TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik | Institut Systemarchitektur | Professur Rechnernetze

WS 2016/2017
LV Rechnernetzpraxis

8. Nachrichtenvermittlung

Dr. rer.nat. D. Gütter

Mail: Dietbert.Guetter@tu-dresden.de
WWW: http://www.guetter-web.de/education/rnp.htm

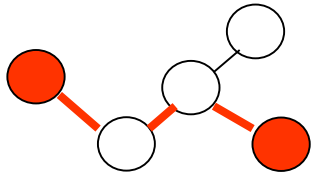
30.11.2016

Vermittlungsschicht

Gewährleistung der
Kommunikation zwischen beliebigen Stationen im Netzwerk

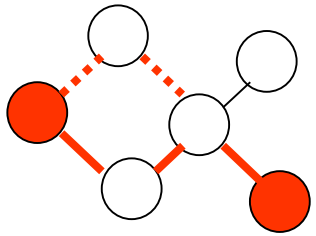
Netzwerke mit **eindeutige**n physischen Übertragungs**wege**n

- Adressierung erforderlich
- einfache Vermittlung (Weiterleitungstabellen)



Netzwerke
mit **alternative**n Übermittlungs**wege**n

- Adressierung erforderlich
- komplexe Vermittlung
- Wegebewertung (Metrik)
  und Wegeauswahl erforderlich (Routing)



Adressierung

Adressen ohne Bezug auf Übertragungsweg
(Skalierbarkeit problematisch!)

z.B. Bernd Mustermann
Matrikel-Nummer: 987654321

Vermittlung unmöglich ohne Zusatzinformationen,
z.B. vom Immatrikulationsamt

Adressen mit Bezug auf Übertragungsweg
(skalierbar!)

z.B. Bernd Mustermann
Tiergartenstr. 4
01219 Dresden

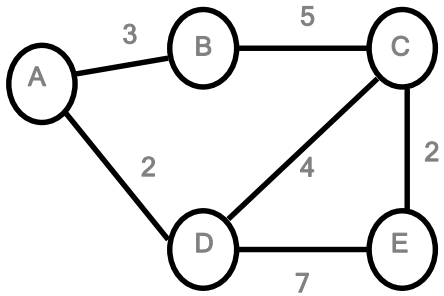
Vermittlung erfolgt schrittweise durch Auswertung der Adresse

Wegewahl - Metriken

Metriken zur Bestimmung optimaler Wege

- Anzahl der Vermittlungsschritte (Hops) von Quelle zum Ziel
- geographische Distanz, Laufzeit
- Kapazität der Leitungen, Datenrate
- Fehlerraten der Leitungen
- mittlere Auslastung der Leitungen und Router
- Kosten der Leitung

→ Gewichtete Bewertungszahl (Metrik), verallgemeinerte Entfernung



Wegewahl Auswahl des kürzesten Weges (optimaler Weg) aus der Menge der möglichen Wege

Wegewahl - Strategien

bester Weg

- Auswahl des „kürzesten" Weges aus der Menge der möglichen Wege

minimale Verzögerung, maximalerDurchsatz, …

- Ressourcenreservierungen in Vermittlungsknoten
- u.U. Aufsplitten des Datenverkehrs auf mehrere Leitungen
- Berücksichtigung der Auslastungen der Leitungen

Gesamtnetzoptimierung

schwierig, da Netze hochdynamisches Verhalten besitzen

Hierarchisches Routing

- Routing in großen Netzen → hohe Netzbelastung, träge, …
- Einteilung in Regionen mit internem Routing
- Routing zwischen Regionen
  ohne Berücksichtigung regional interner Parameter
- evtl. auch Multilevel-Hierarchie

Statisches Routing (Wegewahl)

Einmalige Wegeberechnung → statische Weiterleitungstabellen
veränderte Netzeigenschaften → Neuplanung

zentrale, globale Planung

- Voraussetzung: Kenntnis aller Netzeigenschaften
  Topologie, Netzlast, Übertragungskapazitäten, …
- Berechnung der optimalen Wege im Gesamtnetzinteresse möglich

lokale, isolierte Planung

- Stationen wählen den für sie optimalen Weg, evtl. Interessenkonflikte

verteilte Planung

- Austausch von Weginformationen zwischen Nachbarn
  danach lokale Berechnung, führt zu günstigen Wegen

Adaptives Routing

Permanente Anpassung an Topologie- und Lastveränderungen
→ dynamische Weiterleitungstabellen

zentrale, globale Planung

- zu träge für adaptive Wegewahl,
  ggf. Berechnung der Initialwerte für Tabellen

lokale, isolierte Planung

- geignet in Extremsituationen
  und als Ergänzung zu verteilter Planung

verteilte Planung

- geignet für adaptive Wegewahl
- Problem: Austausch von Weginformationen erfordert Zeit
  → evtl. nichtoptimale Wegewahl
  bei schnellen Laständerungen

Wegewahlverfahren

Fluten: lokales Routing

jede ankommende Nachricht über alle Ports (außer Eingang) weiterleiten

Problem: Überflutung → begrenzte Lebensdauer sichern
Dekrementieren eines Zählers; Terminierung bei Zählerwert 0
(Startwert: max. Anzahl der Zwischenrechner)

Nutzung z.B. in Routing-Lernphasen

selektives Fluten:	Weiterleitung nur an ausgewählte Ports
Zufallsfluten:	Weiterleitung in eine zufällig gewählte Richtung

Rückwärtslernen: (adaptives,) isoliertes Routing

Bei Ankunft einer Nachricht werden Absenderadresse und zugehöriger Port ausgewertet und ggf. in die Weiterleitungstabelle eingetragen.

Verfahren unkritisch für Netze mit eindeutigen Wegen (z.B. bei Baumtopologie)
allgemein anwendbar, wenn die Nachricht eine Bewertung des Weges enthält (z.B. Zähler mit Anzahl der Hops)

Hot Potato: adaptives, isoliertes Routing

bei alternativen Wegen sofort absenden über Port mit kürzester Warteschlange

Wegewahl

Link State Algorithmen

Router verfügen über vollständige Kenntnis der Netzeigenschaften
Routingalgorithmen: lokal und global

- Algorithmus nach Dijkstra

Distanzvektor Algorithmen

- Algorithmus nach Bellmann-Ford

Distanz- und Weiterleitungstabellen

Distanztabellen

enthalten für jedes Ziel Distanzwerte (Kosten)
für Wege über alle Nachbarrouter

Weiterleitungstabellen

ergeben sich durch die Minima
in den Zeilen der Distanztabellen

Ziel	Router R1	Router R2	Router R3	…
x	3	5	8	…
y	7	1	2	…
z	9	7	24	…
…	…			…

Ziel	Router	Kosten
x	R1	3
y	R2	1
z	R2	7
…	…	

Distanztabellen haben den Vorteil,
daß bei Streckenausfällen Ersatzwege ermittelt werden können.

