wieviel für X/Y?

Optimalität

I→J→K optimal ⇒ J→K optimal ⇒ Quelle-Senke-Baum (sink-tree)





Was ist kurz?

- Entfernung
- Übertragungszeit
- Bandbreite
- Durchschnittsverkehr
- Übertragungskosten
- ...

Fluten

- Einfachster Algorithmus
- Hops to Live
- Schleifen verhindern: Sequenznummern pro Quelle
- Robust
- Extrem Teuer

Die Erste Version von Gnutella (0.4) verwendete Fluten. 0.6 strukturierte sich, um auf 50 Millionen Knoten zu skalieren.

Distanzvektoralgorithmus

Routing Tabelle: Ein Eintrag für jeden Router im Netz

- Ausgangsleitung
- Geschätzte Entfernung ($\rightarrow$ was ist kurz?)

Entfernung:

- Direkte Nachbarn: ECHO-Paket.
- Nachbarn schicken ihre Tabellen.
- Pro Router besten Nachbarn + Kosten merken.

Nachteil: Langsame Information über Ausfall

Link-State-Routing

Ersetzen seit 1979 Distanzvektoralgorithmen.

Andere Namen: IS-IS und OSPF.

Fünf Schritte:

- Nachbarn ermitteln
- Kosten zu Nachbarn ermitteln
- Informationen über die Nachbarn an ALLE schicken
- Wissen von ALLEN empfangen
- Kürzeste Pfade berechnen

Vertrauen der Router untereinander nötig!

Link-State-Paket

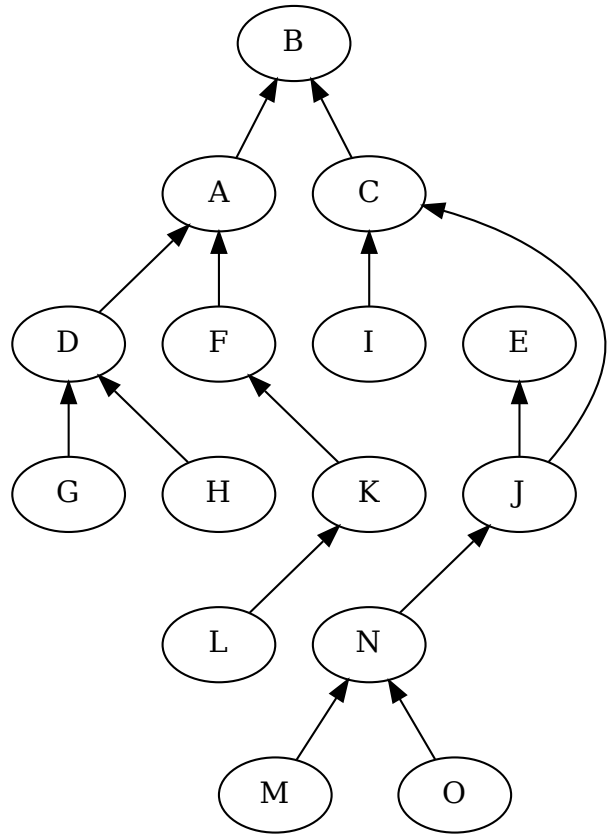
- Identität
- Sequenz-Nummer (lang genug)
- Alter (Lebensdauer, z.B. 10s)
- Nachbarn mit Kosten

Hierarchisches Routing

- Optimierung: Regionen werden zusammengefasst.
- In IP-Adressen mit Netzmasken realisiert.
- Nicht immer die optimalen Pfade, aber berechenbar.

Broadcast

- Fluten
- Reserse Path Forwarding (RPF): Verteile an alle, was von der optimalen Route kommt.
- Sender verwenden Spannbaum (z.B. Quelle-Senke-Baum = sink-tree): Nur auf die Routen schicken, auf denen RPF es annimmt.



*cast

- Multicast: Gestutzte Bäume, enthalten nur Pfade zu Zielen
- Anycast: Bäume in denen mehrere Empfänger als einer betrachtet werden, meist um Anfragen zu schicken; z.B. "von irgendeinem Zeitserver".

