

Digitale Kommunikation und Internetdienste 1

Wintersemester 2004/2005 – Termin 16

Belegnummer Vorlesung: 39 30 02

 Übungen: 39 30 05

Jan E. Hennig

AG Rechnernetze und Verteilte Systeme (RVS)

Technische Fakultät

Universität Bielefeld

`jhennig@rvs.uni-bielefeld.de`

basierend auf den Arbeiten von Michael Blume, Heiko Holtkamp, Marcel Holtmann und I Made Wiryana

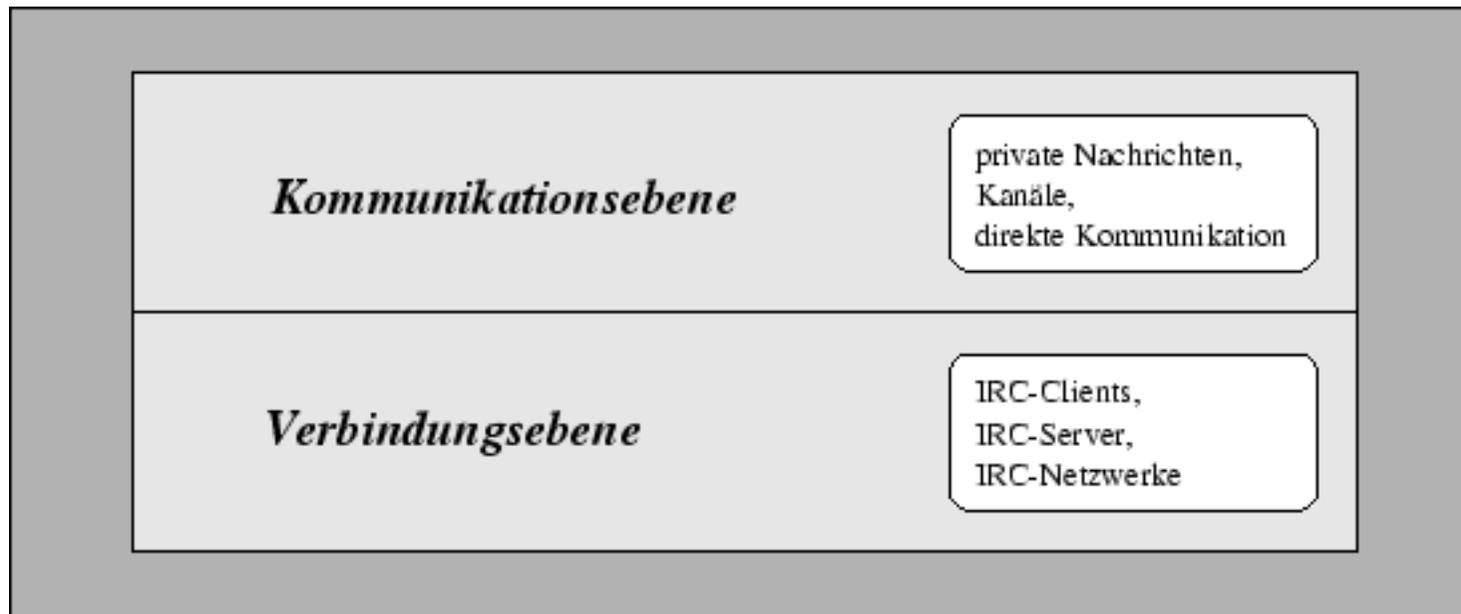
- IRC
- Quality of Service
- Film

- *Internet Relay Chat (IRC)*
- Kommunikationsdienst, der es vielen Benutzern gleichzeitig erlaubt, sich über das Internet miteinander zu unterhalten
- Benutzer sind über ein virtuelles Netzwerk miteinander verbunden
- Unterhaltung kommt dadurch zustande, daß sich die Gesprächsteilnehmer abwechselnd getippte Textzeilen schicken
- diese werden über das virtuelle Netzwerk zum Empfänger transportiert