Problem:
Es kann nicht erkannt werden, ob die Ersatzstrecke über den eigenen Knoten führt

Distanzvektor-Routing

Annahme: Jeder Router kann mit benachbarten Routern kommunizieren und die Distanz der Übertragungstrecke bewerten

(1) Aufbau einer Routingtabelle mit Einträgen zu den Nachbarn

Ziel	Nachbarrouter	Distanz	Ausgabeinterface
..	…	…	…
…	…	…	…

(2) regelmäßiger Austausch der Routingtabellen zwischen den Nachbarn

Berechnung Zielabstand über Nachbar (Abstand zum Nachbarn addieren)

Tabellenabgleich,
wenn neues Ziel in Nachbarroutingtabelle → Eintrag in eigene Tabelle
bei bekanntem Ziel prüfen, welcher Nachbar der günstigste ist

schneller Tabellenaufbau (soviel Austauschzyklen wie max. Hop-Tiefe)
keine globale Sicht der Router auf die Netztopologie
→ **problematische Konvergenz bei Leitungsausfällen**

Algorithmus nach Bellmann-Ford

Gegeben sei ein Netzgraph mit n Knoten k_i und m gewichteten Kanten Gewicht(k_k,k_l)
Für jeden Knoten s = k_i wird eine Liste der optimalen Wege ermittelt

- Vorgänger(k_j) Vorgänger auf dem optimalen Weg
- Distanz(k_j) Distanz von k_j zu s auf dem optimalen Weg (negative Werte sind zulässig)

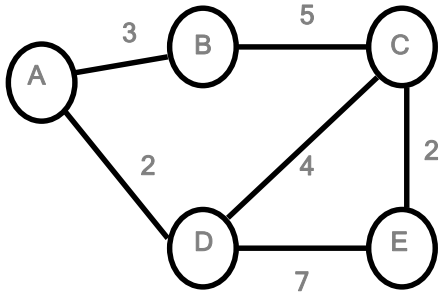
Algorithmus zur Berechnung der Liste für 1 Knoten: O(n*m)

für j=i { Distanz(k_i)=0;} sonst { Vorgänger(k_j):= kein; Distanz(k_j):= ∞ };

wiederhole (n-1) mal
für jede Kante
{ Wegeinformationsaustausch zwischen benachbarten Knoten u und v;
wenn *Distanz(u) + Gewicht(u,v) < Distanz(v)*
{Distanz(v) := Distanz(u) + Gewicht(u,v); Vorgänger(v) := u }
}
Kontrolle für jede Kante
wenn *Distanz(u) + Gewicht(u,v) < Distanz(v)*
{STOP mit Fehlerausgabe "Kreis negativer Länge gefunden" }

Distanzvektor-Routing

Beispiel



Max 3 Hops

→3 mal Tabellenaustausch
Konsistenter Endzustand

Routingtabellen

Router A

Ziel	nächster Router, Kosten
-	-,0
B	B,3
C	D,6
D	D,2
E	D,9

Router B

Ziel	nächster Router, Kosten
A	A,3
-	-,0
C	C,5
D	A,5
E	C,7

Router C

Ziel	nächster Router, Kosten
A	D,6
B	B,5
-	-,0
D	D,4
E	E,2

Router D

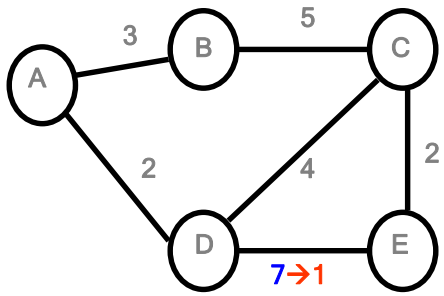
Ziel	nächster Router, Kosten
A	A,2
B	A,5
C	C,4
-	-,0
E	E,6

Router E

Ziel	nächster Router, Kosten
A	D,9
B	C,7
C	C,2
D	D,6
-	-,0

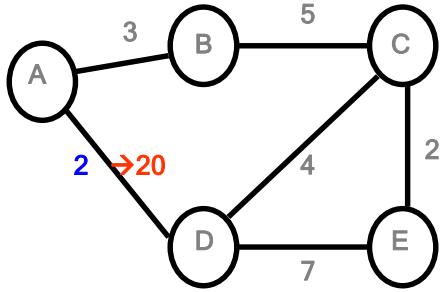
Bewertung Distanzvektor-Routing

Algorithmus einfach und effizient,
aber nach Veränderungen problematische Konvergenz zu neuem konsistentem Zustand