Mobiles Routing

- Heimagent (home agent)
- Mobiler Host teilt dem Heimagent die Care-of-Adresse mit
- Heimagent wird zuerst kontaktiert
- Gibt die Care-of-Adresse weiter

Ad-hoc Netze

- Ein eigenes Thema, hoffentlich nächstes Jahr in Verteilten Systemen :-)
- Fluten geht immer

Zusammenfassung

- Fairness vs. Effizienz
- Optimale Verbindungen als Baum (sink tree)
- Distanzvektoralgorithmus erfährt spät von Ausfällen
- Link-State-Routing verteilt vollständige Informationen
- Fluten geht immer (aber selten gut)
  - Broadcast/Multicast/Anycast reduzieren die Pfade beim Fluten

Mobiles Routing: Care-of-Adresse von Heimagent verwaltet.

ONLINE-PAUSE

Sammeln: Wünscht euch Anwendungen zum Besprechen für die letzten beiden Vorlesungen

Überlastkontrolle: Der Warteschlangen-Algorithmus

$$T = \frac{1}{\mu} \times \frac{1}{1-\rho} \tag{1}$$

- T Verzögerung der Pakete (Zeit in Warteschlange)
- μ unbelastete Paketrate (N / Sekunde)
  $\Rightarrow \frac{1}{\mu}$: Verarbeitungszeit eines Pakets
- ρ Auslastung

Bei 50% Last ist die Wahrscheinlichkeit 50%, dass ich bei Ankunft eins Paketes gerade an einem anderen arbeite.

Bei 95% Last, ist die Verzögerung durchschnittlich die 20-fache Verarbeitungsdauer eines Paketes.

Prinzipien der Überlastüberwachung

Von langsam zu schnell:

Provisioning Neue Geräte kaufen

Verkehrsabhängiges Routing Nach Bandbreite und Übertragungsverzögerung. Last verursacht Oszillationen.

Zugangskontrolle Keine E-Mail zur WM

Drosselung Rückmeldung der Überlast

Lastabwurf Pakete verwerfen.

Zugangssteuerung

- Virtuelle Verbindungen ablehnen.
- Einfach: Statische Anteile. Ineffizient durch variable Nutzung.
- Token Bucket: Durchschnittliche Datenrate + begrenzte Burst-Größe.
- Virtuelle Verbindungen über unbelastete Pfade
- Oft mit Dienstgüte: Bevorzugten Diensten

Weswegen Sie bei manchen Providern keine E-Mail schicken können, wenn Ihre Nachbarn alle Fernsehen.

Zugangssteuerung: Token Bucket

Vertraglich vereinbarte Nutzung: Burst + Durchschnittsrate.

Virtueller Zusatzpuffer, um den Knoten die vereinbarte Paketzahl pro Sekunde überschreiten dürfen, ohne aus dem Netz zu fliegen.

