Verteilte Systeme SS2002: Gruppenkommunikation

Marcel Waldvogel

Übersicht

- Paare und Gruppen
- Vor-/Nachteile von Gruppen
- Protokolle
  - Multicast etc.
- Applikationen
  - News
  - Zeit

Kommunikationsarten

- Paarweise
  - Client-Server
  - Peer-to-Peer
- Gruppenweise
  - Peer-to-Peer
  - Mit Koordinator

Client-Server-Paar

- Anzahl Klienten
  - Dedizierte Verbindung
    - Polling, Sensor
  - Variabler einzelner Klient
    - X11 Window Manager
  - Konstante Anzahl Klienten
    - Messzentrale
  - Variable Anzahl Klienten
    - Typischer Fall
- Koordination beim Server
  - Einzelner Worker-Prozess
  - Mehrere: Gegenseitiger Ausschluss

Peer-to-Peer-Paar

- Beispiel
  - Primary-/Backup-Server
  - Protokolle: TCP, . . .
- Gegenseitiger Ausschluss nötig?
  - Koordination?
  - Zuverlässigkeit?
- Authentisierung
  - IP-Adresse
  - Kryptografisch

Wieso Gruppen?

- Vorteile
  - Metcalfe's Law:
    Der Nutzen von Kommunikationsdiensten steigt quadratisch mit der Anzahl Benutzer (und damit proportional zu den möglichen Verbindungen)
    - Geschlossene vs. offene Standards; Beispiel WWW
  - The Computational Grid
  - Skalierung!
- Nachteile
  - Koordination
  - Netzwerk (Bandwidth, Delay, Loss)
  - Mehrere Rechner (Administration, Ausfall, ...)

Kategorien von Gruppen

- Welche Probleme tauchen auf bei mehreren
  - Klienten?
  - Servern?
  - Peers?
- Abhängig von Applikation?

Skalierung

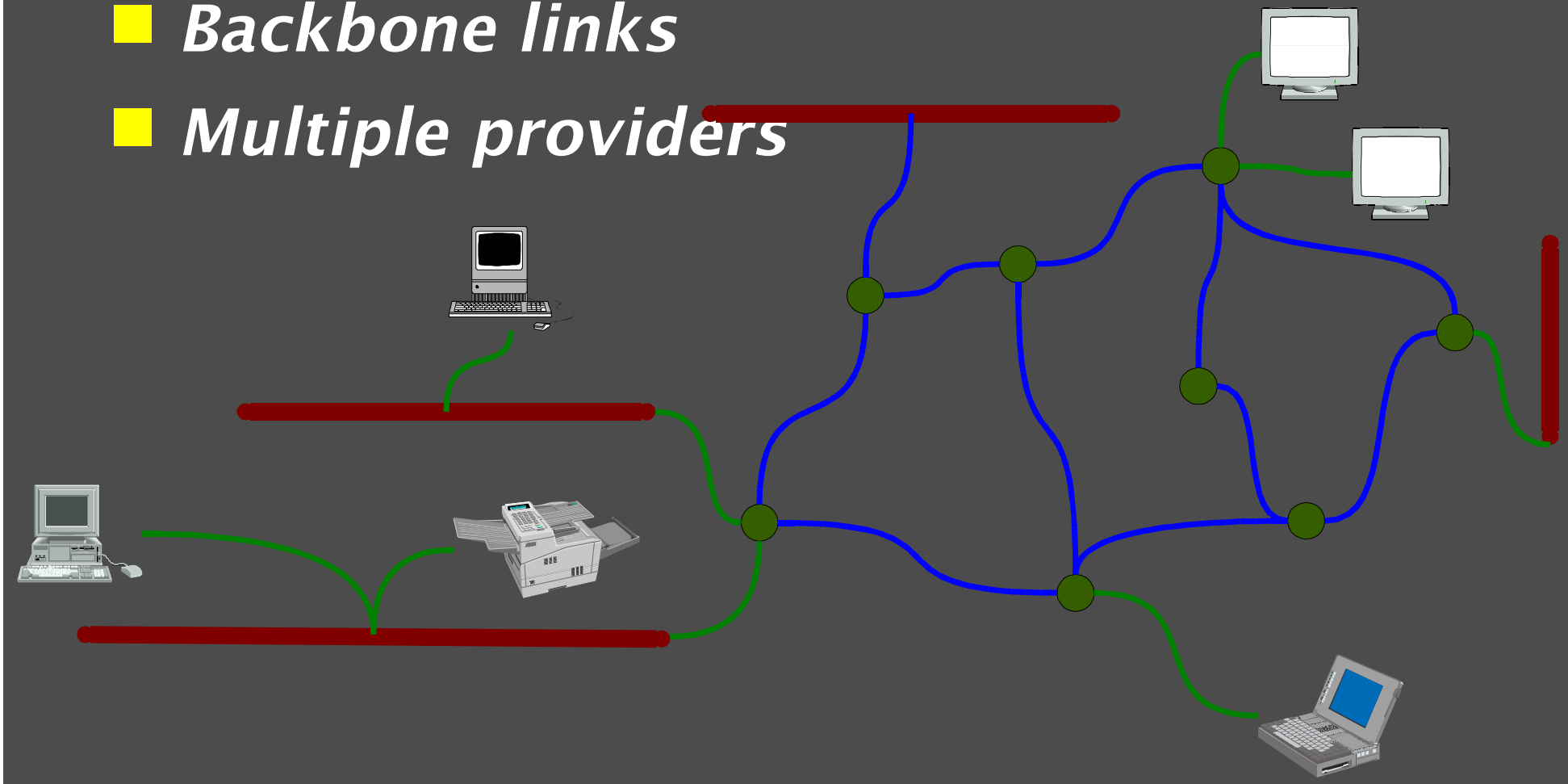
- Parallelität
  - Analog zu Parallelrechnern
- Enge vs. lose Koppelung
  - distributed.net
    - Client-Server
  - Datenbanken
    - Peer