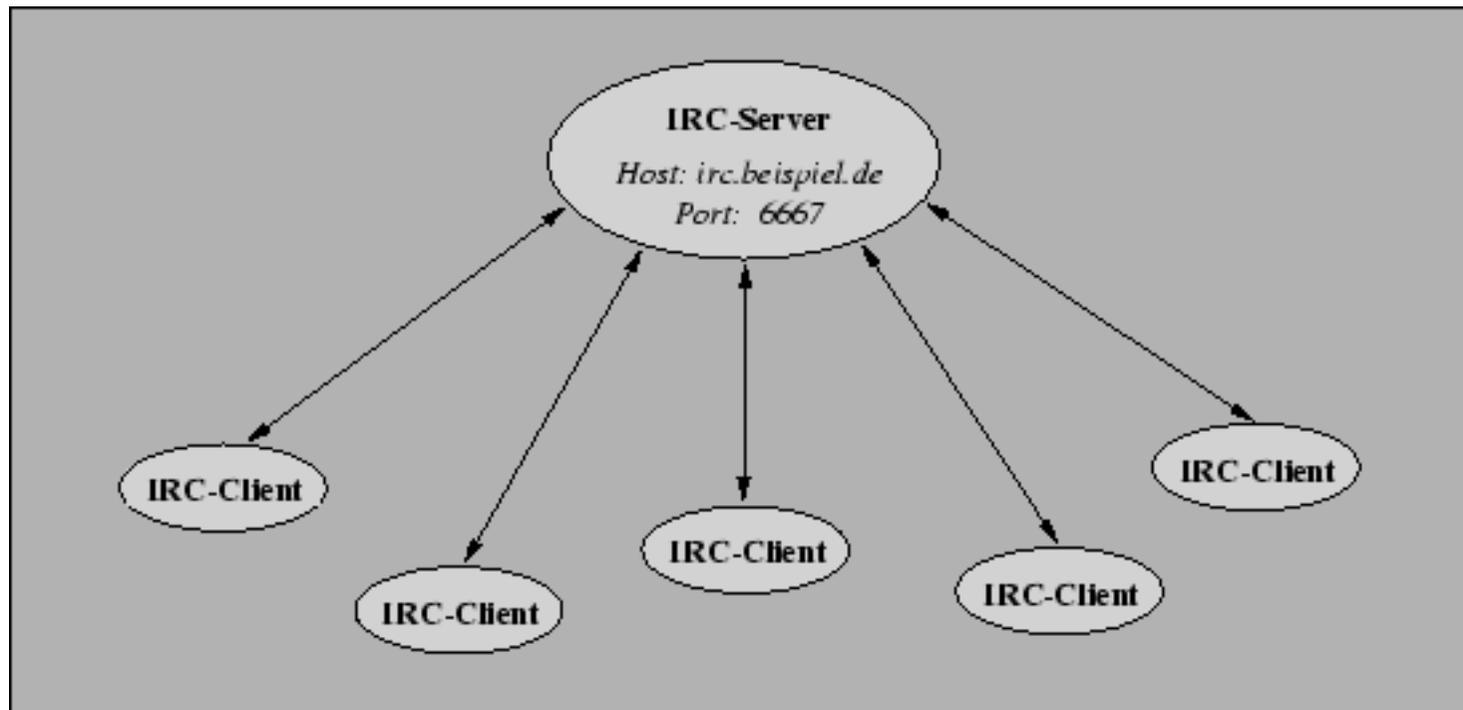
- Jarkko Oikarinen, Student der Universität von Oulu (Finnland)
- er entwickelte 1988 einen interaktiven Kommunikationsdienst für sein Bulletin Board System (BBS)
- Besonderheit: mehrere Benutzer konnten gleichzeitig daran teilnehmen
- Erweiterung für den Nachrichtenaustausch auch zwischen den Servern führte zum IRC

- mehrere populäre Netze entstanden
- während des Golfkrieges im Jahr 1991 waren die aktuellsten Berichte über den Verlauf des Krieges über den IRC erhältlich
- ähnliches Szenario (Live-Berichterstattung) im September 1993 beim Putsch gegen Boris Jelzin in Moskau