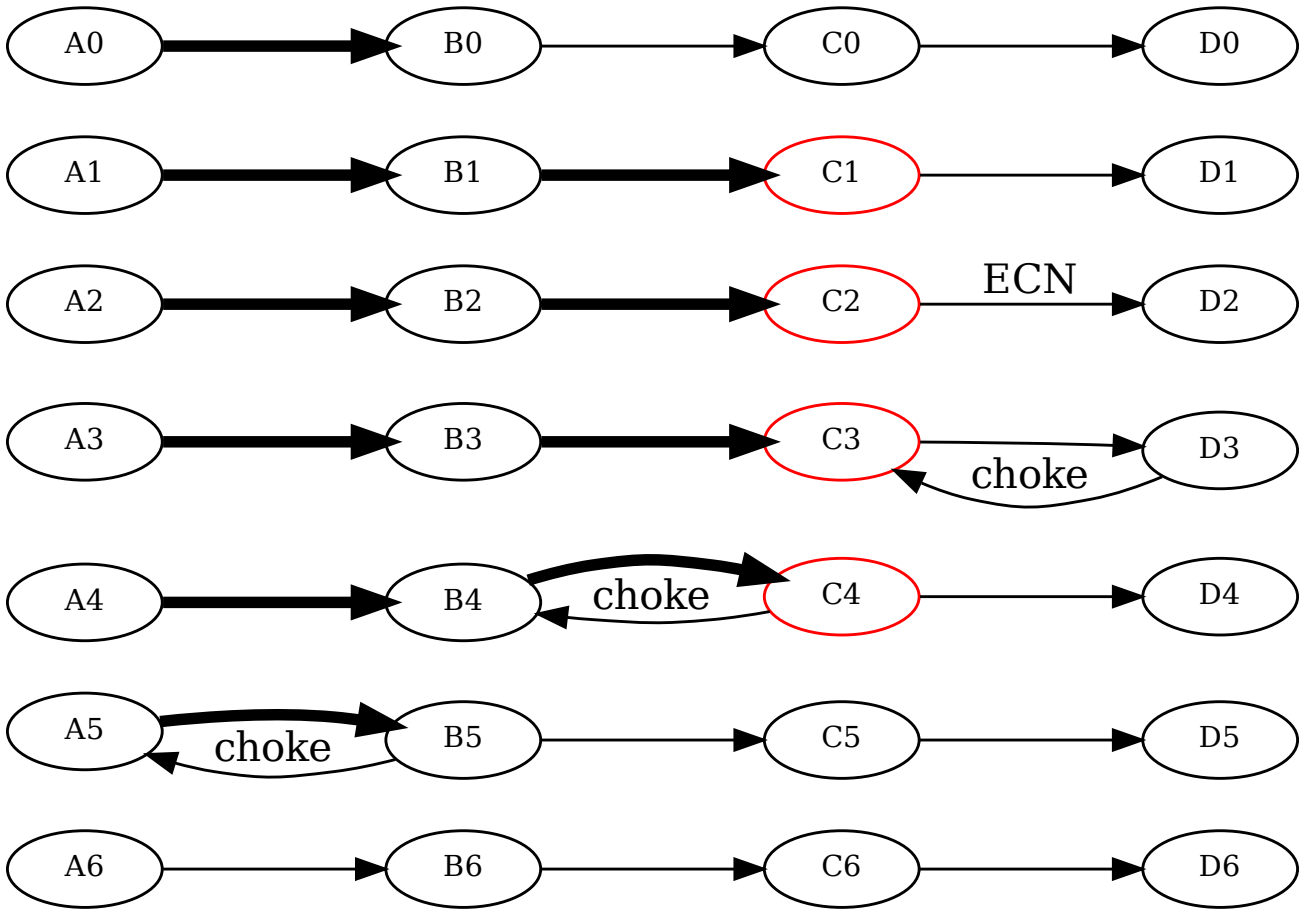
Wird auch zur Definition von Dienstgüte verwendet.

Drosseln

- Überlastung vorhersehen: Verbindungsauslastung, **Pufferfüllstand**, Paketverlust
- Moving Average glättet Bursts.
- Ab Schwellwert:
  - ECN (Explicit Congestion Notification): Zwei Bits in Paket zeigen dem Empfänger Überlast.
  - Empfänger schickt **Choke**-Antwort an Sender
  - Hop-by-hop Rückstau: Knoten auf Zwischenstationen drosseln bereits

Grundprinzip aus p2p-Entwicklung: Alle Warteschlangen sind immer voll.