Gute Nachricht

→Schnelle Konvergenz



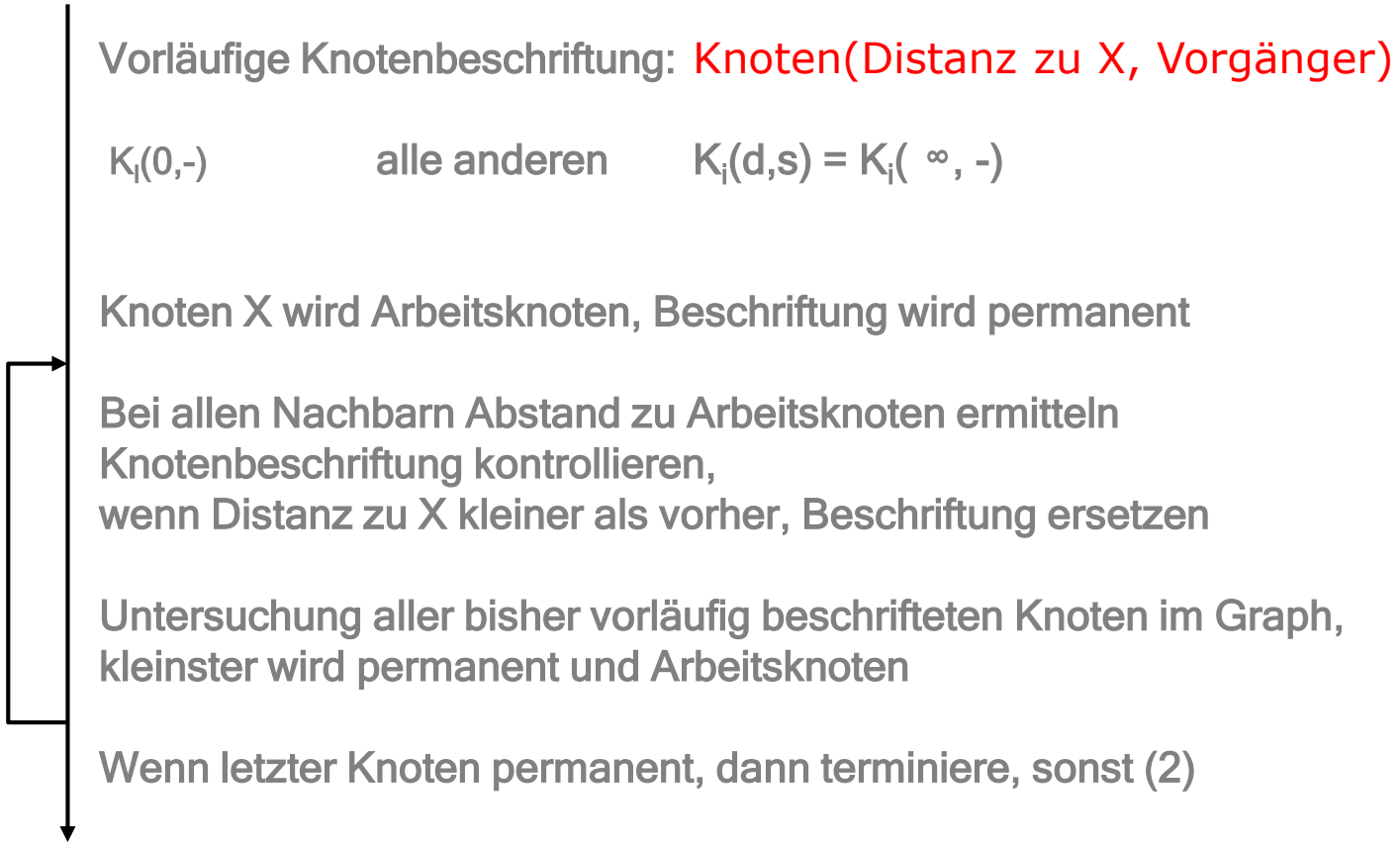
Schlechte Nachricht

→ count-to-infinity
sehr schlechte Konvergenz

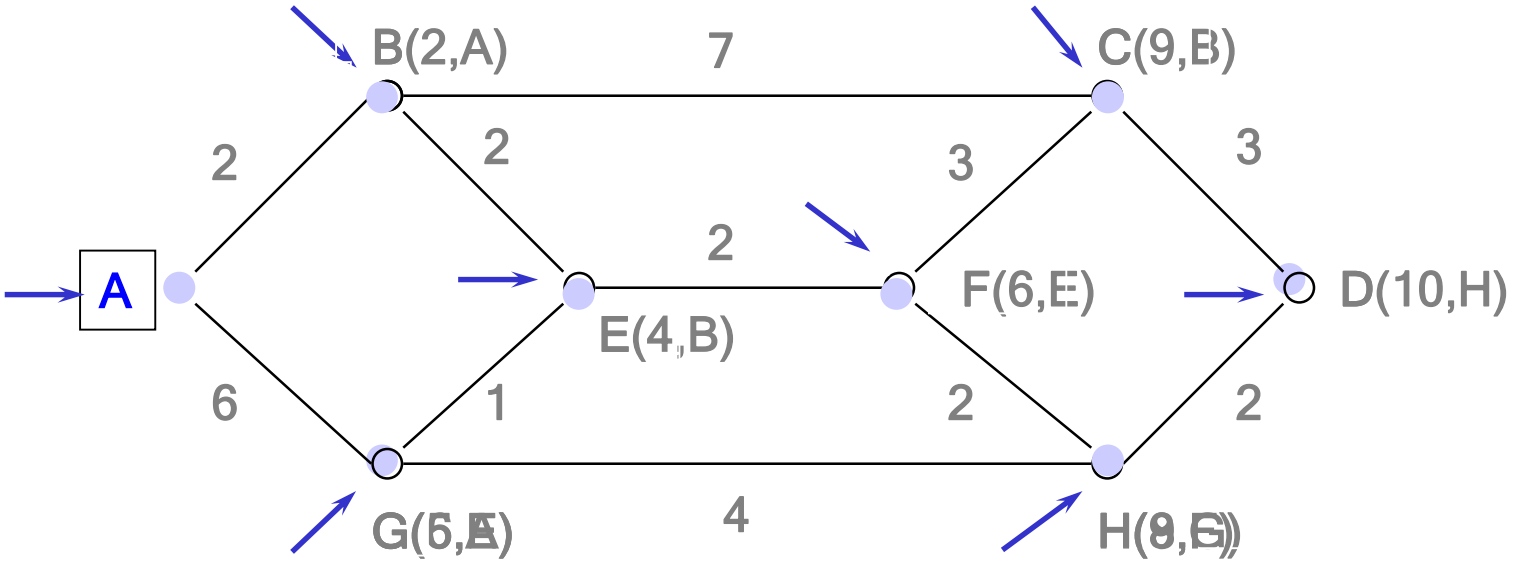
Lösung:
Unerwünschte Wege zeitweilig blockieren (unendliche Distanz)
→schnellere Konvergenz

Shortest Path Routing (nach Dijkstra, 1959)

Gesucht:
Liste kürzester Wege vom Quellknoten X zu den anderen Knoten K_l



Dijkstra-Algorithmus



permanente Knoten
Arbeitsknoten

Graph zeigt optimale Wege zu Knoten A

z.B.
von Knoten H über Knoten F mit Distanz 8
dann über E .. B .. bis zu A

Bewertung Dijkstra-Algorithmus

schrittweiser Aufbau einer globalen Sicht auf die Netzwerkstrecken

schnelle und hochwertige Wegewahlentscheidungen
Konvergenz nach **O(n*log[n]+m)** Schritten (besser als Bellmann-Ford)

Ergebnisse jedoch u.U. nicht konsistent, wenn Metrik zeitveränderlich

z.B.

- bei Topologieänderungen
  (relativ selten, neue Konvergenz nach Routerinformationsaustausch
  deshalb in der Praxis kein großes Problem)

- bei schnellen Laständerungen
  (Berechnung beruht u.U. auf veralteten Streckenbewertungen,
  damit nicht Ergebnisse optimal,
  in Extremfällen falsche Wegewahl,
  sogar Regelschwingungen (Oszillationen) möglich)

Überlastüberwachung

- Choke-Pakete (Warnungen):

  Zwischenrechner adaptiert Lastmessung für jede Ausgangsleitung:

  $Last_{neu} = a * Last_{alt} + (1 - a) * Last_{aktuell}$

  a = 0: sofortige Aktualisierung
  a > 0: langsamere Anpassung

  Senden Choke-Paket an Nachbar bei Schwellwertüberschreitung
  → dieser muß Übertragung für bestimmte Zeit einstellen

- Begrenzung der Datenrate (Traffic Shaping)
- Bandbreitenreservierung (z.B. RSVP - Resource Reservation Protocol)
- Verkehrsklassen unterschiedlicher Priorität (z.B. Differentiated Services)

Paketvermittlung

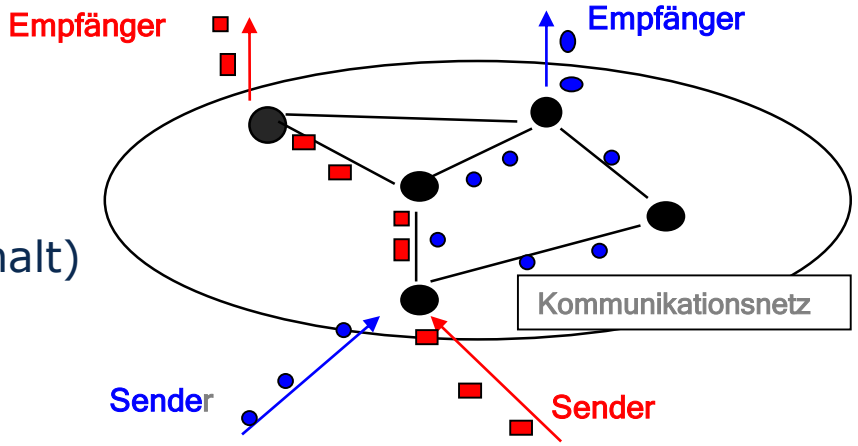
1 Phase

Übertragungsphase

- Leitungen fest geschaltet
- Paketübertragungszyklus
  (Header mit Adresse, …+ Inhalt)
- Multiplex von Info-Strömen

- hoher Overhead
- Wegewahl für jedes Paket
  u.U. alternative Wege
- schwankende Qualität
  (Ü-Rate, Jitter, …)
- Robustheit
  gegen Teilstreckenausfälle

- Lastanpassung möglich
- meist gute Ressourcenauslastung



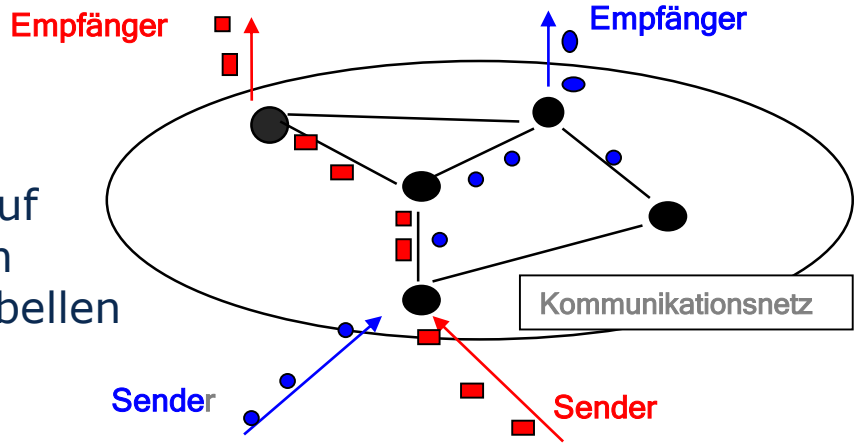
Beispiel:
IP (Internet-Netzwerkschicht)

Virtuelle Verbindungen

Phasen

1. Verbindungsaufbau

- Auswahl/Zuordnung von Nutzungskontingenten auf fest geschalteten Teilstrecken
- Aufbau von Weiterleitungstabellen
- Ressourcenreservierungen