Mechanismen

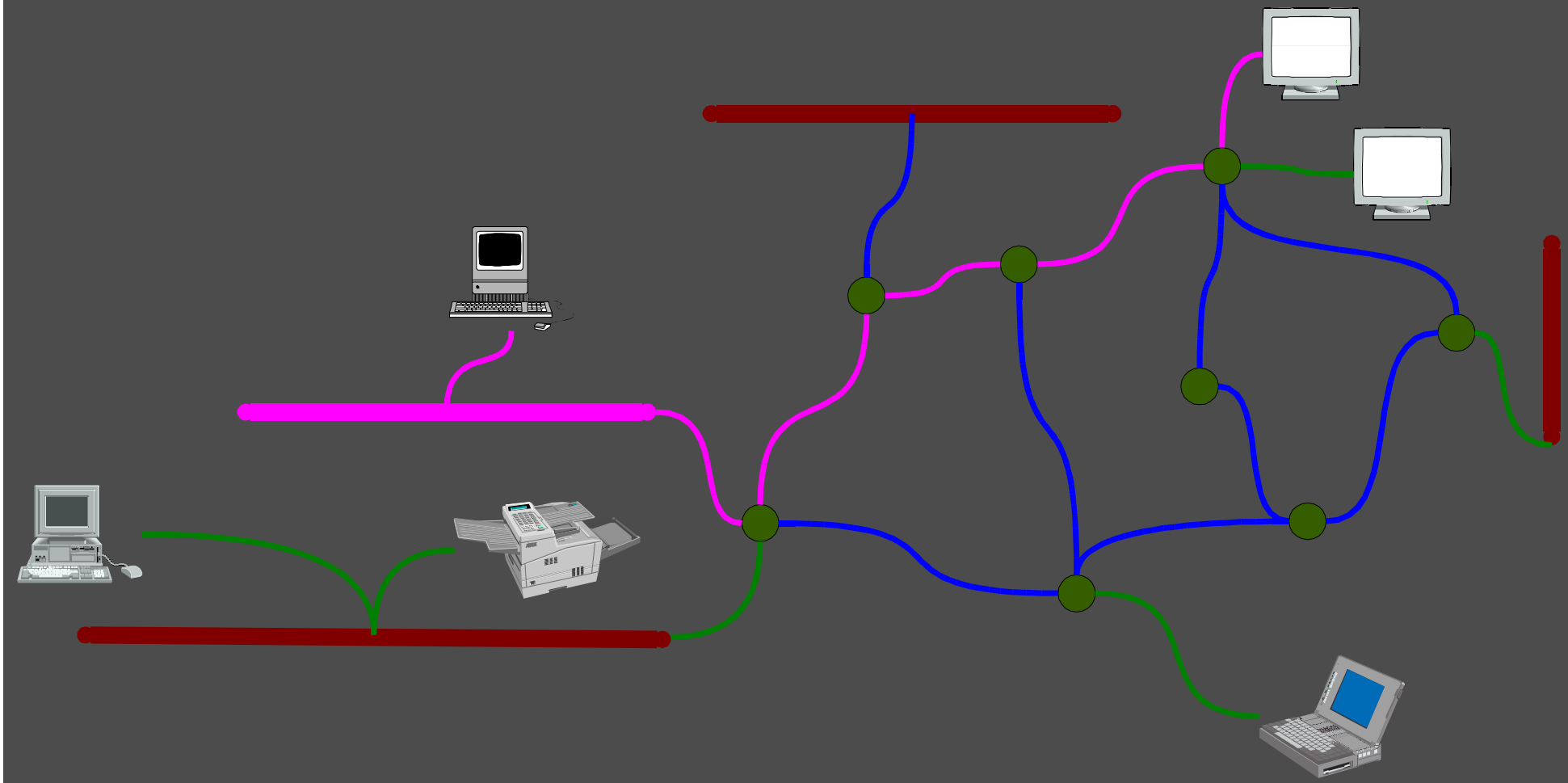
- *cast
  - Unicast
  - Broadcast
  - Multicast
  - Anycast
- Ringe
- Bäume
- Netze