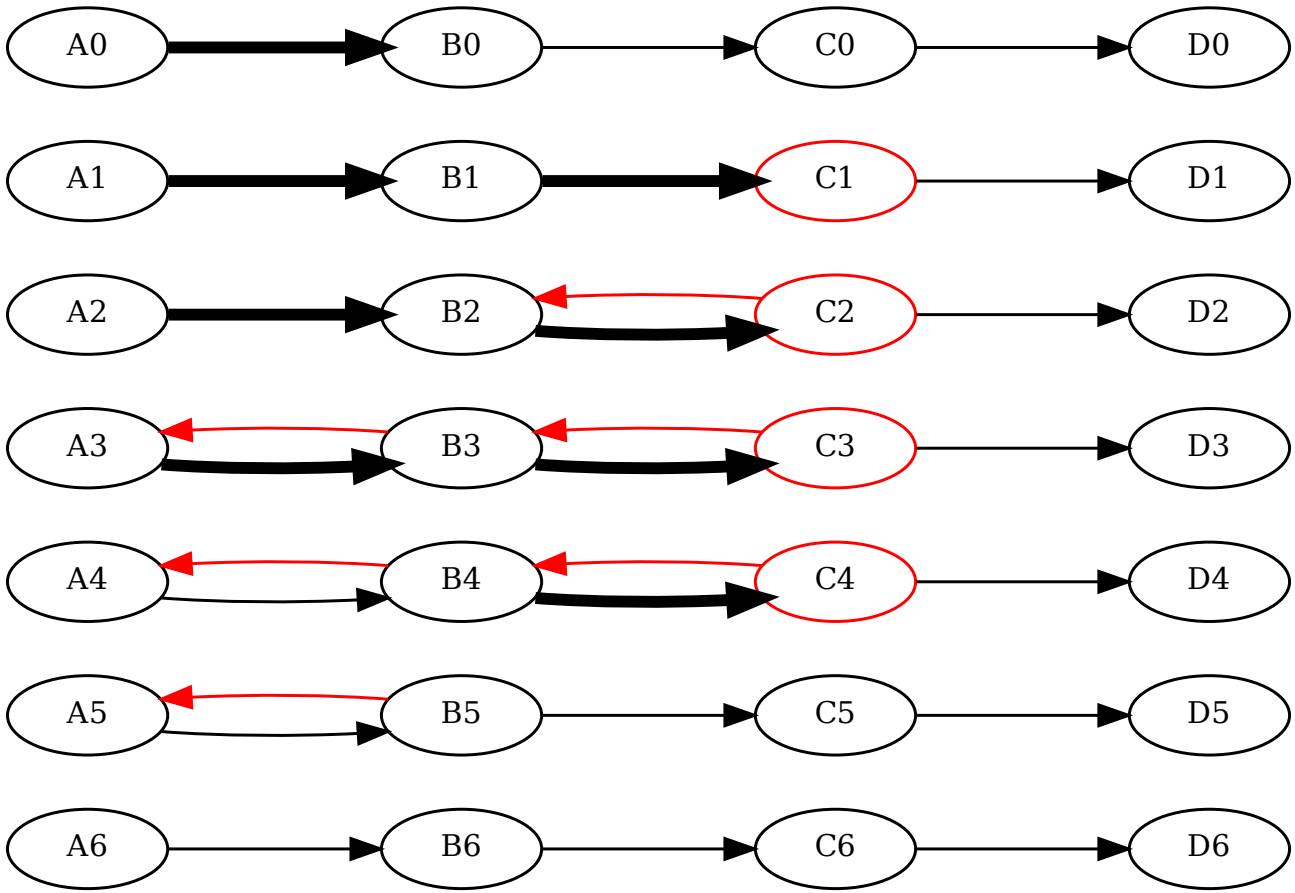
Drosseln, Beispiel



Lastabwurf

- Welches Paket behalten?
  - Lieber alt als neu (Wein): Dateiübertragungen (Neuübertragung)
  - Lieber neu als alt (Milch): Echtzeit-Streaming (Verlust OK)
  - Paketart (z.B. keine key-frames von Videos)
- Früherkennung nach Zufallsprinzip (RED: Random Early Detection)
  - Nur Verlust ist ein zuverlässiger Indikator für Überlast
  - Zufall: Wahrscheinlich vom schnellsten Sender.
  - Wie ein Choke-Paket, nur ohne Paket.
- Ersetzte Choke: `https://www.rfc-editor.org/rfc/rfc1812#section-5.3.6`

Lastabwurf: RED, Beispiel



Zusammenfassung

- Provisioning: Mehr Hardware kaufen
- Route nach Bandbreite und Verzögerung
- Zugangsbegrenzung durch definiertes Verhalten (Token Bucket)
- Drosseln durch Rückmeldung (ECN, Choke)
- Lastabwurf mit Paketauswahl, zufälliges Verwerfen als Indikator für Überlast

Dienstgüte

Unterschiedliche Anforderungen:

- Latenz / Jitter: Interaktives
- Bandbreite: Alles andere

Stichworte:

- Integrated Service: Reservieren von Verbindungen
- Differentiated Service: Klassenbasiert (z.B. Expresspakete)

Priorisierung oft im Streit mit Netzneutralität.