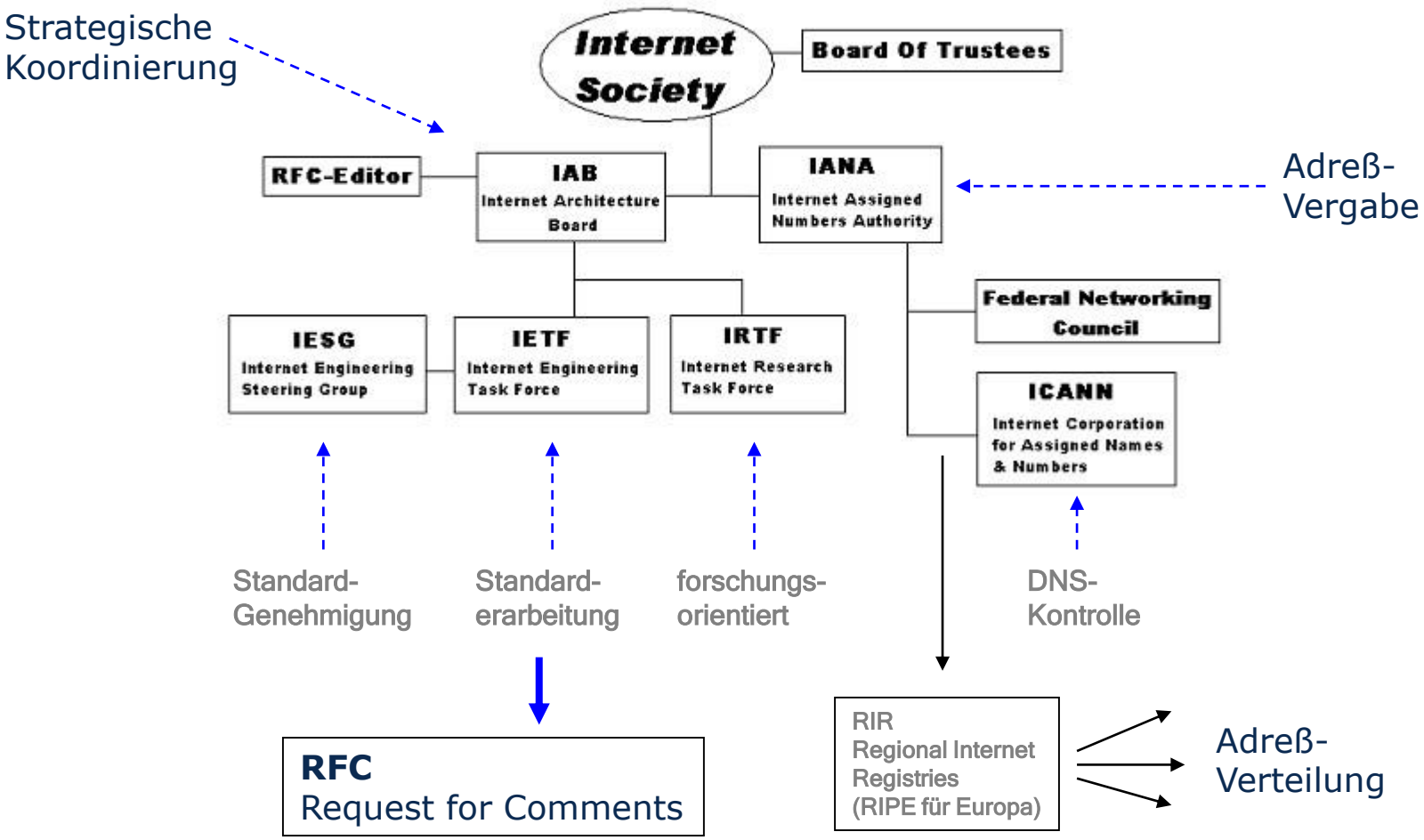
2. Übertragung

- immer gleiche Wege
- Übertragungszyklus von Paketen/Zellen
- Overhead (Verbindungskennungen, …)
- garantierte Qualität (Ü-Rate, Jitter, …)
- Verbindungsabbruch bei Teilstreckenausfall
- meist gute Ressourcenauslastung

Beispiele:
- X.25
- Frame Relay
- ATM

3. Verbindungsabbau

Internet - Organisation



Internet - Entwicklung

1970	ARPANET (ca. 100 Rechnernetze), 56 kbit/s Leitungen
1973	TCP/IP, Internetz-Architektur
1974-75	Verbesserte TCP/IP-Versionen,PRNET (Packet Radio), 100 kbit/s, SATNET (Satellite Net), 64 kbit/s,(USA,Deutchland,Italien etc.)
1979	ARPA ICCB (Internet Configuration Control Board)
1980	INTERNET-kompatible Lokale Netze TCP/IP ist Anteil von Berkeley UNIX v4.2 geworden
1984	Abspaltung des MILNET Spezielle Förderung des CSNET (Comp. Science Net)
1985	IAB (Internet Activity Board) mit Task Force als Ersatz für ICCB, neue Netze in Europa etc.
1989	IAB aufgeteilt in IETF (Internet Engineering Task Force) und IRTF (Internet Research Task Force), beginnende kommerzielle Nutzung des INTERNET
1990	NSFNET (National Science Foundation Net) mit bis zu 1,5 Mbit/s als Ersatz für ARPANET
1992	WWW, Internet Society (starke Industriebeteiligung)
1995	Neue Protokolle (IP next generation etc.)
1996	Weitere Internet-Provider, zunehmende kommerzielle Nutzung
1999	Zunehmende Verbreitung von Electronic Commerce und Multimedia
2000	Weitere deutliche Leistungserhöhung; Gigabit-Wissenschaftsnetz
2001	Zunehmender Einsatz optischer Netze, WDM (Wave Division Multiplex)
2002	Trend zu optischem Switching

RFC (Request for Comments)

Standardisierungsstatus

- Offene Diskussion in IETF-Arbeitsgruppen
  → Veröffentlichung als „Proposed Standard"
- Analyse/Test abgeschlossen, Modifikation noch möglich
  → „Draft Standard"
- Standard zur Nutzung freigegeben
  → „Full Standard"

Protokollstatus

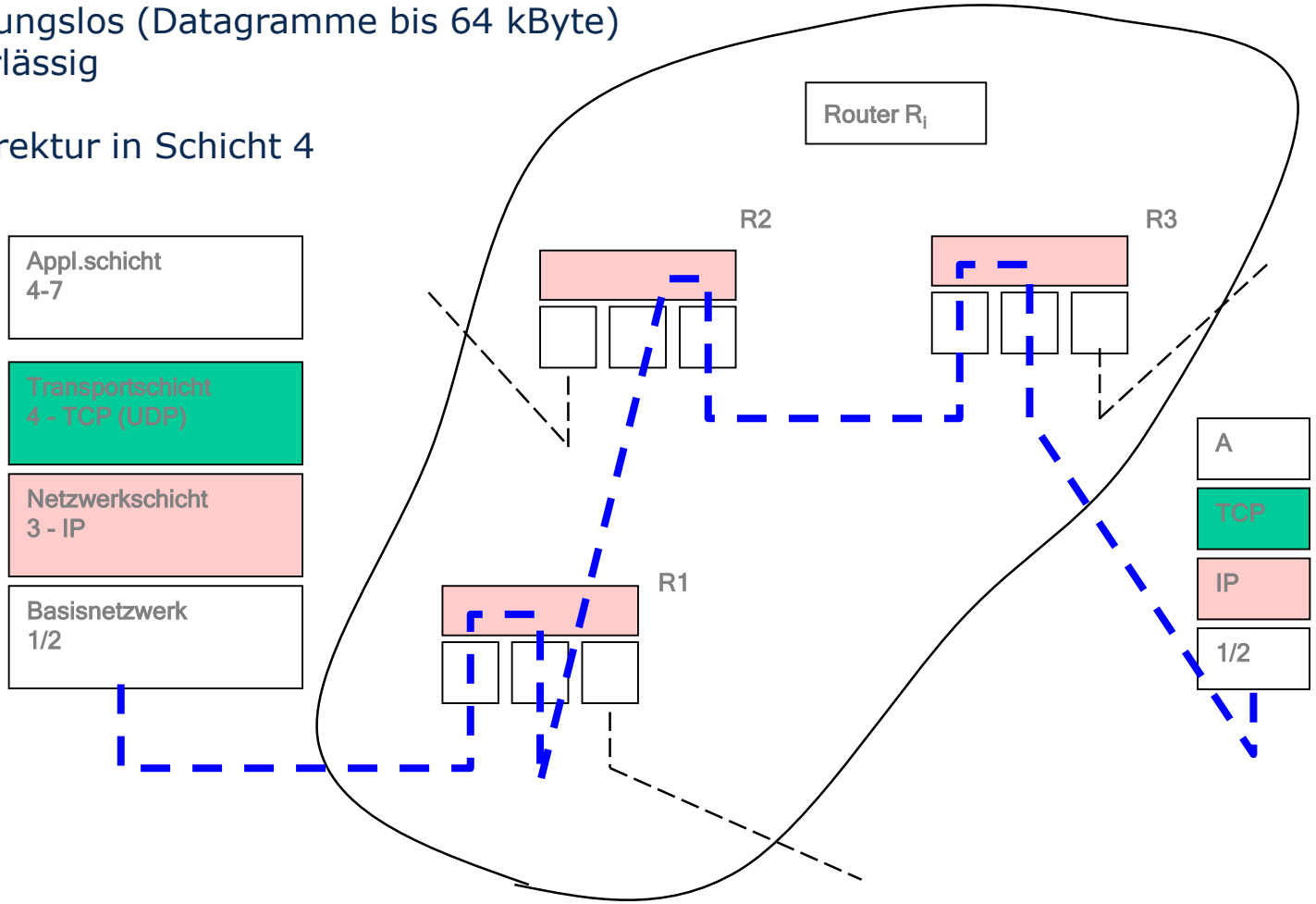
• „experimental"	Testphase
• „informational"	Diskussion, evtl. prorietär, …
• „historic"	Protokoll veraltet
• „required"	zwingend zu verwenden
• „recommended"	zur Verwendung empfohlen
• „not recommended"	… nicht empfohlen