- IRC baut auf TCP auf
- IRC selbst wird in zwei Ebenen unterteilt:
- *Verbindungsebene*: Verknüpfung von verschiedenen Softwarekomponenten
- *Kommunikationsebene*: Unterstützung für verschiedene Kommunikationsformen

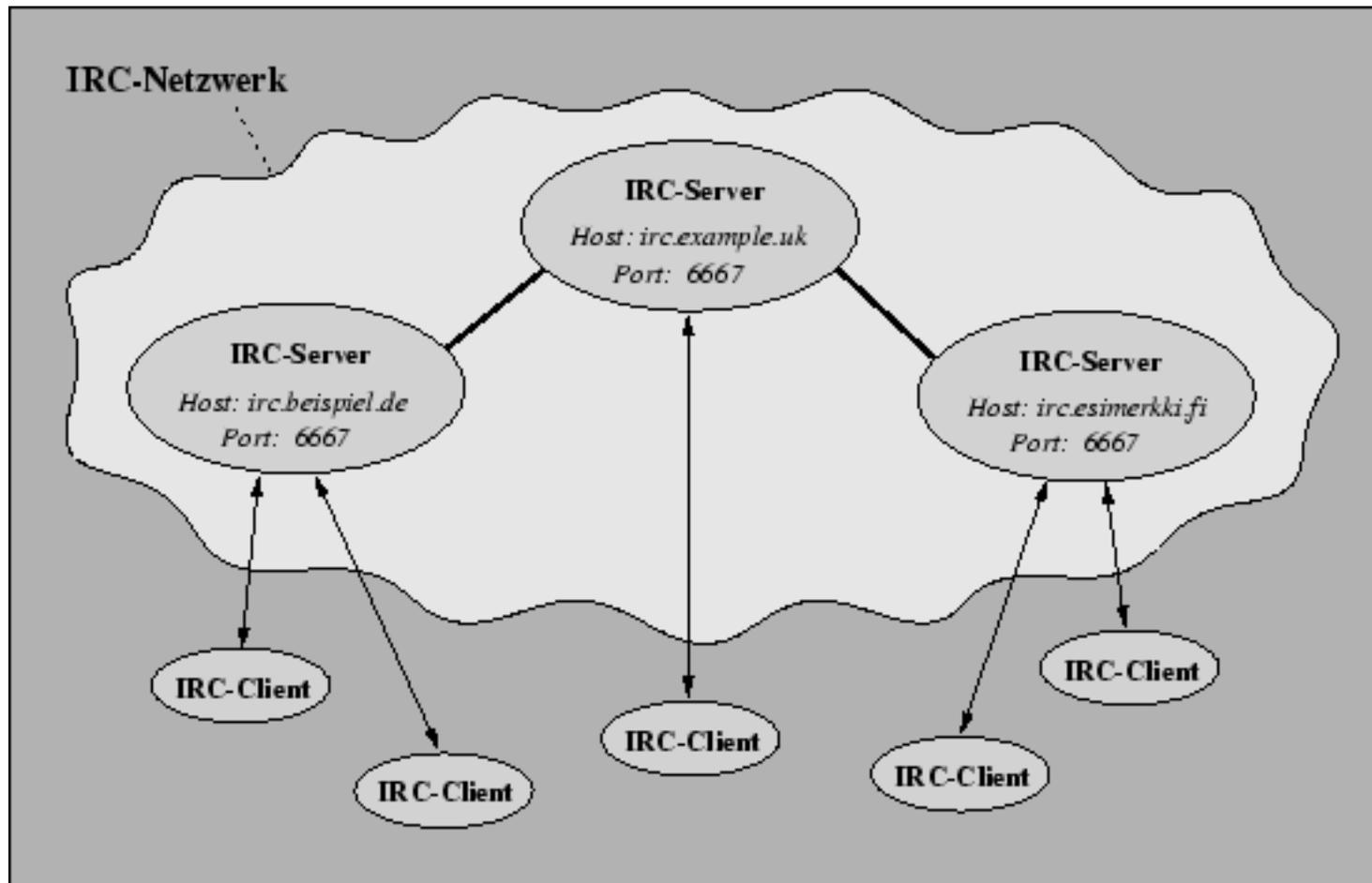


- Client-Server-Prinzip:
- mehrere Clients verbinden sich mit einem Server



- ein IRC-Server bildet das Zentrum für die Kommunikation
- wartet i.d.R. an Port 6667/tcp auf Verbindungen
- Benutzer stellen jeweils mit einem IRC-Client-Programm eine Verbindung zum Server her
- Server ermöglicht daraufhin durch die Vermittlung von Nachrichten die Kommunikation zwischen den Benutzern, die über ihn miteinander verbunden sind (*Relaying*)

- mehrere IRC-Server können zu einem IRC-Netzwerk zusammengeschlossen werden



- Server informieren sich gegenseitig über die Aktivitäten in ihrem IRC-Netz (z.B. über die Anmeldung eines neuen Benutzers)
- ermöglichen einen Nachrichtentransport zwischen allen angeschlossenen Benutzern
- Verteilung der Netzlast so möglich (erst bei vielen Benutzern sinnvoll)
- Nachrichten, die nur für lokale Benutzer bestimmt sind, können in einem Serverbereich verbleiben

- Pseudonyme Kommunikation als Standard:
- jeder Benutzer muß für sich beim Verbindungsaufbau zum IRC-Server einen Spitznamen (nick name) wählen