Beschreibung von Dienstgüte

Token Bucket was versprochen wird, was genutzt werden darf

Prioritäten Forderung: für stärkere Annäherung bestimmter Nutzer an Minimal-Latenz und Maximal-Bandbreite.

Flussspezifikation Token-Bucket-Rate, Token-Bucket-Größe, Spitzendatenrate, Minimale Paketgröße, Maximale Paketgröße.[1]

[1] Beispiel nach RFC 2210 und 2211.

Scheduling

Round Robin:

- Ein Paket aus jeder Quelle
- Bevorzugt große Pakete

Fair Queueing:

- Das Paket zuerst, das zuerst fertig gewesen wäre, wenn es bei seiner Ankunftszeit gestartet wäre.
- Annäherung an Multiplexing auf Byte-Ebene.
- Mehr Rechenaufwand, auf sehr schnellen Leitungen zu viel.

Es gibt noch viele weitere.

Zusammenfassung

- Unterschiedliche Anforderungen. V.a. Latenz vs. Bandbreite.
- Beschreibung von Versprechen und erlaubter Nutzung
- Scheduling realisiert die Dienstgüte

Internetworking

Repeater, Hub Analog, verschieben Bits

Bridge, Switch Sicherungsschicht, verbinden ähnliche Netze (z.B. 100 MBit Ethernet mit 10 MBit Ethernet)

Router Pakete aufteilen (fragmentieren), Header tauschen, Pakete verpacken (tunneling).

Internetwork-Routing

- Autonome Systeme
- Interne Informationen verstecken
- Geregelt über Geschäftsvereinbarungen

Internetsteuerprotokolle (Praxis)

ICMP (Internet Control Message Protocol) Zustandsinfos an den Sender, z.B. Ziel unerreichbar, oder ECHO

ARP (Address Resolution Protocol) „Wem gehört diese IP?"

MPLS (Label Switching) Verbindungs-Infos in Zwischen-Netzen vor dem IP-Header: Label + Quality of Service

OSPF (Open Shortest Path First) Link-State-Routing innerhalb von ISP-Netzen. Alternativ IS-IS.

BGP (Border Gateway Protocol) Routing Zwischen ISPs (u.ä.), mit durchsetzung von Regeln, z.B. geschäftliche

Adressen im Internet

IP Internet Protocol

IPv4 32 Bit Addressen, fester 20 Byte Anfang, Options 0-40 Byte.

IPv6 128 Bit Adressen, fester 40 Byte Anfang, “Next Header”-Feld

IPv4

Version	IHL	Diff. Serv.	Total Length			
Identification			x	DF	MF	Fragment Offset
Time to Live	Protocol		Header Checksum			
Source Address						
Destination Address						
Options: 0 bis 10 Zeilen						

Version 4

IHL Header-Länge in Zeilen, 5-15 (4 bit).

Differentiated Services
- Expedited Forwarding (Expresspaket)
- Assured Forwarding (nicht fallen lassen)
- ECN (Überlastung erfahren?)

Total length Header + Daten, bis zu 65535 Byte

IPv4, Fragmentierungs-Felder

Version	IHL	Diff. Serv.	Total Length			
Identification			x	DF	MF	Fragment Offset
Time to Live	Protocol		Header Checksum			
Source Address						
Destination Address						
Options: 0 bis 10 Zeilen						

Identification Fragment-ID

x Ungenutztes Bit

DF Don't Fragment (= Fehler statt Fragmentieren)

MF More Fragments (kommen)

Fragment Offset Index in 8-Byte Blöcken. 13 Bit -f 8192 Fragmente.