Internet: A Peek Inside

- Computers and LANs
- Backbone links
- Multiple providers



Single Server and Single Client



Many Clients

- Issues
  - >50 Million Hosts
  - »Slashdot Effect«
  - Popular Broadcasting Events
- Effects
  - Server overload
  - Network overload close to the server

General Solutions

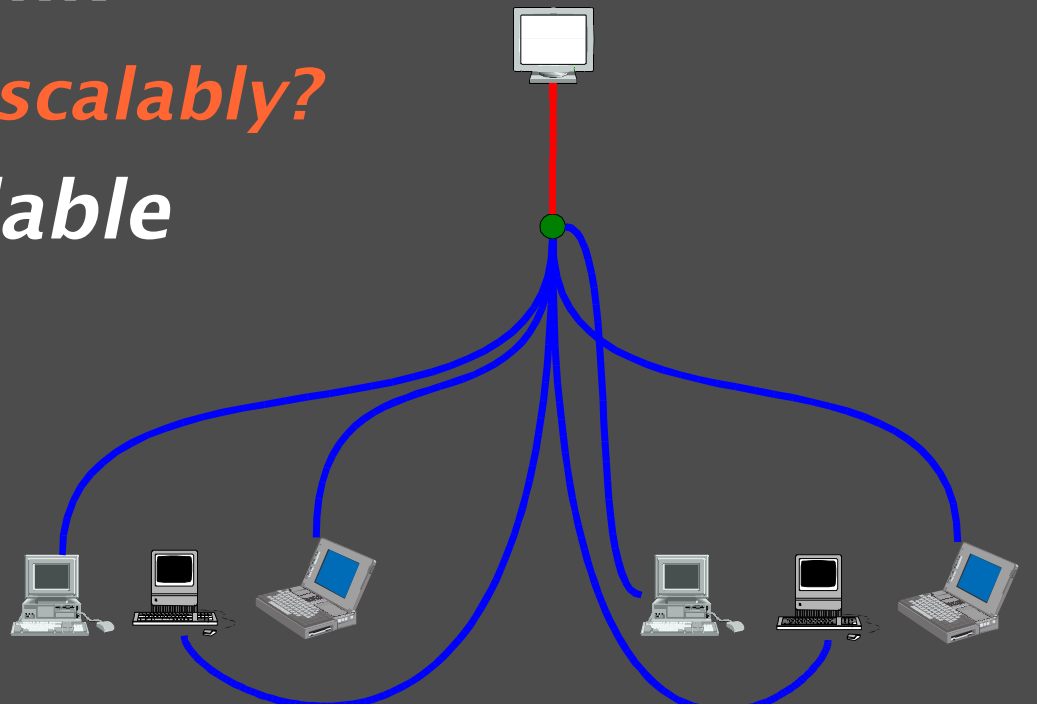
- Raw power
  - Bigger Servers
  - Faster Network
- Decentralization
  - More Servers
- Network support
  - Caches
  - Broadcast/Multicast
- New paradigms