- sollte ein Benutzer bei der Anmeldung einen Spitznamen gewählt haben, der bereits vergeben ist, dann wird er vom Server aufgefordert, einen anderen Spitznamen zu wählen
- ein Server, der die Anmeldung eines Benutzers entgegennimmt, informiert anschließend die restlichen Server des IRC-Netzes über diesen neuen Benutzer

- zwei Befehle zum Nachrichtenversand: PRIVMSG und NOTICE
- Nachrichtenformate unterscheiden sich beim Transport nur durch den Namen des Befehls voneinander
- Auswertung aber unterschiedlich:
- Messages für die übliche Kommunikation zwischen zwei Gesprächspartnern
- Notices dienen dazu, etwas bekanntzugeben, das keiner Antwort bedarf

- Möglichkeiten für Zielparameter:
- einzelne IRC-Benutzer
- eine Liste von IRC-Benutzern
- Kanäle
- Broadcast-Messages (nur Operatoren, für gesamtes IRC-Netz eines Servers)

- jeder Benutzer kann einen eigenen Kanal anlegen
- Kanalnamen müssen wie die Spitznamen der Benutzer innerhalb eines IRC-Netzes eindeutig gewählt werden
- der erste Benutzer, der einen neuen Kanal betritt, wird Kanal-Operator genannt
- er kann z.B. anderen Benutzern dieses Kanals die Redefreiheit entziehen oder weitere Leute zu Kanal-Operatoren ernennen

- IRC besteht heute aus mehreren Protokollen
- IRC-Protokoll (RFC 1459) beschreibt Struktur des IRC und das Format der Nachrichten
- Client To Client-Protokoll (CTCP)
- Direct Client to Client-Protokoll (DCC)
- weitere RFCs: 2810, 2811, 2812, 2813