IPv4, weitere Felder

Version	IHL	Diff. Serv.	Total Length			
Identification				DF	MF	Fragment Offset
Time to Live		Protocol	Header Checksum			
Source Address						
Destination Address						
Options: 0 bis 10 Zeilen						

Time to Live (TTL) Lebensdauer in Hops, ursprünglich mal Sekunden

Protocol Protokoll-Nummer, z.B. TCP oder UDP (RFC 1700, iana.org)

Header Checksum Bringt die Einerkomplement-Summe des Headers auf 0, bei jeder Übermittlung neu berechnet

Source Address 32 Bit IP-Addresse

Destination Address 32 Bit IP-Addresse

IPv4 Optionen

Bis zu 40 Byte für Optionen:

Security ignoriert

Strict Source Routing Explizite Route in IP-Addressen (debug)

Loose Source Routing Router, die getroffen werden müssen (debug)

Record Route Router hängen ihre Addressen an

Timestamp IP-Addressen + Zeitstempel

heutzutage nur noch schlecht unterstützt.

IPv4-Adressen

Hierarchisch: Netz-Präfix + Host-ID

Abschnitt	Präfix = L Bit (z.B. 24)	32 - L Bit
	Netz	Host
Netzmaske:	11111111111111111111111	00000000

Netz-Adresse: Netzmaske UND IP-Addresse

Router braucht nur den Netz-Abschnitt. Angabe der Addresse mit Präfix-Länge (Anzahl von Einsen).

IPv4-Adresse, Beispiel

192.168.2.105/24

- Netzmaske: /24 = 255.255.255.0
- Netz = 192.168.2.0
- Host-Teil: 105

255.255.255.255 geht an alle Hosts im Netz.
192.168.2.255 an alle im Subnetz 192.168.2.0

Subnetze: Zusätzliche Netzmasken im Host-Teil für lokale Router (nach Außen unsichtbar).

CIDR (classless inter-domain routing)

Automatische Zusammenfassung von Netzmasken.

Früher verwendete Klassen:

- A 0... 128 Netze mit 16 Millionen Hosts
- B 10... 163854 Netze mit 65536 Hosts
- C 110... 2 Millionen Netze mit 256 Hosts
- D 1110... Multicast
- E 1111... reserviert

B ist für die meisten Unternehmen zu groß, C zu klein. CIDR: Dynamische Klassen.

NAT

Router ersetzt IP und Port von Paketen, damit mehrere Rechner nach außen mit der gleichen IP auftreten können.

- Verwaltet Zustand der Verbindungen
- Interne IP ohne Hilfer vom Router nicht von Außen auffindbar. Ein Fluch für Peer-to-Peer-Anwendungen.
- Bricht die Abstraktion (Router sollte nichts von Ports wissen müssen)
- IP-Adressen werden auch innerhalb von Protokollen verwendet

IPv6

Am 1. Februar 2011 vergab IANA die letzten beiden freien IPv4-Adressblöcke 39/8 und 106/8 an das Asia-Pacific Network Information Centre APNIC. Am 3. Februar 2011 starteten IANA und ICANN daraufhin die sog. „Exhaustion Phase“, in der je einer der letzten fünf Adressblöcke für die RIRs reserviert wurde. Damit ist der IPv4-Adresspool der internationalen Vergabestelle IANA ausgeschöpft.

- https://de.wikipedia.org/wiki/Internet_Protocol

2019-10: Etwa 40% der Hosts haben IPv6, Indien 58%[2]
2020-10: Etwa 31% der Google-Nutzer haben IPv6[3], Indien 62%[2]
2022-11: Etwa 40% der Google-Nutzer, Indien 64%[3]

[2] akamai.com/de/de/about/our-thinking/state-of-the-internet-report/state-of-the-internet-ipv6-adoption-visualization.jsp

[3] google.com/intl/en/ipv6/statistics.html#tab=ipv6-adoption

IPv6 Ziele

- Milliarden von Hosts
- Kleinere Routing-Tabellen
- Einfacheres Protokoll für schnellere Router
- Authentifizierung und Datenschutz
- Diensttypen, v.a. Echtzeitdaten
- Umfang von Multicasting angeben
- Mobilität ohne Address-Änderung
- Koexistenz mit IPv4

IPv6 Header

Version	differentiated services	flow label
payload length	next header	hop limit
Source Address		
Source Address		
Source Address		
Source Address		
Destination Address		
Destination Address		
Destination Address		
Destination Address		

IPv6 header-start

Version	differentiated services	flow label
payload length	next header	hop limit

Version 6

Differentiated Services 8 Bit, wie in IPv4

Flow Label ID für Pseudoverbindungen, bis zu 2^{20} unterscheidbare Datenflüsse zwischen Quelle und Ziel

Payload Length Länge **ohne** Header ⇒ Maximal 65535 Byte pro Paket.