Internet - Architektur

Paketvermittlungsnetz

- verbindungslos (Datagramme bis 64 kByte)
- unzuverlässig

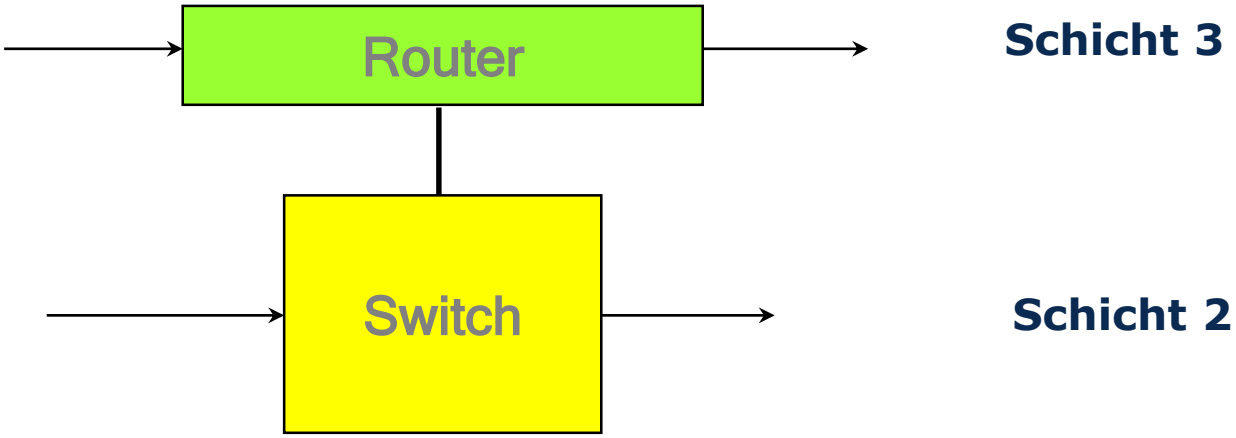
Fehlerkorrektur in Schicht 4



IP-Switches

Kombination von Router und Switch

Bei neuen Datenströmen zunächst Routing,
bei längerer Dauer ggf. Durchschalten von Verbindungen mittels Switching („Layer-3-Switches“)



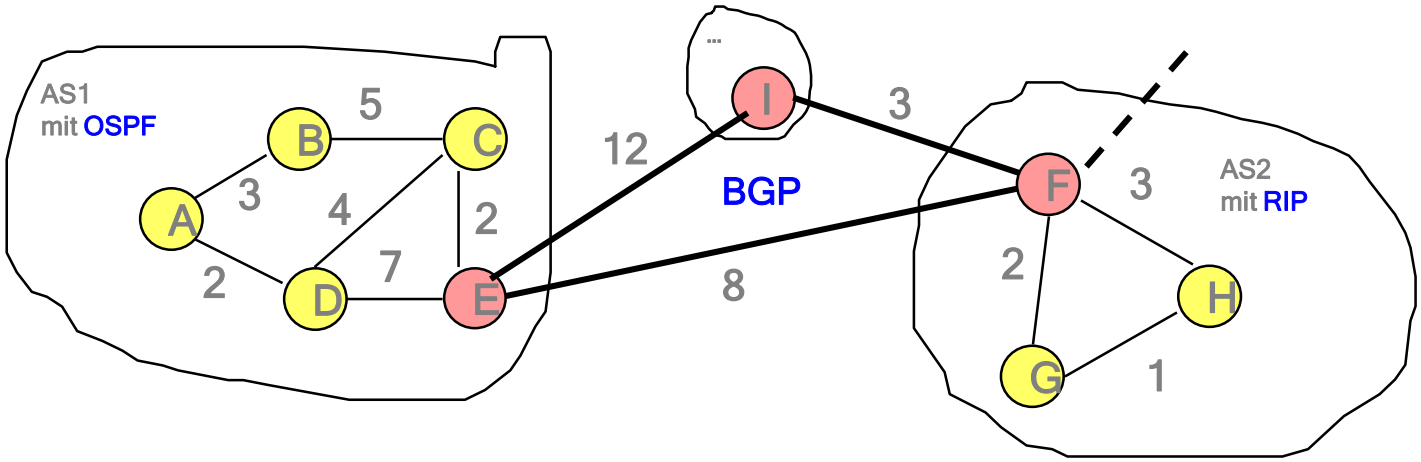
Teilweise auch Berücksichtigung von Informationen der Transportschicht („Layer-4-Switches“)

Internet „Netz von Netzen“

Gesamtnetz-Wegewahl nur für kleinere Netze beherrschbar

→ **Unterteilung des Internet in autonome Systeme (AS)**
2008: ca 110 000 AS; Anmeldung durch Provider bei RIR`s

- AS-internes Routing
  in verschiedenen AS u.U. verschiedene Routingprotokolle
- Routing zwischen den AS
  AS-Adressierung über 16-bit-ID, neuerdings auch über 32bit-ID



Internet „Netz von Netzen“

intra-AS Routing Protocol (bzw. Interior Gateway Protocol)

z.B. RIP (Routing Information Protocol)

OSPF (Open Shortest Path First)

IS-IS (Intermediate System to Intermediate System Protocol), ähnlich OSPF

EIGRP (Enhanced Interior Gateway Routing Protocol), Cisco proprietär

inter-AS Routing Protocol (bzw. Exterior Gateway Protocol)

BGP (Border Gateway Protocol)

AS - Hierarchie

Tier-3 untere Schicht (Edge-Networks), AS ausschließlich für Endkunden

- Verkehr zu anderen AS über „uplink" zu übergeordneten Systemen

Tier-2 mittlere Schicht, große Systeme

- besitzen Endkunden und
- vermitteln AS-Verkehr, aber nicht weltweit

Tier-1 obere Schicht , sehr(!) große Systeme

- keine Endkunden, vermitteln weltweit Verkehr zwischen AS
- ca. 10 Unternehmen weltweit (Verizon, Level3, …)

weltweit (2008) größter **Austauschknoten** DE-CIX (Frankfurt)

- Vermittlung für 240 AS, 350 Gbit/s, ausbaufähig
  Layer-3-Switch-System mit Redundanz, geringe Verzögerung

Verkehrsabkommen (Transitvolumen, …)

Format eines IP-Headers

Bit	1 … 4	5 … 7	8 … 15	16 .. 19	20 … 31
	Version | IHL	Service Typ	Gesamtlänge		
	Identifikation		Flags |	Fragment Offset	
	Time To Live	Protocol	IP-Header-Prüfsumme		
	IP Source Adresse				
	IP Destination Adresse				
	Optionen …		Füllzeichen		

IHL	Header Length (variabel)
Type of Service:	Angaben zur gewünschten Dienstqualität
DF	„Don´t Fragment" – Paket darf nicht zerlegt werden
MF	„More Fragments" – weitere Fragmente des Pakets folgen
Fragment Offset	Einordnung des aktuellen Fragments
Time to Live	Max. Lebensdauer (in Sek., in der Praxis Angabe der Hops)
Protocol	Angabe des Transportprotokolls
Header Checksum	Fehlerprüfung (nur Header, nicht ganzes Paket)

IP - Paketbearbeitung

Kontrolle

- Headerlänge
- IP-Versionsnummer
- Paketlänge
- Prüfsumme
- Lebenszeit
- Protokollidentifikation
- Adreßklasse bei Quell- und Zieladresse

Fehlerfall? Fehlernachricht an Partner-Router (über Protokoll ICMP)

Korrekt? Lebenszeit um 1 dekrementieren
Header neu berechnen (Kontrollsumme)
Paket weiterleiten

IP - Prüfsequenz

RFC 1071

Berechnung über Header, einschließlich evtl. Optionen

1. Prüfsequenz-Feld im Header auf Null setzen
2. Summe von 16-Bit-Worten (2Byte) bilden, ggf. 1 Byte auffüllen
3. Einerkomplement der Summe → Prüfsequenzfeld

Kontrolle beim Empfänger

1. Einerkomplement über Header berechnen
2. i.O. falls 0xffff (-1)
   sonst fehlerhafter Header

→ Neuberechnung bei jedem Hop erforderlich (TTL)

IP - Prüfsequenz / Beispiel

Sender

	16-Bit-Worte	
	10110011 01011000	
	01001011 00101110	
	10001100 00010011	
	00000000 00000000	
1	10001010 10011001	Σ_S
	10001010 10011010	Übertrag zuaddieren
	01110101 01100101	Einerkomplement → Header

Empfänger

	16-Bit-Worte	
	10110011 01011000	
	01001011 00101110	
	10001100 00010011	
	01110101 01100101	
1	**11111111 11111110**	Σ_E
	11111111 11111111	Übertrag zuaddieren

IPv4 - Adressierung

32-Bit-Adressen (4 Byte)

⇒ 2^{32} *(= 4 294 967 296)* mögliche Adressen

Vergabe durch **IANA** (Internet Assigned Numbers Authority)
an regionale Organisationen, z.B. RIPE NCC (Network Coordination Centre)
Unterverteilung durch Internetprovider

Schreibweise: byteweise dezimal durch Punkt getrennt
z.B. **141.76.40.3**

Problem: Adressen enthalten keine Routinginformationen
Internetrouter ← 2^{32} Tabelleneinträge !!!!

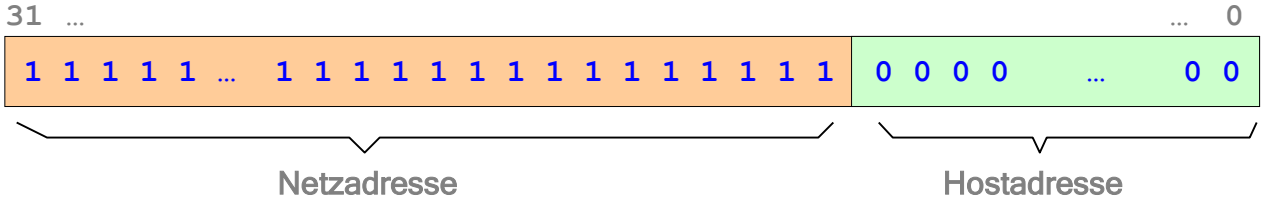
Lösung: Unterteilung IP-Adresse in Netz- und Hostadresse
Internetrouter ← nur noch ***ein*** Eintrag pro Netzwerk

Wieviel Bits sollten für Netzadresse reserviert werden ?
➔ variable Anzahl,
da große und kleine Netzwerke existieren

IPv4 – Strukturierung in Netz- und Hostadresse

Regelung über Netzmaske, (veraltet)

z.B. 255.255.255.0



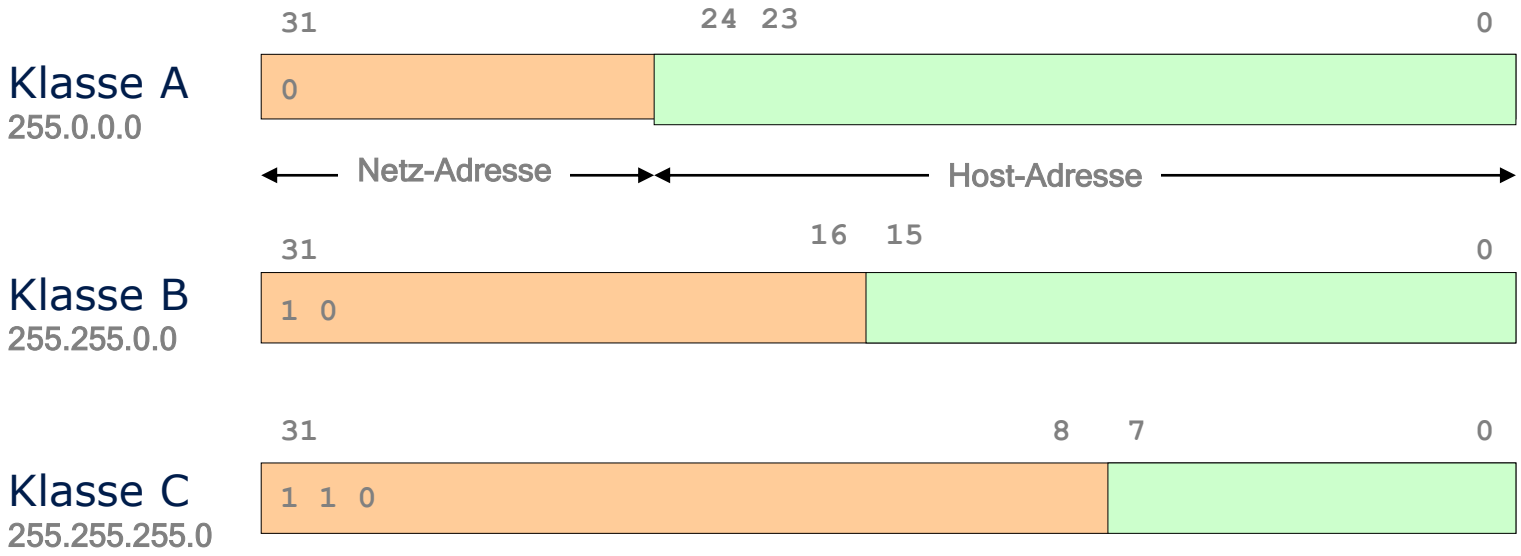
Berechnung

Netzadresse := IP-Adresse ∆ Maske

bitweise logische UND-Verknüpfung

IP - Netzklassen

primäre Netzwerkadressen (Unterscheidung durch führende Bits)



Klasse D (Präfix 1110) für IP-Multicast
Klasse E (Präfix 1111) für experimentelle Zwecke.