Raw Power

- Bigger, faster computers and networks
- Split the problem
  - Distribute the requests
    - Client side: Randomly pick a server from a list
    - Server side: Virtual server, distribute requests to real servers
- Costly
- Does not scale well
- Servers and network bandwidth need to grow linearly

Multicast Problems

- Packet loss and retransmission
  - »Sender implosion«
  - Guaranteeing delivery
- Fair rate at each link
  - How to determine scalably?
- Make routing scalable

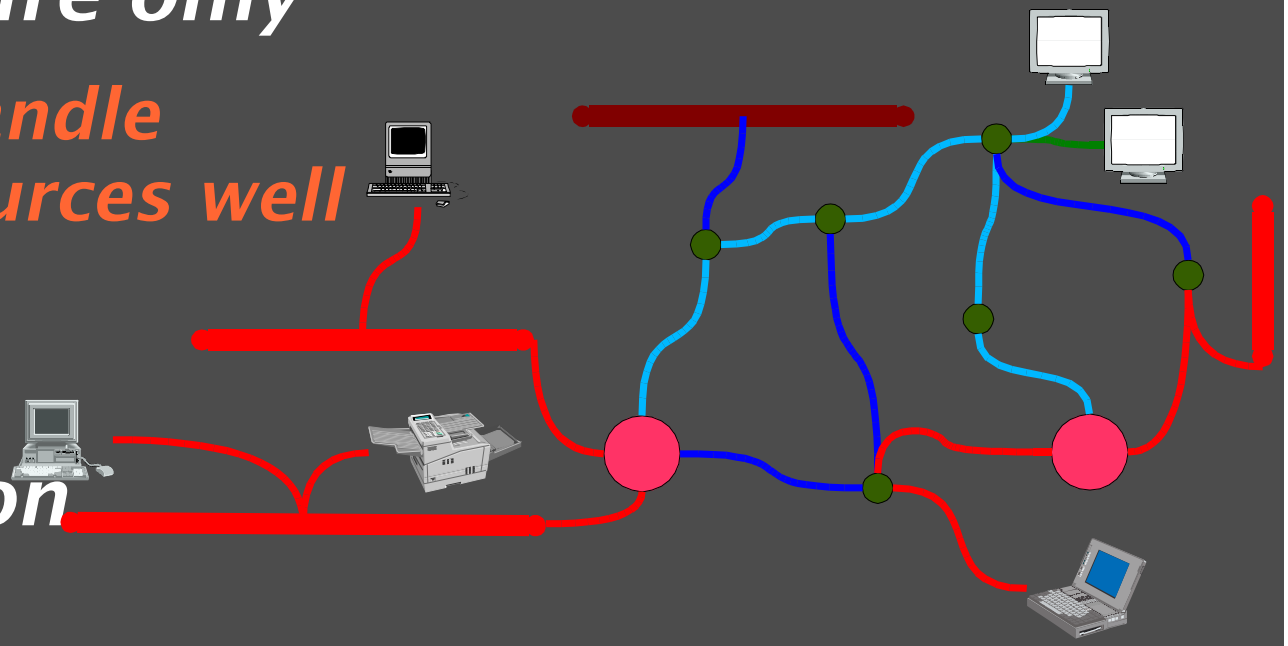


Distribute Servers

- Have servers all over the world
- Network bandwidth no longer a bottleneck
- Better latency
- Issues
  - Machines still need to grow linearly
  - Synchronization
  - Management nightmare
  - How do clients find the closest server?

Caching

- Well-known and widely used for WWW
- Static content only
- User tracking hard (e.g., shopping basket)
- Tree structure only
  - Does not handle multiple sources well
- Needs manual configuration

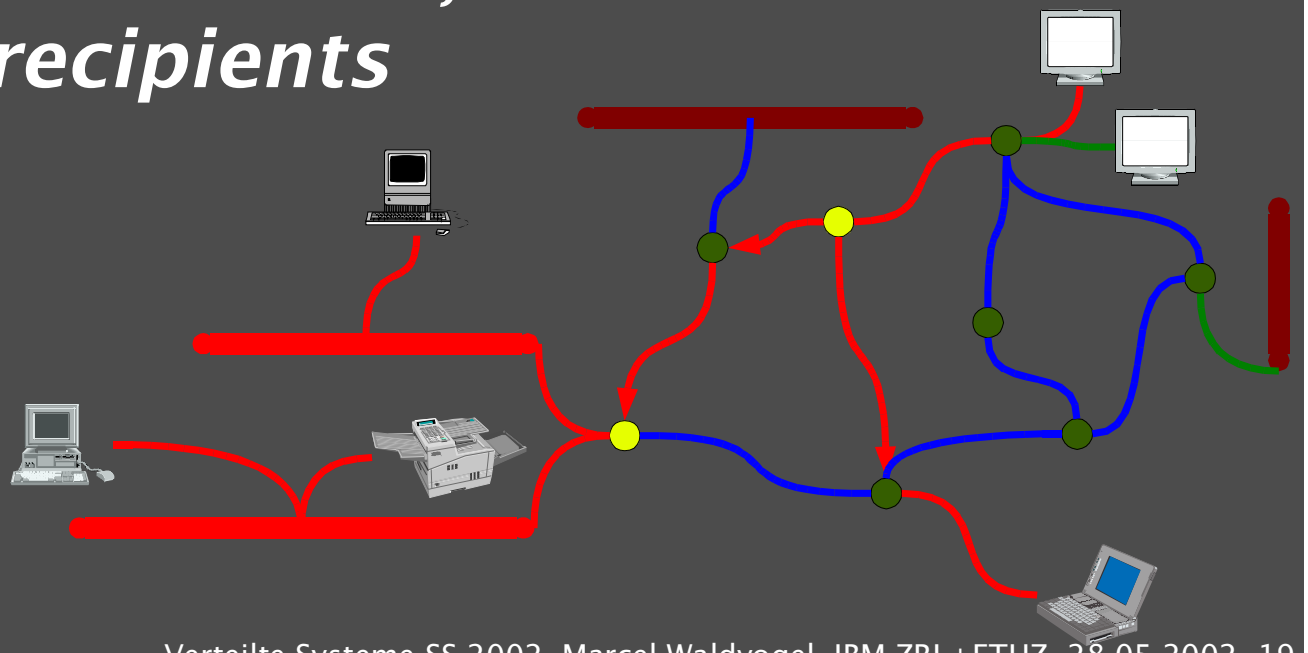


Broadcast

- Well-known from TV and Radio
- Frequencies (=bandwidth) used in entire reception area
  - Independent of interested receivers
- Does not scale to global reception of many stations
- Broadcast: »dumb« air waves
- Network: »intelligent« routers
  - Improvements worth the cost?

Introducing: Multicast

- *Unicast: Every router sends data out on a single link to get it closer to the single destination*
- *Multicast: Data goes out on more than one link, if multiple recipients exist*



Multicast

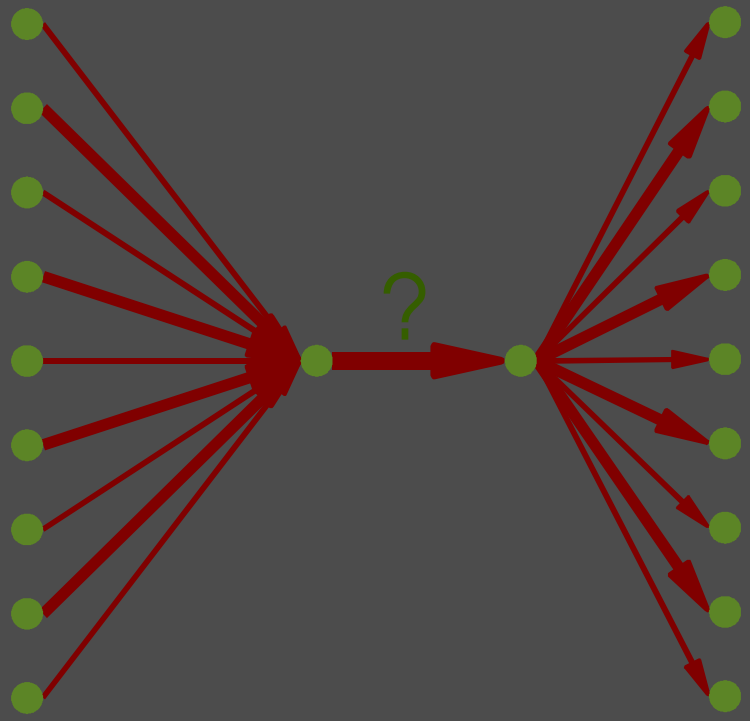
- Perfect solution?
  - Defined for Internet since 1991
  - Extremely limited availability
- Routing protocol expensive
  - Routing traffic
  - Router memory
- ISPs are afraid
  - Data traffic
  - Reliability
  - Charging

Internet Policy

- Fair bandwidth sharing
  - No enforcement
- Routers still relatively dumb
  - Cost/performance
  - Only tries to forward packets
  - No retransmits
  - No information processing
  - Overload notified as packet loss

Congestion Control

- *Why fairness?*
- *How to achieve?*
- *How to find out about fair share?*



Congestion Control: TCP/IP

- Router »dumb«
- Router provides random packet losses on overutilized links
  - Receiver reports loss to sender
  - Sender reduces transmission rate at every loss
- Large flows see more losses
- To use available bandwidth, senders increase data rate slowly

Congestion Control: Multicast Issues

- Worst case congestion as basis
  - As with unicast
  - Policy cut-off
- Packet loss feedback not scalable
  - Implosion
- Drop-to-zero problem
  - Loss rate, not loss events

SRM/CC

- Only small number (~1) of receivers provide feedback
- Dynamic election process
  - Worst candidate
  - Piggybacked on retransmission request
    - Aggregated
  - Probabilistic
    - Most losses
  - Low-pass filter

Anycast

- *IPv6*
- *Global Internet Anycast*
- *Adresse eines möglicherweise replizierten Dienstes*
- *Routinginformation zur Lokalisierung*

Applikationen

- *Zeitsynchronisation (NTP)*
- *NNTP*
- *Datensynchronisation*
- *AFS*

Zeit

- Synchrone Zeit wichtig
- Lichtgeschwindigkeit endlich
  - 1GHz ~ 1ps ~ 20cm
  - 1kHz ~ 1ms ~ 200km
- Global Position System (GPS)
- Network Time Protocol (NTP)
  - Netzwerke weder deterministisch noch symmetrisch
  - Frequenz und Phase
- Ordnung

NNTP

- Network News Transport Protocol, 1986
  - Globales Diskussionsforum
  - Replikation
  - Grosse Datenmengen
- Redundantes Netz
- IHAVE/SENDME mit Message-IDs
  - Ineffizient

Massensynchronisation

- Effizienz steigern, aber wie?
- Annahmen:
  - Viele Nachrichten
  - Relativ wenige Quellen (Hunderte)

Andrew File System (AFS)

- Verteiltes Dateisystem (CMU, IBM, Open)
- Baum von (replizierten) Server
  - 2PC
- Aggressives Caching der Clienten
  - Callbacks mit Limiten
  - Nach Dateimodifikation: Client als Server

Weitere Applikationen