Next Header Typ des nächsten Headers **oder** Art des Transportprotokolls (z.B. TCP oder UDP).

Hop Limit Hops to Live

https://tools.ietf.org/html/rfc2460

IPv6 Addressen

- 8 Blöcke mit je 4 Hexadezimalzeichen. Getrennt mit Doppelpunkt
- Der längste Block mit Nullen kann wegelassen werden, im Zweifel der erste.
- Die letzten 4 Blöcke Gerätespezifisch, z.B. die MAC-Addresse

Gültige Addressen:

```
8000:0000:0000:0000:0123:4567:89AB:CDEF
```

```
8000::0123:4567:89AB:CDEF
```

Auch IPv4:

```
::ffff:192.31.20.46
Im Browser: http://[::ffff:192.31.20.46]:8083
```

IPv4 über IPv6

Deprecated: IPv4-compatible IPv6 address

`::23.23.23.23`

"deprecated because the current IPv6 transition mechanisms no longer use these addresses." —
https://datatracker.ietf.org/doc/html/rfc4291

Neu: IPv4-mapped IPv6 addresses

`::ffff:42.42.42.42`

Hier werden IPv4-Pakete ausgetauscht.
https://datatracker.ietf.org/doc/html/rfc4038
Beide nicht in Windows verfügbar.
https://en.wikipedia.org/wiki/IPv6#IPv4-mapped_IPv6_addresses

Multicast

`ff00::/8` + 4 bit scope (z.B. 0 = von IANA[4]), Gültigkeitsbereich (1 interfacelocal → e global).

Beispiele:

`ff01::1` `ff02::1` Alle → Broadcast (01: interface-local, 02: link-local)

`ff01::2`, `ff02::2`, `ff05::2` Alle Router (01: interface, 02: link, 05: site)

[4] IANA: Internet Assigned Numbers Authority

Optionale Header

Hop-By-Hop Options Optionen, für alle IPv6-Geräte, die das Paket durchläuft

Routing Pfad des Paketes durch das Netzwerk beeinflussen, z.B. für Mobile

Fragment Und es gibt es doch :-)

Authentication Header (AH) Optionen für Vertraulichkeit, IPsec

Encapsulating Security Payload (ESP) Daten zur Verschlüsselung des Paketes, IPsec

Destination Options Optionen für den Zielrechner

Mobility Für Mobile IPv6

Ziele erfüllt?

Hosts 7×10^{23} Adressen pro Quadratmeter. Pessimistisch geschätzt: Unterstützt 1000 Geräte pro Quadratmeter.

Routing-Tabellen

Einfacheres Protokoll Fragmentierung und Checksum entfernt, Optionen können Übersprungen werden

Authentifizierung und Datenschutz Durch Erweiterungssheader, aber keine Verschlüsselung

Diensttypen Differentiated Services Header.

Multicasting Flow Label für Pseudoverbindungen

Mobilität *Keine Einigung* $\rightarrow$ *Heimagent, RFC 6275.*

Koexistenz mit IPv4

Nachteile von IPv6

- The internet is middleboxes, old middleboxes[5, 6]
  - Schreiben Pakete beliebig um.
  - Habt ihr in Java Reflection genutzt? Abstraktionsbruch.
- Fragmentierung lässt sich nicht reduzieren