Besondere Adressen	Default	0.0.0.0
	Schleife	127.0.0.1
	Netzadr.	alle Host-Bit=0
	Broadcastadr.	alle Host-Bit=1

IP – Adreßraum

Klasse	Adreßraum	Anzahl der Netze/Hosts	Anteil
A	0.xxx.xxx.xxx 127.xxx.xxx.xxx	N: $126 = 2^7 - 2$ H: $16777214 = 2^{24} - 2$	2^{31} => 50%
B	128.0.xxx.xxx 191.255.xxx.xxx	N: $16382 = 2^{14} - 2$ H: $65534 = 2^{16} - 2$	2^{30} => 25%
C	192.0.0.xxx bis 223.255.255.xxx	N: $2097152 = 2^{21}$ H: $254 = 2^8 - 2$	2^{29} => 12.5%
D	224.0.0.0 bis 239.255.255.255	Adressen $268435456 = 2^{28}$	2^{28} => 6.25%
E	240.0.0.0 bis 255.255.255.255	Adressen $268435456 = 2^{28}$	2^{28} => 6.25%

reserviert für Intranet	Adreßraum	Kommentar Adressen nicht weltweit eindeutig !
A [255.0.0.0]	10.0.0.0	Intranetadressen können in privaten Netzen verwendet werden. Im Internet werden IP-Pakete mit diesen Adressen nicht geroutet!
B [255.240.0.0]	172.16.0.0	
C [255.255.255.0]	192.168.0.0	

IPv4 - Subnetzbildung

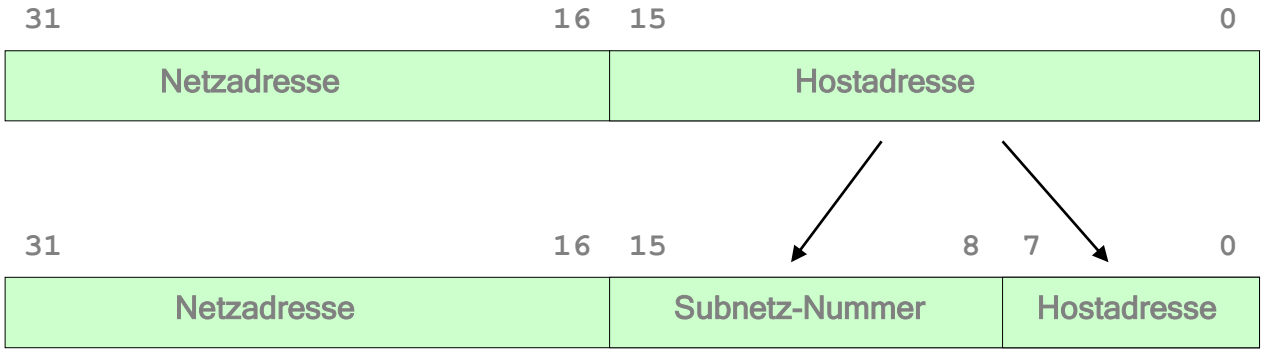
Probleme mit IPv4 :

Adressraum zu klein für umfassende Nutzung

Ungünstige Größen der Netzklassen, viel ungenutzter Adreßraum

⇒ Unterteilung in Subnetze, z.B.

(B-Klasse)-Netz 141.76.0.0 Maske 255.255.0.0



Subnetze 141.76.x.0 Maske 255.255.255.0

IP – klassische Routingtabellen

Ziel-Netzadresse	Ausgabeport	Distanz	nächster Router, evtl. kompletter Übertragungsweg
…	…	…	…
141.76.0.0	eth1	4	192.168.1.2
0.0.0.0	eth0	?	192.168.1.1

Keine Maskenangabe ! Maske wird aus Netzklasse abgeleitet

- A-Adreßraum weitgehend ungenutzt
- B-Adreßraum für die meisten Netzwerke zu groß
- C-Adressenanzahl unzureichend

für alle Zielnetze müssen Tabelleneinträge existieren

- Provider haben oft
  keinen einheitlichen, zusammenhängenden Adreßraum
- Aggregation nicht möglich (Zusammenfassung von Netzen)

Probleme mit Netzklassen

Anfang der 90-er Jahre extreme Steigerung der Netzadressenanzahl

- **IP-Adressenmangel**
- riesige Routingtabellen

fehlende Unterstützung von **VLSM** (Variable Length Subnet Mask)

durch Routingprotokolle RIP, IGRP, …

- Problem für größere Netze (z.B. B- Klasse-Netze)
  mit Unterteilung in kleinere Netze mit unterschiedlicher Maskenlänge

Lösung

- **CIDR** (Classless Inter Domain Routing)

CIDR (Classless Inter Domain Routing)

1993 RFC 1518 / RFC1519

Routingprotokolle mit VLSM-Unterstützung, z.B. OSPF, EIGRP, RIP-2 ermöglichen CIDR

Neue **Notation**
Netzklassen werden ignoriert
Angabe der Maske entfällt, dafür Anzahl der Bits der Netzadresse

z.B.	141.76.40.3**/24**	statt	141.76.40.3
		mit Maske	**255.255.255.0**

Aggregation von IP-Adressen zu Adreßblöcken beliebig,

Vergabe der Adreßblöcke z.B. nach

- Kontinent, Land
- Provider

CIDR - Routing

- Eine Routingtabelle für alle Netzadressen
- Tabelleneinträge mit IP-Adresse und Maske
- Reduktion der Tabellengrößen durch Aggregation

  z.B. nur 1 Eintrag für gesamten Verkehr zu einem Provider
- Subnetzbildung vor Ort

- Problem Umzug → widersprüchliche Einträge ?

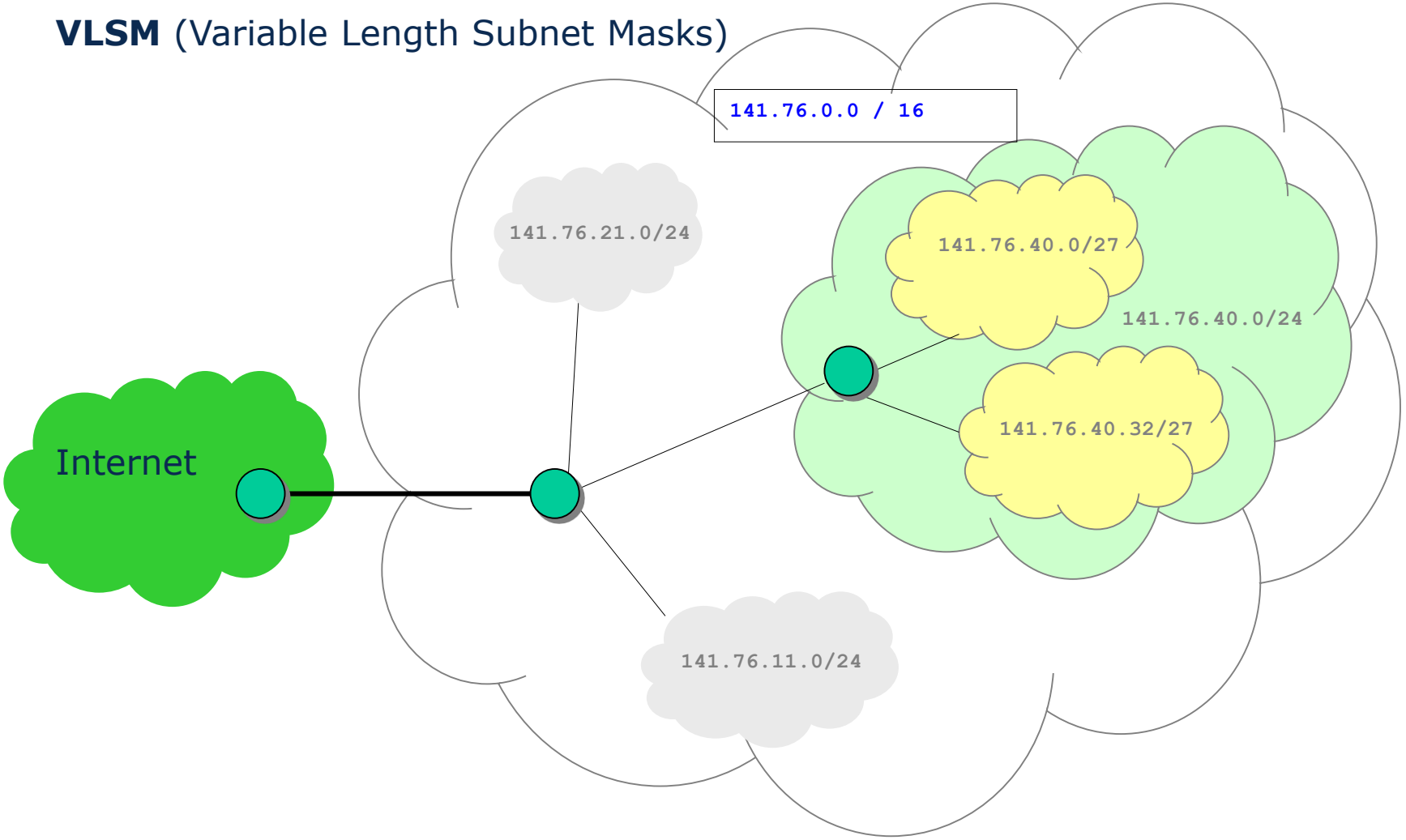
Destination	Port	…
141.76.16.0/24	eth0	
141.76.16.32/27	eth1	