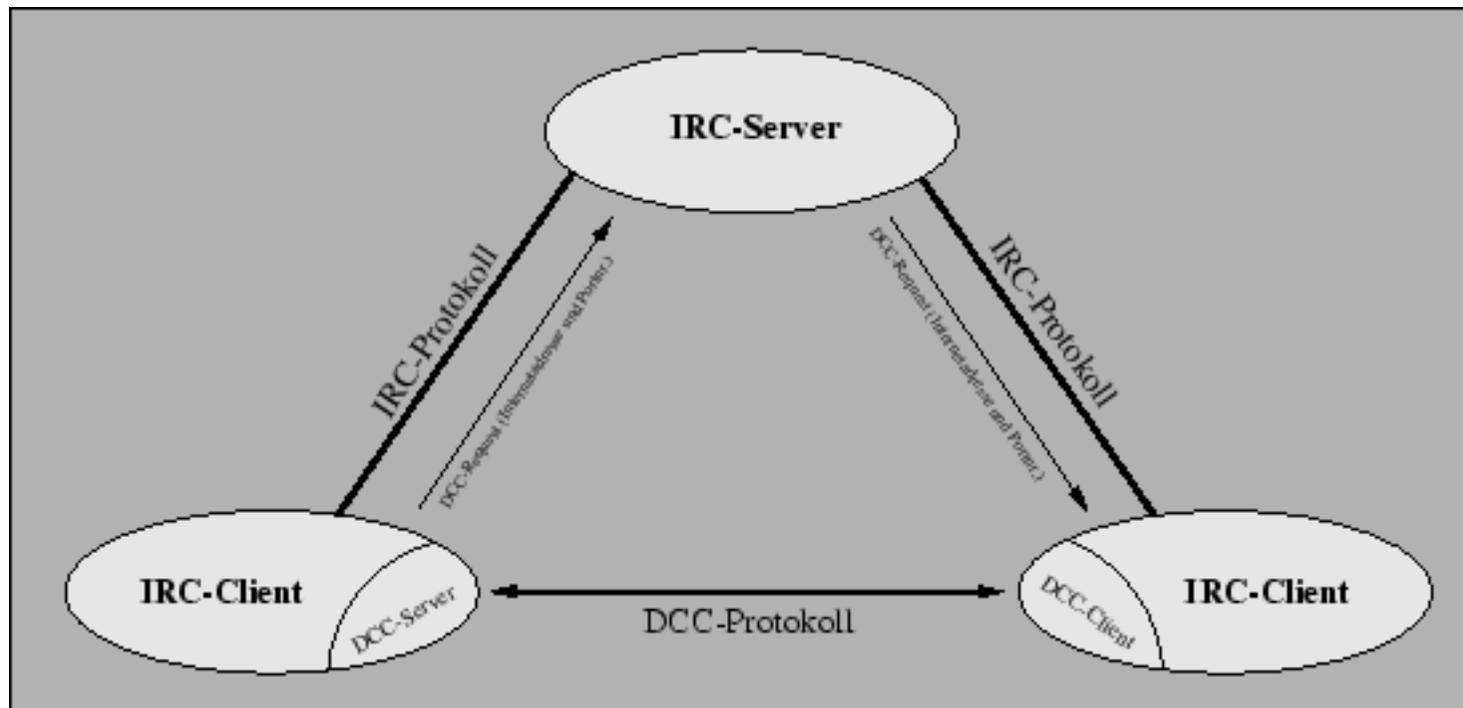
- Anmeldung mit PASS, NICK, USER
- Authentifizierung über DNS-Lookup des Hostnames, Benutzername und (optional) Kennwort
- MOTD-Befehl: Message of the day
- üblicherweise hinterlegt in einer Textdatei

- PRIVMSG oder NOTICE
- Parameter: ein oder mehrere durch Kommata getrennte Empfänger
- Anschließend der zu sendende Text
- als Empfänger können Nicknames oder Kanalnamen angegeben werden
- Beispiel: PRIVMSG erwin2001 Hallo Erwin, wie geht es?

- ERROR <Nachricht>
- z.B. bei fehlgeschlagener Anmeldung
- QUIT <Nachricht>
- zum normalen Abmelden

- Erweiterung des IRC-Befehlssatzes
- VERSION, PING, CLIENTINFO, ACTION, DCC, FINGER, TIME
- ACTION: zum Beschreiben einer Handlung (z.B. <Nickname> tut dies und jenes)
- DCC: Aufbau einer Direct Client-to-Client-Verbindung
- FINGER: Informationen zum Benutzer, z.B. Name des Rechners oder Dauer, die der Benutzer im IRC untätig war (idle time)
- TIME: Abfragen der Systemzeit des Gesprächspartners (z.B. für internationale Gespräche)

- Sammlung von direkten Übertragungsprotokollen (Umgehen des IRC-Servers nach dessen Vermittlung)



- DCC-Chat: direkte Verbindung analog zu PRIVMSG
- DCC-Send: Senden einer Datei
- DCC-Mime: Ersatz für DCC-Send mit Angabe des MIME-Typs

Nicht oder nur in geringem Maße unterstützt werden:

- Verschlüsselung einer Unterhaltung
- Authentifizierung der Gesprächspartner
- Überprüfung der Integrität empfangener Nachrichten

- IRC sehr weit verbreitet, stößt aber an Grenzen
- Unix talk: Direktverbindung zwischen zwei Benutzern
- ICQ („I seek you“): Benutzer über Seriennummer wieder auffindbar, erkennbar, wenn ein Bekannter online ist
- Jabber: verschlüsselte Verbindungen zum Server möglich (optional)
- ...

- Quality of Service (QoS)
- Qualität eines Dienstes?
- Anforderungen für einen Dienst?

- DIN 55350 Teil 11: Gesamtheit von Merkmalen (und Merkmalswerten) einer Einheit bezüglich ihrer Eignung, festgelegte und vorausgesetzte Erfordernisse zu erfüllen
- ANSI / ASQC A3-1978: Quality is the totality of features and characteristics of a product or a service that bear on its ability to satisfy given needs
- es gilt also Erfordernisse (Bedürfnisse) zu erfüllen

- Effizienz: Zeitbedarf, Ressourcenbedarf
- Funktionalität: Vollständigkeit, Korrektheit, Sicherheit, Kompatibilität
- Wartbarkeit: Hard- und Softwareunabhängigkeit, Wiederverwendbarkeit
- Zuverlässigkeit: Mängellosigkeit, Fehlertoleranz, Verfügbarkeit
- Benutzbarkeit: Verständlichkeit, Erlernbarkeit, Mitteilsamkeit

- Möglichkeiten grob bestimmbar anhand der Eckdaten, z.B.:
- Bandbreite
- Latenz
- Jitter (Schwankung der Latenz)
- Berechnung wird komplex, sobald mehrere Netzwerksegmente zu überbrücken sind

- Vorhersagen für den realen Betrieb schwer:
- gemischte Anwendungen in einem Netzwerk
- z.B.: zufällig versuchen alle Anwendungen gleichzeitig zu senden
→ Auslastungsspitzen
- je nach Anwendung oder Dienst sind gewisse Zusicherungen nötig
- Bedarfsanalyse für Netzwerk sollte daher auch Worst-Case-Szenario berücksichtigen

- verschiedene Dienste besitzen verschiedene Anforderungen an die Qualität
- meist werden Mindestanforderungen angegeben
- z.B. MP3-Audiostream: konstante Datenrate mit z.B. 128KBit/s
- z.B. IRC: Latenz und Jitter gering, z.B. maximal 50ms Verzögerung der RTT

- Bandbreite vorgegeben
- verschiedene Anwendungen mit unterschiedlichem Bedarf
- z.B. FTP: viele Daten aber unkritisch bei Zeit
- z.B. Chat: wenig Daten aber kritisch bei Latenz
- z.B. Webseiten: viele Auslastungsspitzen neben langen Leerlaufzeiten

- Einteilen des Netzwerkverkehrs in verschiedene Klassen
- den Klassen werden verschiedene Garantien gegeben
- die Klassen werden zudem priorisiert
- Anwendbarkeit getrennt für jeweils eine Richtung (Upstream/Downstream)

- DSL-Anschluß, 128Kbit/s Upstream
- Telefonieren mit Voice-Over-IP benötigt etwa 90Kbit/s pro Richtung, Telefonieren aber selten
- Chat ist zeitkritisch, aber überträgt kaum Daten
- nebenher soll ein FTP-Transfer die restliche Telefonie-Kapazität nutzen können
- außerdem: ACK-Pakete möglichst vorziehen für schnelle Bestätigung

- mögliche Vorgaben:
- Priorität 1 mit 4 Kbit/s reserviert für Chat und ACKs
- Priorität 2 mit 90 Kbit/s reserviert für Voice-Over-IP
- Priorität 3 für restliche Anwendungen

- solange nicht telefoniert wird, bleiben die 90 Kbit/s für niedriger priorisierte Klassen übrig
- somit kann FTP-Transfer schneller ablaufen
- wenn Telefonat beginnt, werden die 90 Kbit/s in Anspruch genommen
- FTP-Transfer wird nun auf restliche 37 KBit/s beschränkt

Beispielhafte Umsetzung unter Linux mit Hierarchical Token Bucket (HTB)

```
#!/bin/sh
```

```
INET=ppp0
```

```
LAN=br0
```

```
# "root handle" vorgeben
```

```
/sbin/tc qdisc add dev $INET root handle 1:0 htb default 12
```

```
# Klassen mit zugelassenen Datenraten erstellen
```

```
/sbin/tc class add dev $INET parent 1:0 classid 1:1 htb rate 128kbit ceil 128kbit
```

```
/sbin/tc class add dev $INET parent 1:1 classid 1:10 htb rate 4kbit ceil 4kbit prio 0
```

```
/sbin/tc class add dev $INET parent 1:1 classid 1:11 htb rate 90kbit ceil 90kbit prio 1
```

```
/sbin/tc class add dev $INET parent 1:1 classid 1:12 htb rate 37kbit ceil 127kbit prio 2
```

- woran kann ein Router erkennen, was für eine Art Verkehr vorliegt?
- Klassifikation des Verkehrs nach typischen leicht verfügbaren Merkmalen
- z.B. Port, Paketgröße (z.B. reine ACKs sehr klein), Type of Service-Feld (oft nicht korrekt verwendet)
- sicherere Klassifikation: komplettes Paket untersuchen (auf Anwendungsebene → aufwendig)
- anschließend Markieren der Pakete und Zuweisung in Klassen

Beispielhafte Umsetzung unter Linux mit iptables

```
# Klassifizieren und Markieren der Pakete

# kleine Pakete (z.B. ACKs) nach Länge
/sbin/iptables -A POSTROUTING -t mangle -o $INET -p tcp -m length --length :64 -j MARK --set-mark 10

# IRC
/sbin/iptables -A POSTROUTING -t mangle -o $INET -p tcp --dport 6667 -j MARK --set-mark 10

# H.323 Host Call
/sbin/iptables -A POSTROUTING -t mangle -o $INET -p tcp --dport 1720 -j MARK --set-mark 11
/sbin/iptables -A POSTROUTING -t mangle -o $INET -p tcp --sport 1720 -j MARK --set-mark 11

# FTP
/sbin/iptables -A POSTROUTING -t mangle -o $INET -p tcp --dport 20 -j MARK --set-mark 12
/sbin/iptables -A POSTROUTING -t mangle -o $INET -p tcp --dport 21 -j MARK --set-mark 12

# Zuordnen der markierten Pakete zu den HTB-Klassen
/sbin/tc filter add dev $INET parent 1:0 prio 0 protocol ip handle 10 fw flowid 1:10
/sbin/tc filter add dev $INET parent 1:0 prio 0 protocol ip handle 11 fw flowid 1:11
/sbin/tc filter add dev $INET parent 1:0 prio 0 protocol ip handle 12 fw flowid 1:12
```

- Frage: Reicht das?
- Wirkung ist beschränkt auf Weiterleitung an einem Router
- Konfigurationsmöglichkeit muß auf dem Router gegeben sein
- jedoch: im Internet nicht jeder Router unter eigener Kontrolle

- praktikabel in zwei Fällen:
- QoS-Lösung wird nur im eigenen Netz (unter Kontrolle) benötigt
- man selbst kontrolliert den langsamsten Abschnitt im Pfad
- im Zweifelsfall: Anmieten/Verlegen separater/weiterer Leitungskapazität
- denn: die Bandbreiteneinschränkungen sind nicht durch andere Verteilung des Verkehrs aufhebbar

- Film:
- Erklärung: „Wie funktioniert das Internet“
- Grobe Abstraktion von Provider, DNS, Routing
- Stand des Films: Anfang 1999

Ende Termin 16. Danke für die Aufmerksamkeit.