[5] 2022, Middleboxes, Internet architecture: youtu.be/kKZIPeyef0k

[6] heise: Middleboxen verkalken das Internet

Zusammenfassung

IPv4

- 32 Bit Addressen
- Variabler Header: 20 - 60 Byte
- Checksum
- Fragmentierung

IPv6

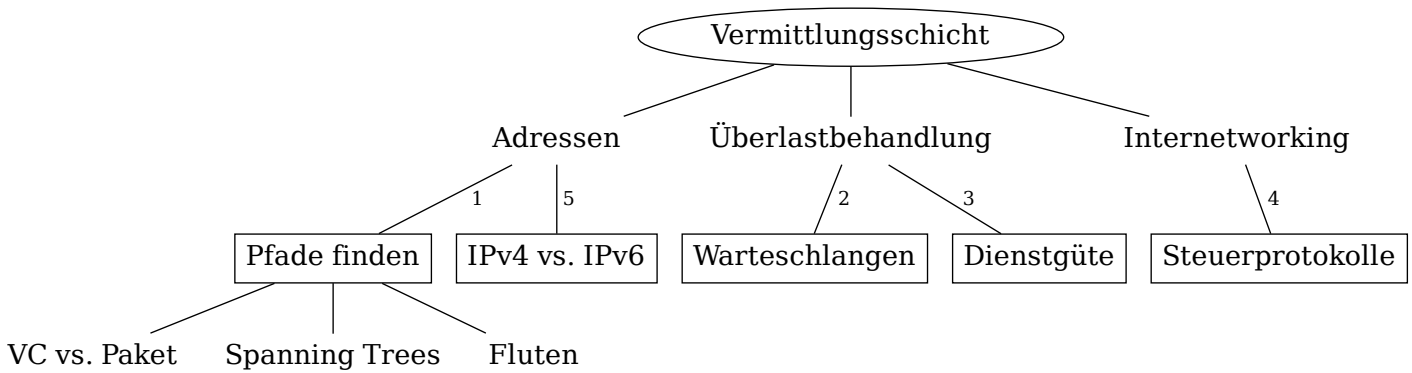
- 128 Bit Addressen
- Fester Header: 40 Byte
- Next Header für Optionen
- Flow Label für Pseudoverbindungen

Fragen für die Prüfung?

Ideensammlung:

- Unterschied IPv4 vs. IPv6: 2^{23} Addressen vs. 2^{128} Addressen, z.B. „ist das IPv4 oder IPv6?“
- Maßnahmen gegen Überlast:
  - Provisioning: Mehr Hardware kaufen
  - Route nach Bandbreite und Verzögerung
  - Zugangsbegrenzung (Token Bucket)
  - Drosseln (ECN, Choke)
  - Lastabwurf, zufälliges Verwerfen
- Token Bucket mit Burst und Bandbreite beschreiben
- Round-Robin und Fair Queueing: Abarbeitungsreihenfolge von Paketen bestimmen, z.B. für 2 Quellen von denen zusammen 3 Pakete kommen.

Zusammenfassung, Struktur



Zusammenfassung, kurz

- VC kleiner als Adresse (aber stup)
- Fairness vs. Effizienz
- Optimale Pfade (Baum),
- Fluten geht immer (aber selten gut)
- Überlast: Provisionierung, Routing, Zugang, Drosselung, Lastabwurf.
- Dienstgüte beschreibt Versprechen und erlaubte Nutzung
- Scheduling realisiert die Dienstgüte

Viel Erfolg auf dem Weg durch den Semester!

Selbststudium diese Woche I

- Simulieren Sie eine Warteschlange mit 3 Quellen und zufällig verteilten Paketgrößen. Plotten Sie die durchschnittliche Zeit bis zur Übertragung als Funktion der Paketgröße. Nutzen Sie dafür Round Robin und Fair Queueing.

Hamming-Beispiele

- Der vorherige 11,7: https://hg.sr.ht/~arnebab/wisp/browse/examples/hamming.w?rev=ad2b1867648a
- Generisch, ineffizient, mit Bugs: https://hg.sr.ht/~arnebab/wisp/browse/examples/hamming-file.w?rev=ad2b1867648a
- 7,4 Hamming Code-Golf: https://codegolf.stackexchange.com/questions/45684/correct-errors-using-hamming7-4

Verweise I

Bilder: