Routing (4)

- Wir kennen bereits einige Routing-Grundverfahren
  - broadcast oder flooding (eigentlich *kein* Routing)
  - Aufbau eines Spannbaums (einige Verbindungen bleiben ungenutzt)
  - Vector-Distance-Verfahren zum Aufbau der Routing-Tabellen
    - da Austausch ganzer Tabellen: eher für kleine Netze geeignet
    - verwendet bei RIP ("Routing Information Protocol", war Teil der BSD-UNIX-Distribution und wurde im Internet eingesetzt)
    - RIP: autonomes Versenden der Tabellen an alle Nachbarn alle 30s; getriggertes Versenden durch Änderung der eigenen Tabelle
    - RIP war relativ primitiv: nur Hopcount als Metrik, max. 15 Hops (neuere Version: RIP-2)

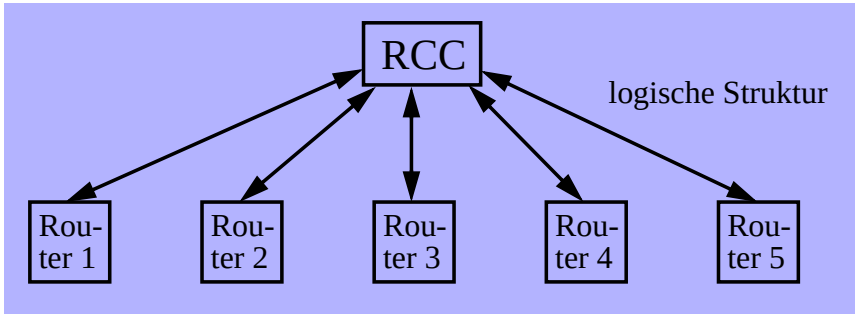
- Beachte: Routing im Internet auf IP-Ebene basiert nur auf netid-Teil der Adresse, nicht auf hostid-Teil
  - gesucht wird in der Routing-Tabelle der Eintrag mit der längsten Übereinstimmung bzgl. der Adresse
  - Bsp.: Wenn die Tabelle einen Eintrag zu 130.32.1.0 und zu 130.32.0.0 enthält, wird bei der Zieladresse 130.32.11.1 der erste Eintrag gewählt

Zentralisierte Leitwegbestimmung

- Knoten senden in periodischen Abständen und bei wichtigen Veränderungen die relevanten Informationen einem "Routing Control Center" (RCC)

- Das RCC berechnet mit globaler Kenntnis optimale Wege (z.B. mit Kürzeste-Wege-Algorithmus)

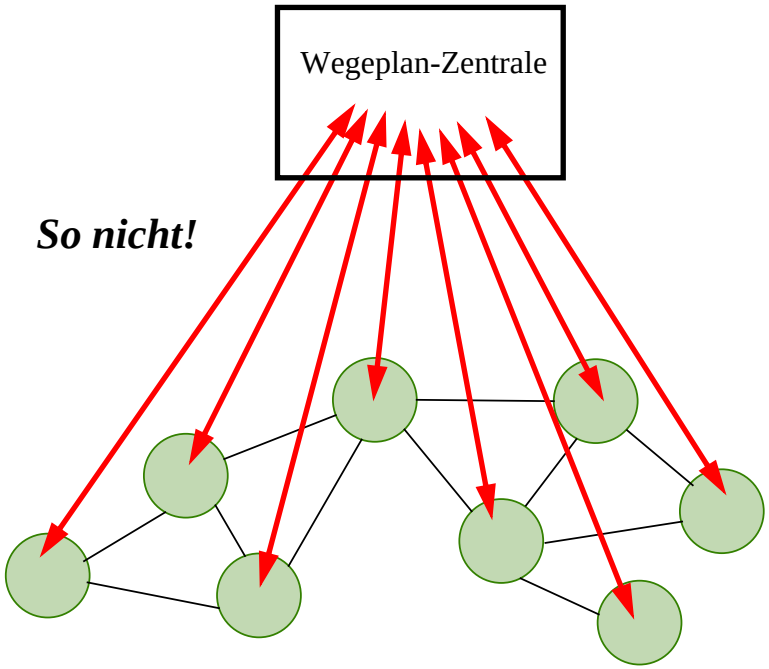
- Relevante Information wird dann wieder vom RCC an die einzelnen Knoten verteilt



- Vorteil: Konsistenz leichter durch globale Sicht

- Aber: Information oft überaltet; Verfahren nicht fehlertolerant; RCC bildet einen Engpass (schlecht skalierbar)

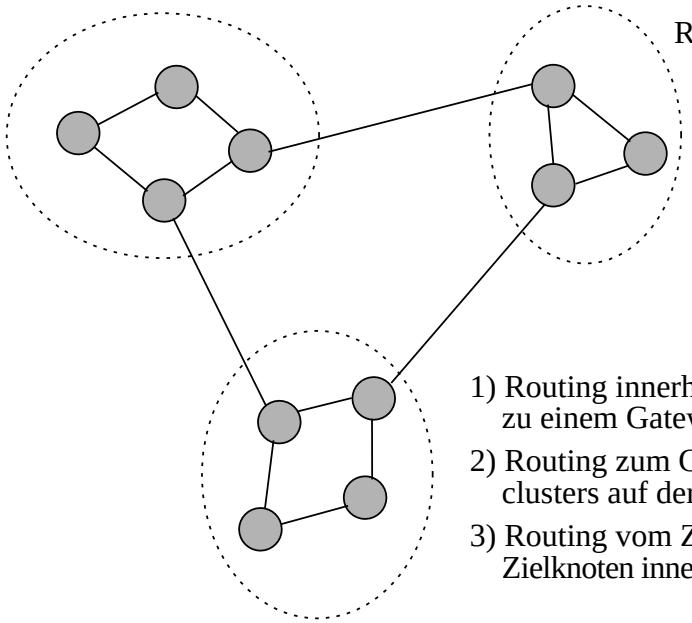
Wieso im Internet dezentral?



- Es gibt keinen Gesamtplan des Internet
- Zentrale Berechnung aller “Wegweiser” würde zu lange dauern
  - alle müssten bereit sein, Wegedaten an die Zentrale zu liefern und Routinginfomation von dort entgegenzunehmen
- Zentrale wäre störungsempfindlich
  - Sicherheitsrisiko, Probleme bei Ausfall oder Fehlverhalten
- Internet ist traditionell “anarchisch” organisiert

Hierarchisches Routing

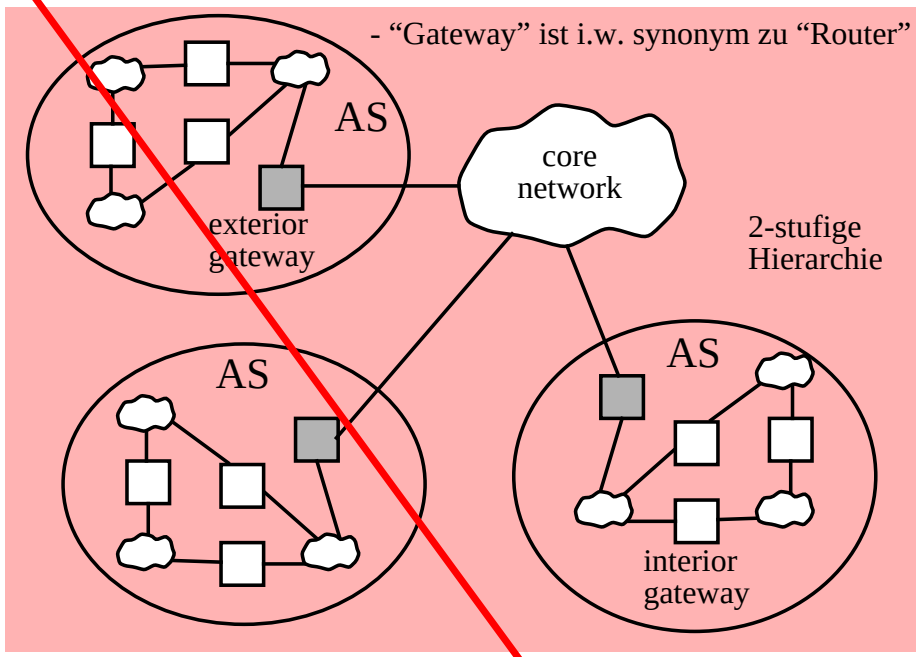
- Hierarchie unterstützt Skalierbarkeit
  - Aufwand für Berechnung der Routing-Tabellen wächst (sonst) nichtlinear mit der Anzahl der Knoten
  - Router brauchen nur ihr eigenes Cluster genau zu kennen, nicht aber die interne Struktur anderer Cluster



1) Routing innerhalb eines Clusters zu einem Gateway (“Repräsentant”)
2) Routing zum Gateway des Ziel-clusters auf der höheren Ebene
3) Routing vom Zielgateway zum Zielknoten innerhalb des Zielclusters

- Vorteile:
  - Topologieänderungen und Änderungen der Kosten haben oft nur lokale Auswirkungen (Update von Tabellen etc.)
  - Routing-Tabellen sind kleiner
- Nachteile:
  - Setzt i.w. eine hierarchischeAdressstruktur voraus (vgl. Telefonnetz)
  - ggf. längere Routen als eigentlich nötig

"Autonome Systeme" (AS) im Internet



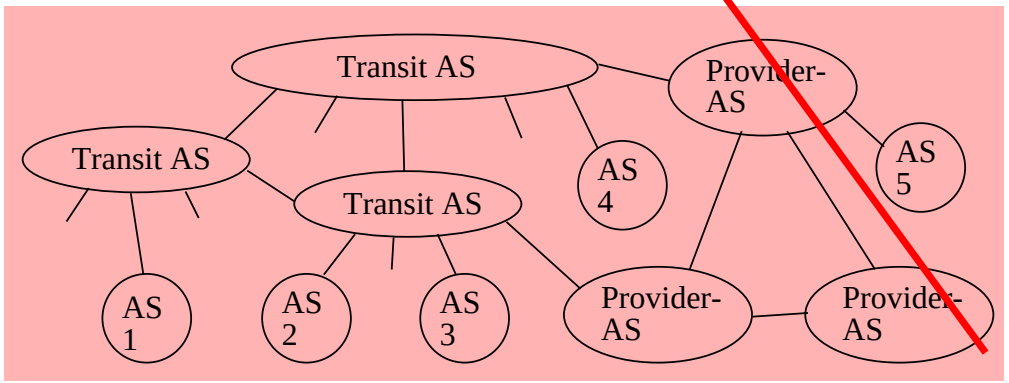
- "Autonomes System": Gruppe von IP-Netzen, welche untereinander ein bestimmtes Routing-Verfahren einsetzen (und i.a. gemeinsam verwaltet werden)
  - jedes AS (bzw. "routing domain") hat eine Nummer (16 Bit)
- AS haben sich i.a. historisch unterschiedlich entwickelt
  - daher oft verschiedene Routing-Strategien in verschiedenen AS
- Interior Gateway Protocol (IGP) für Routing innerhalb
  - z.B. RIP, OSPF ("Open Shortest Path First")
- Exterior Gateway Protocol (EGP) - Standard im Internet
  - derzeit BGP ("Border Gateway Protocol")

Autonome Systeme

- Auszug aus der Liste der Institutionen
  - es gibt derzeit einige tausend autonome Systeme
  - Nummern von der IANA ("Internet Assigned Numbers Authority") verwaltet

11 HARVARD	1744 LOCKHEED
26 CORNELL	1748 FINNAIR
32 STANFORD	1754 DESY-HAMBURG
71 HP	1893 DKRZ-HAMBURG
108 XEROX	2022 SIEMENS
163 IBM-RESEARCH-	2278 FRANCE-TELECOM
294 FRANCE-IP-NET	2535 BP
513 CERN	3680 NOVELL
517 XLINK-UKA	3958 AIRCANADA
553 BELWUE	4179 CITY-OF-LA
680 WIN	4183 COMPUSERVE
715 APPLE	6142 SUN-JAVA
1248 NOKIA	6307 AMERICAN-EXPRESS
1275 DFN	6431 ATT-RESEARCH

- Typischerweise drei Klassen von AS:
  - Institutionen mit einer einzigen Verbindung zum restlichen Graph
  - regionale Internet-Provider; Institutionen mit Peering-Abkommen (Durchleiten von fremdem Datenverkehr i.a. nur beschränkt)
  - "Transit AS": Internet-Backbone hoher Kapazität



Link-State-Routing-Verfahren (1)

- Jeder Knoten (= Router) kennt den Zustand der Verbindung ("link state") zu jedem seiner Nachbarn
  - sowie die Kosten (delay etc.)

- Dieses Wissen wird allen(!) Knoten im Netz mitgeteilt
  - nicht nur den Nachbarn
  - Information wird dazu in ein Link-State-Paket (LSP) verpackt

- LSP enthält
  - ID des Erzeugerknotens
  - Kosten (bzw. ∞ bei "down") zu allen direkten Nachbarn
  - Sequenznummer
  - Time-to-live (TTL)

- LSP wird mit "reliable flooding" verteilt
  - eigenes LSP periodisch versenden (Sequenznummer jeweils erhöhen)
  - ein Empfänger merkt sich das jeweils neueste eines Absenders
  - ein neueres empfangenes LSP wird allen (anderen) Nachbarn mitgeteilt
  - alle LSPs werden mit einer Ack-Nachricht bestätigt

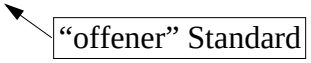
- Alterungsprozess:
  - TTL jedes gespeicherten LSPs wird langsam heruntergezählt
  - TTL wird stets dekrementiert, bevor LSP weitergeleitet wird
  - falls TTL 0 wird: LSP löschen, aber an Nachbarn weiterleiten (diese löschen dann auch etc.)

Link-State-Routing-Verfahren (2)

- Nachdem ein Knoten ein LSP aller empfangen hat, berechnet er die jeweils kürzeste Route zu den anderen
  - z.B. mit Kürzeste-Wege-Algorithmus nach Dijkstra
  - woher aber ist die Gesamtmenge aller Knoten (von denen ein LSP eintreffen muss) bekannt?

- Eigenschaften
  - reagiert relativ schnell auf Topologieänderungen
  - Nachrichtenoverhead in der Praxis erstaunlich gut verkraftbar
  - Informationsmenge pro Knoten ist allerdings beachtlich (ein LSP für jeden anderen Knoten)
  - Berechnung ist bei grossen Topologien relativ aufwendig (allerdings ist beim "shortest path first"-Verfahren die Nutzung grösserer Teile einer früheren Berechnung bei einer Neuberechnung möglich)
  - Skalierbarkeit?

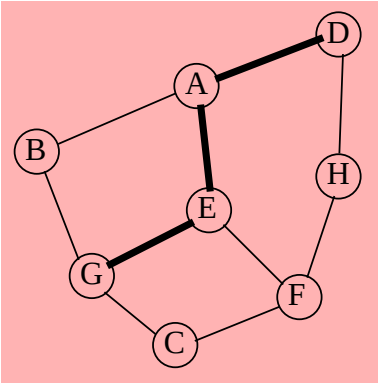
- Link-State-Verfahren wird in der Praxis oft benutzt, z.B. beim OSPF ("Open Shortest Path First") im Internet



- OSPF:
  - oft für das Routing *innerhalb* von Internet-Teilbereichen verwendet
  - einfache Authentifizierungsmöglichkeit (Password; Prüfsumme)
  - Ermittlung von up/down durch periodisches Versenden von zu bestätigenden IP-Kontrollpaketen an alle Nachbarn
  - Kennenlernen der Nachbaridentitäten durch Hello-Pakete (z.B. nach einem reboot)

BGP (Border Gateway Protocol)

- Routing zwischen verschiedenen “Autonomen Systemen” (“AS”)
  - AS = Internet-Teilbereich: Gruppe gemeinsam verwalteter IP-Netze
- Jeder BGP-Router eines AS kennt stets die beste Route als vollständigen Pfad zu jedem anderen BGP-Router



Beispiel: G kennt zu D die Route GEAD

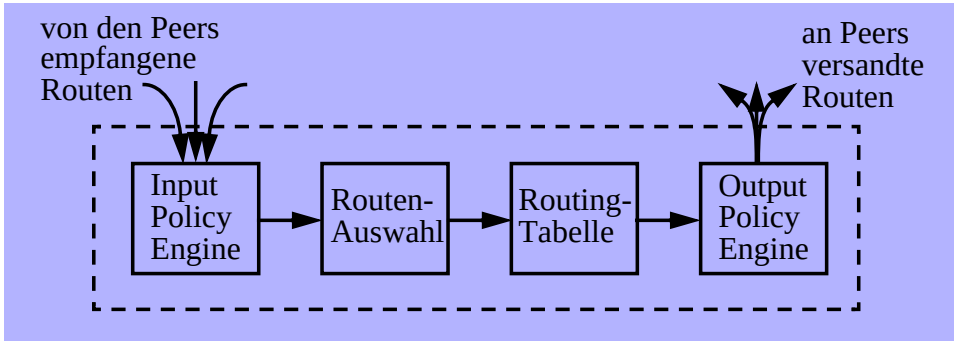
Falls nun EA gestört ist, kann z.B. mit GBAD oder GCFHD ein alternativer Weg zu D gewählt werden, sofern die Routing-Policy dies gestattet

Dies spätestens dann, wenn G von seinen beiden Nachbarn deren Pfade BAE bzw. CFHD zu D erfährt

- Ein BGP-Router eines AS (der “speaker”) informiert periodisch alle “Peers” (i.a. Nachbarn) über seine Routen
  - über TCP, port 179
  - alternative Routen zu einem Ziel werden unter Berücksichtigung der Routing-Politik bewertet (∞, falls eine Bedingung nicht erfüllt ist), die jeweils “billigste” Route wird dann verwendet
  - i.a. kein Austausch ganzer Tabellen, sondern Pfade neuer Routen inkrementell an die Peers weitermelden
  - eigene AS-Nummer beim Weiterleiten an das Pfadende anhängen, Zyklentest bei Empfang eines Pfades (taucht eigene AS auf?)
  - Widerrufen von Routing-Pfaden ist möglich
  - KEEPALIVE-Nachricht zu Nachbarn typw. jede Sekunde

BGP: Policy-Routing

- BGP kann eine “Routing-Policy” berücksichtigen
  - z.B. “Pakete von X nach Y dürfen nicht über Z laufen” (wobei Z eine fremde Firma, das Ausland, ein teurer Provider... sein kann)
  - z.B. “A leitet nur Pakete von P oder Q weiter”
  - Routing-Policy muss manuell spezifiziert werden
  - Routing-Policy meist durch ökonomische Bedingungen oder Sicherheitsüberlegungen (z.B. “route stets über Firewall”) bestimmt
- Man spricht allgemein von “Policy-Routing”, wenn neben der Zieladresse auch andere Aspekte in die Routingentscheidung eingehen (z.B. Absenderadresse)
- Modell eines BGP-Routers:

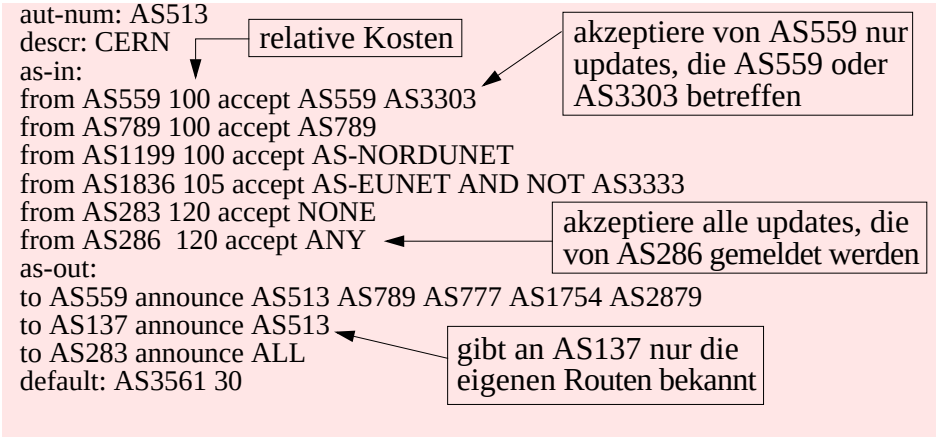


- Policy Engines können Routen filtern und Attribute manipulieren
  - Routen sind typischerweise durch reguläre Ausdrücke über AS-Nummern oder IP-Nummern spezifiziert
  - damit Änderung der Priorität bzgl. einiger Routen oder Nachbarn; Zurückhalten gewisser Informationen...
  - i.a. unterstützt durch Filter im Firewall

Austausch von Routing-Information

- Wer an wen Routing-Information (update-Nachrichten) sendet und empfängt, muss (manuell) spezifiziert werden
  - hierzu existieren einfache formale Sprachen (z.B. RIPE-181 oder RPSL)

- Die Update-Policy für ein AS sieht dann z.B. so aus:



Einige Routing-Probleme

1) Zyklen durch fehlerhafte Konfiguration etc.:

```
...
 4  ZR-Frankfurt1.WiN-IP.DFN.DE (188.1.11.113)  4 ms  4 ms  4 ms
 5  ZR-Koeln1.WiN-IP.DFN.DE (188.1.144.33)  9 ms  9 ms
 6  ZR-Hannover1.WiN-IP.DFN.DE (188.1.144.25)  20 ms  24 ms *
 7  ZR-Hamburg1.WiN-IP.DFN.DE (188.1.144.21)  27 ms  20 ms  26 ms
 8  ZR-Berlin1.WiN-IP.DFN.DE (188.1.144.17)  46 ms  38 ms  49 ms
 9  ZR-Hamburg1.WiN-IP.DFN.DE (188.1.144.18)  34 ms  27 ms  29 ms
10  ZR-Berlin1.WiN-IP.DFN.DE (188.1.144.17)  56 ms  59 ms  54 ms
11  ZR-Hamburg1.WiN-IP.DFN.DE (188.1.144.18)  34 ms *  32 ms
12  ZR-Berlin1.WiN-IP.DFN.DE (188.1.144.17)  63 ms *  56 ms
13  ZR-Hamburg1.WiN-IP.DFN.DE (188.1.144.18)  39 ms  41 ms  40 ms
14  ZR-Berlin1.WiN-IP.DFN.DE (188.1.144.17)  50 ms  57 ms  51 ms
15  ZR-Hamburg1.WiN-IP.DFN.DE (188.1.144.18)  43 ms  43 ms  45 ms
16  ZR-Berlin1.WiN-IP.DFN.DE (188.1.144.17)  67 ms  68 ms  67 ms
17  ZR-Hamburg1.WiN-IP.DFN.DE (188.1.144.18)  55 ms  54 ms  47 ms
18  ZR-Berlin1.WiN-IP.DFN.DE (188.1.144.17)  75 ms  69 ms  75 ms
...
```

2) Stark wachsender Datenverkehr an zentralen Routern

3) Wachsende Routing-Tabellen
   - Verdoppelung alle 10 Monate zwischen 1988 und 1991
   - Gegenmassnahmen wie Aggregation von Routen bremsen das Wachstum
   - Anzahl der Routen im Internet für einen typischen Router:
     - 1988: 350
     - 1992: 8500
     - 1994: 20000
     - 1995: 34000
     - 1996: 42000
     - 1999: 80000

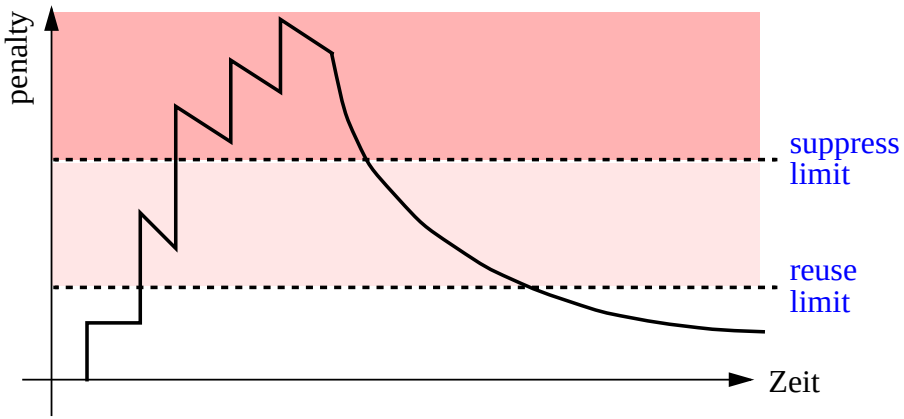
Routenfluktuation ("flapping")

- Routen können ggf. immer wieder verschwinden und kurz danach wieder auftauchen

- Ursachen vielfältig
  - z.B. Probleme der Hardware, in der Software, bei der Konfiguration

- Schafft Probleme
  - Probleme eines AS pflanzen sich ggf. im Internet fort
  - Overhead; Belastung der Router und des gesamten Internet
  - dadurch ggf. Nicht-Erreichbarkeit von Teilen des Internet

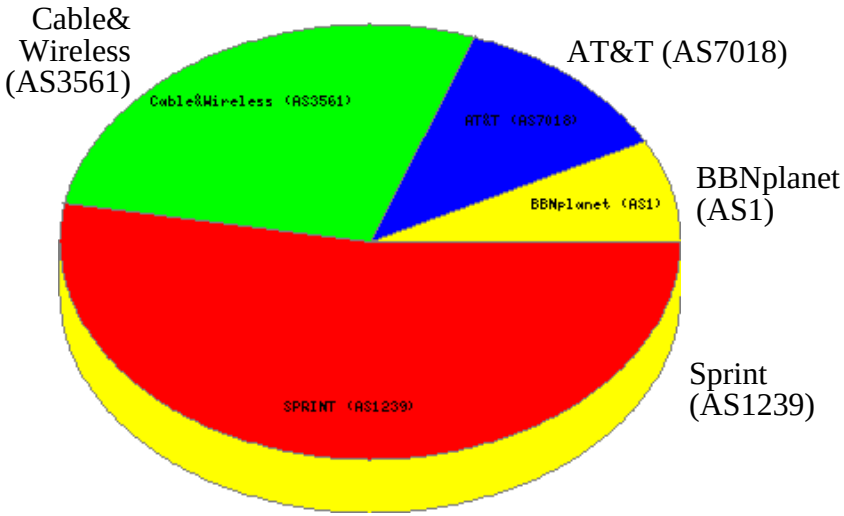
- Daher "Dämpfung" notwendig
  - Route oder Änderung einer Route unterdrücken, bis Vertrauen in die Stabilität der Route besteht



- Strafpunkte für eine Route
  - Route unterdrücken, wenn "penalty" grösser als suppress limit
  - Route erst bei Unterschreitung des reuse limits freigeben (Hysterese)
  - Strafpunkte verfallen exponentiell (Halbwertszeit konfigurierbar)

Routenfluktuation (2)

01/10/00 Internet Routing Instability at the Mae-East NAP
Instability Contributed by each Peer



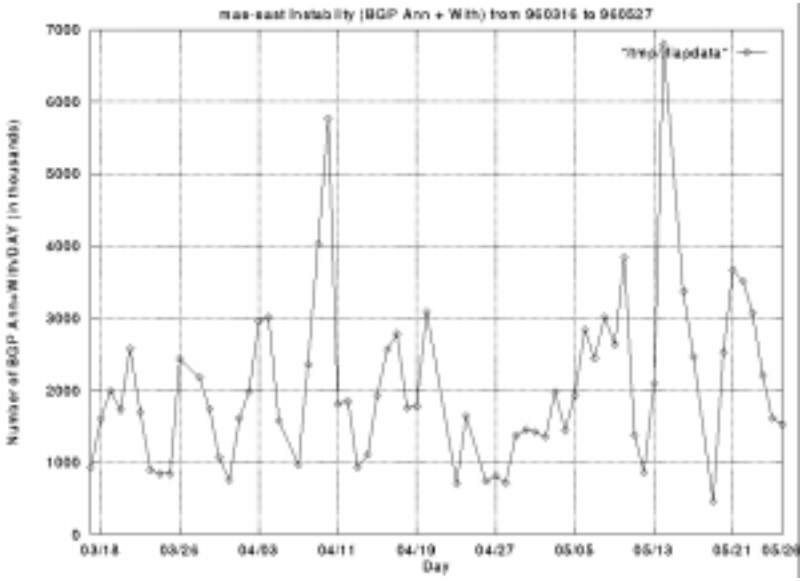
April 25, 1997. A misconfigured router maintained by a small Virginia service provider injected an incorrect routing map into the global Internet. This map indicated that the Virginia company's network provided optimal connectivity to all Internet destinations. Internet providers that accepted this map automatically diverted all of their traffic to the Virginia provider. The resulting network congestion, instability, and overload of Internet router table memory effectively shut down most of the major Internet backbones for up to two hours. Incorrect published contact information for operations staff, and lack of procedures for inter-provider coordination exacerbated the problem.

Quelle: http://www.eecs.umich.edu/techreports/cse/1998/CSE-TR-382-98.pdf

Routenfluktuation (3)

1996

The Internet core currently exchanges an average of **five million routing updates a day**. This graph shows the aggregate number of BGP route announcement and withdraws over a three month period. Note that April 10 shows a spike of ten million BGP updates and May 14 shows a peak of 7 million updates. Both of these large peaks were caused by single, small Internet service providers. These peaks show that individiual ISPs can have an inordinate affect on the level of routing instability in the core Internet.



Routers “flap,” or lose their ability to route efficiently, for a number of reasons, including configuration errors, network links going up and down, software bugs, and other network problems. All these problems cause a very high number of routing updates to be passed to the core Internet exchange points. Route flaps spread from router to router and propagate throughout the network; at the extreme, route flaps have led to the collapse of portions of the Internet.

Routing instability, or the rapid fluction of Internet routing information appears to be growing at an exponetial rate in the Internet “core.” This is a significant problem because severe levels of routing instability can lead to poor network performance (packet loss, latency, and interruptions of service).

Quelle: http://compute.merit.edu/analysis/routing.html

Router: Anforderungen und Trends

- Klassische Aufgaben eines Routers:
  - Matching des längsten Präfixes in der Routing-Tabelle
  - TTL-Feld dekrementieren
  - Header des Ebene 2-Protokolls bzgl. des Next-hop-Routers anfügen und ggf. Prüfsummen etc. der Ebene 2 berechnen und anfügen
  - ggf. ICMP-Fehlermeldung generieren, Statistiken führen

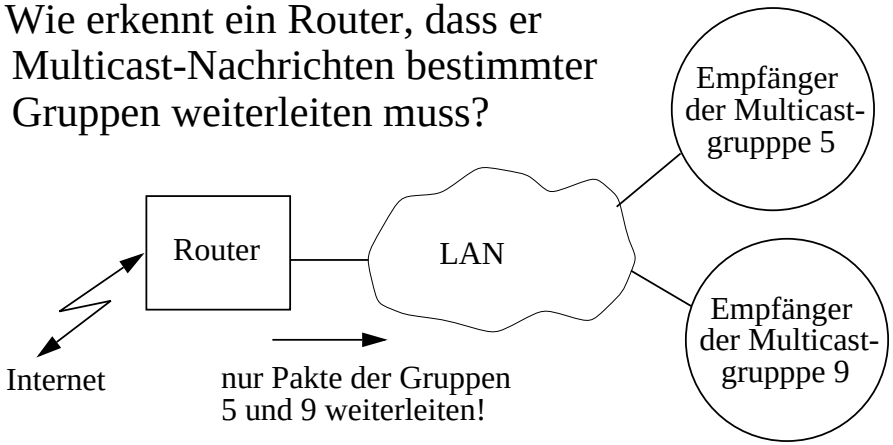
- Neue Anforderungen
  - neuere Services wie Multicast, Sprache, Quality of Service, Sicherheit etc. erfordern ggf. (bei hoher Datenrate!) eine Analyse und spezielle Behandlung einzelner Datenpakete

- Router sind kritische Internet-Infrastrukturkomponenten
  - zu verarbeitendes Datenvolumen erhöht sich schnell; bei 10 Gb/s und 1000 Byte-Paketen z.B. 1 000 000 lookups pro Sekunde notwendig
  - zukünftig Terabit-Netze --> Gefahr, dass Router einen Engpass bilden
  - lookup bisher mit geeigneten Datenstrukturen / Algorithmen (z.B. Hashing oder Patricia-Bäume); neuere Verfahren sind patentiert oder nicht veröffentlicht: Minimierung der Speicherzugriffe, oft unter Verwendung sehr grosser Speicher
  - sehr leistungsfähige Backplane (z.B. crossbar statt Bus, oder shared memory für Input- und Outputports, wobei nur der header bewegt wird)
  - Verwendung schneller Puffer und Caches
  - modularer Aufbau; “line cards” für diverse Protokolle und Medien
  - zukünftig vermehrter Einsatz von Parallelität (bis auf die Bitebene mittels bit slicing und data striping)

IP-Multicasting

- Einzelnes IP-Datagramm wird an mehr als eine Station adressiert
  - die ersten vier Bits des Adressfelds im Header sind 1110
  - danach folgt eine 28-Bit-lange Multicast-Gruppennummer

- Wie erkennt ein Router, dass er Multicast-Nachrichten bestimmter Gruppen weiterleiten muss?



- IGMP (Internet Group Management Protocol)
  - spezielle IP-Pakete mit Wert = 2 im Protocol-Feld des IP-Headers
  - Router sendet periodisch Gruppenzugehörigkeitsanfragen an alle Rechner des LAN (per Broadcast)
  - durch TTL = 1 wird die Anfrage nur im LAN verbreitet
  - Rechner antworten mit ihrer Gruppennummer nach einer zufälligen Zeit per LAN-Broadcast
  - andere Rechner dieser Gruppe im LAN, die das empfangen, antworten dann nicht mehr
  - Router erhält alle Antworten und aktualisiert seine Routingtabelle; mehrfach nicht bestätigte Gruppen werden gelöscht
  - tritt ein Rechner einer Gruppe bei, so sendet er sofort eine entsprechende Mitteilung an alle Router im LAN

TCP (Transmission Control Protocol)

- Verbindungsorientiert (Ebene 4); setzt auf IP auf

- Motivation: IP hat einige “Anwendungsschwächen”:
  - Verlust von Paketen
  - ggf. Umordnung von Paketen (parallele Übertragungswege!)
  - beschränkte Länge von Paketen

- Fast jede Anwendung muss daher gewisse (i.w. die gleichen!) Massnahmen treffen

- TCP erledigt dies weitgehend für “alle”
  - i.w. mit sliding window
  - gute Parameterwahl (timeouts, Fenstergrössen) nicht einfach, da das Protokoll für sehr unterschiedliche Situationen effizient sein soll (LAN / WAN; 10kb/s - 1Gb/s; kleine / grosse Zeitschwankungen...)

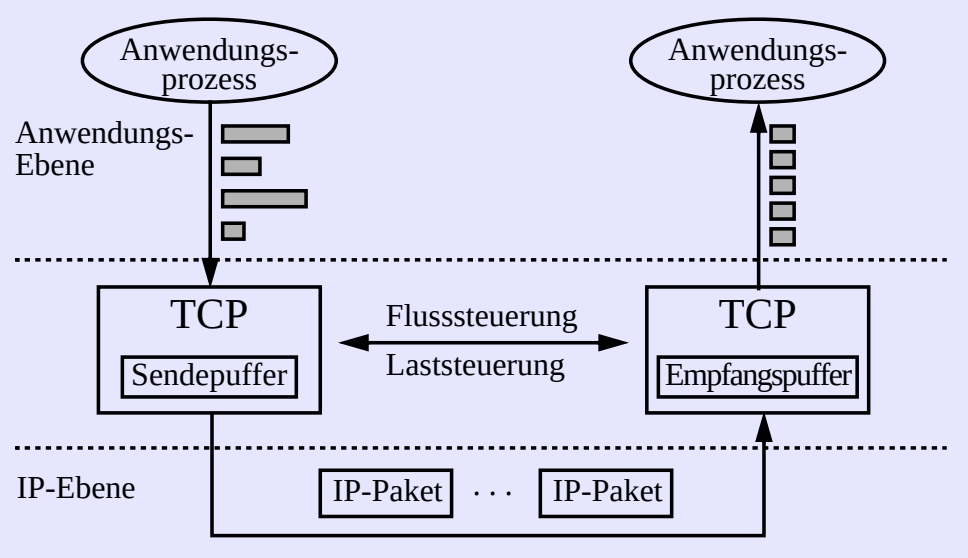
- TCP bietet einen zuverlässigen Kanal für einen Bytestrom
  - zwischen zwei Ports (=Kommunikationsendpunkte)
  - voll duplex
  - unstrukturiert (nur Bytes; Nachrichtengrenzen gehen verloren), z.B.:

    send (170 Bytes); send (230 Bytes);
    receive (20 Bytes); receive (270 Bytes); receive (110 Bytes);

  - vgl. dies mit Schreiben / späteres Lesen von Bytes auf eine Platte

- Verbindungsaufbau vor Datenübertragung notwendig
  - Verbindungsabbau danach

Bytestrom



- *Flusssteuerung*: Sender soll *Empfänger* nicht überlasten
- *Laststeuerung* (congestion control): Sender soll das *Netz* nicht verstopfen
- Datenstrom wird von TCP in *IP-Pakete* aufgeteilt
  - nach dem IP-Header folgt vor den Daten ein TCP-Header
- Ein einzelnes Datenbyte verursacht so ggf. einen Over-head von einigen zig Bytes (z.B. Echo bei einem Editor)!
  - Gegenmassnahme: z.B. kurze (!) Verzögerung beim Sender, um ggf. weitere Bytes der Anwendung in das gleiche IP-Paket zu packen

Die Socket-Programmierschnittstelle

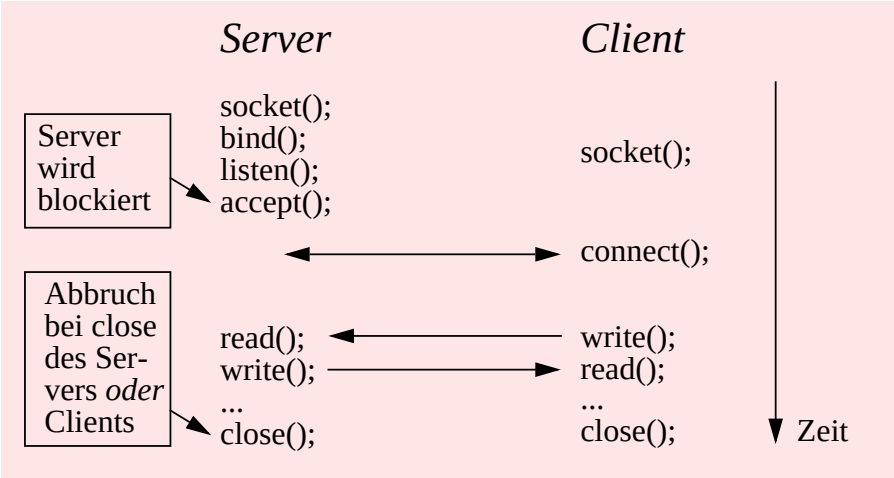
- Zu TCP (bzw. UDP) gibt es keine festgelegten “APIs”
- Bei UNIX sind dafür “sockets” als Zugangspunkte zum Transportsystem entstanden
  - etwas modernere Alternative: TLI (Transport Layer Interface)
  - sockets in leicht veränderter Form auch unter MS-Windows
- Semantik eines sockets: analog zu Datei-Ein/Ausgabe
  - ist insbesondere bidirektional (“schreiben” und “lesen”)
  - ein socket kann aber auch mit mehreren Prozessen verbunden sein
- Programmiersprachliche Einbindung (z.B. in C)
  - sockets werden wie Variablen behandelt (können Namen bekommen)
  - Beispiel in C (Erzeugen eines sockets):

```
int s;
s = socket(int PF_INET, int SOCK_STREAM, 0);
```

“Family”: Internet oder nur lokale Domäne

“Type”:Angabe, ob TCP verwendet (“stream”); oder UDP (“datagram”)

Client-Server mit Sockets (Prinzip)



- Voraussetzung: Client "kennt" die IP-Adresse des Servers sowie die Portnummer (des Dienstes)
  - muss beim connect angegeben werden
- Mit "listen" richtet der Server eine Warteschlange für Client-connect-Anforderungen ein
  - Auszug aus der Beschreibung: *"If a connection request arrives with the queue full, tcp will retry the connection. If the backlog is not cleared by the time the tcp times out, the connect will fail"*
- Accept / connect implementieren ein "Rendezvous"
  - mittels des 3-fach-Handshake von TCP
  - bei "connect" muss der Server bereits listen / accept ausgeführt haben
- Rückgabewerte bei read bzw. write: Anzahl der tatsächlich gesendeten / empfangenen Bytes

Sockets unter Java

- Paket java.net.* enthält u.a. die Klasse "Socket"
- Stream- (verbindungsorientiert) bzw. Datagrammsockets
- Beispiel (für einen Client):

```
DataInputStream in;
PrintStream out;
Socket server;
...
server = new Socket(getCodeBase().getHost(),7);
  // Klasse Socket besitzt Methoden
  // getInputStream bzw. getOutputStream, hier
  // Konversion zu DataInputStream / PrintStream:
in = new DataInputStream(server.getInputStream());
out = new PrintStream(server.getOutputStream());
...
  // Etwas an den Echo-Server senden:
out.println(...)
...
  // Vom Echo-Server empfangen; vielleicht
  // am besten in einem anderen Thread:
String line;
while((line = in.readLine()) != null)
  // line ausgeben
...
server.close;
```

Herstellen einer Verbindung

Hostname

Echo-Port

Port Nummer 7 sendet alles zurück

- Zusätzlich: Fehlerbedingungen mit Exceptions behandeln ("try"; "catch")
  - z.B. "UnknownHostException" beim Gründen eines Socket

Echo-Port war beliebt bei Hackern: denial of service-Attacke durch Endlosschleife (gefälschter Absender127.0.0.1 als "lokaler Rechner" im IP-Paket)

Port-Nummern im Internet

- Ports sind logische Kommunikationsendpunkte; mit einem Port ist ein Programm (“Service”) verbunden
- Der Sender (“Client”) muss den “richtigen” Port kennen, den er adressieren soll
- Ports sind numeriert
  - einige (0 - 1023) sind weltweit eindeutig (“well known port”)
  - andere werden dynamisch alloziert (und müssen den Kommunikationspartnern zuvor mitgeteilt werden)
- Die Nummern der “well known ports” werden von der IANA (Internet Assigned Numbers Authority) verwaltet

> The IANA is chartered to act as the clearinghouse to assign and coordinate the use of numerous Internet protocol parameters.
>
> The common use of the Internet protocols by the Internet community requires that the particular values used in these parameter fields be assigned uniquely. It is the task of the IANA to make those unique assignments as requested and to maintain a registry of the currently assigned values.

```
echo          7    Echo
daytime      13    Daytime
ftp-data     20    File Transfer[Default Data]
ftp          21    File Transfer[Control]
telnet       23    Telnet
smtp         25    Simple Mail Transfer
domain       53    Domain Name Server
finger       79    Finger
http         80    World Wide Web HTTP
kerberos     88    Kerberos
pop3        110    Post Office Protocol - Version 3
nntp        119    Network News Transfer Protocol
ntp         123    Network Time Protocol
bgp         179    Border Gateway Protocol
irc         194    Internet Relay Chat Protocol
imap3       220    Interactive Mail Access Protocol
```

Auszug aus der Liste der well-known ports