2. Netz ist Untermenge vom 1. Netz !!!

Wohin?
Absenden in Netz mit längerer Adresse
(**longest prefix matching**)

Subnetz - Hierarchie

VLSM (Variable Length Subnet Masks)

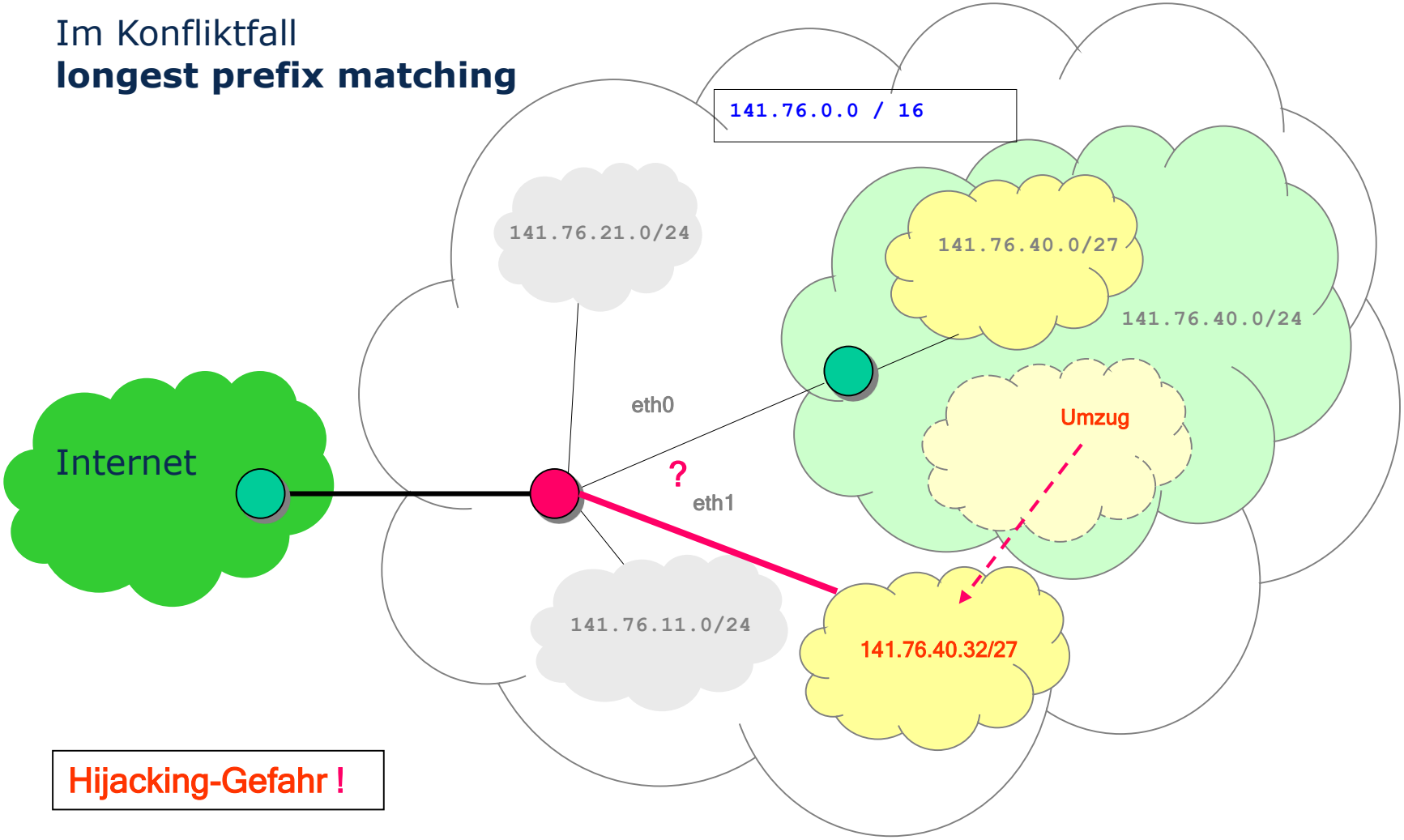


Subnetz - Beispiel

	Maske	
Base-Net	**10001101.01001100**.00000000.00000000	141.76.0.0/16
Subnet 1	**10001101.01001100.00101000**.00000000	141.76.40.0/24
Subnet 1-1	**10001101.01001100.00101000.000**00000	141.76.40.0/27
Subnet 1-2	**10001101.01001100.00101000.001**00000	141.76.40.32/27
Host #1	**10001101.01001100.00101000.001**00001	141.76.40.33/27
Host #2	**10001101.01001100.00101000.001**00010	141.76.40.34/27
Host #3	**10001101.01001100.00101000.001**00011	141.76.40.35/27
...	...	...
Host #29	**10001101.01001100.00101000.001**11110	141.76.40.62/27
Broadcast	**10001101.01001100.00101000.001**11111	141.76.40.63/27
Subnet 1-3	**10001101.01001100.00101000.010**00000	141.76.40.64/27
Subnet 1-4	**10001101.01001100.00101000.011**00000	141.76.40.96/27
...	...	...

Subnetz - Hierarchie

Im Konfliktfall
longest prefix matching



Hijacking-Gefahr !

Regeln für Arbeitsweise der Router

Regel 1 **Basic Match**

Selektion aller in Frage kommenden Einträge der Routingtabelle

Regel 2 **Longest Match**

Auswahl des Eintrages mit der längsten Netzadresse

Regel 3 **Weak Type of Service**

Berücksichtigung von Prioritäten, …, selten implementiert

Regel 4 **Best Metric**

Auswahl des optimalen Weges

Regel 5 **Vendor Policy**

finale Entscheidungskriterien,
z.B. im Zweifelsfall Route mit geringster Auslastung

ICMP (Internet Control Message Protocol)

RFC 792 bzw. **RFC 1885** für ICMPv6

Hilfsprotokoll für den Austausch von Internetsteuernachrichten

- Fehleranzeigen (z.B. fehlerhafte IP-Pakete, Nichterrreichbarkeit, …)

  zwischen Host und Router
  zwischen Routern

- Testnachrichten
- Flußsteuernachrichten (ähnlich "Choke-Pakete")

ICMP-Daten werden als in Form von IP-Paketen übertragen

Nutzung durch IP
und spezielle Administrationsprogramme, wie *ping, traceroot, …*

ICMP – Nachrichtenauswahl

Typ	Kode	Kommentar
0	0	Echo Antwort
3	0	Netzwerk nicht erreichbar
3	1	Host nicht erreichbar
3	2	Protokoll nicht erreichbar
3	3	Port nicht erreichbar
3	4	Fragmentierung erforderlich, DF gesetzt
3	5	Falsche Routingangabe bei Source Routing
3	6	Netzwerk unbekannt
3	7	Host unbekannt
4	0	Routerüberlastung, Warteschlangenüberlauf
8	0	Echo Anforderung
11	0	IP-Paket verworfen (Lebenszeit abgelaufen)
11	1	Paketfragment (Lebenszeit abgelaufen)
12	0	Fehlerhafter IP-Paket-Header
13	0	Zeitstempel Anforderung (Sendezeitpunkt, …)
14	0	Zeitstempel Antwort (Sende-, Empfangs- und Antwortabsendezeitpunkt)

ICMP – Programme ping und traceroot

PING (Packet Internet Groper)

Test auf Erreichbarkeit eines Rechners mit gegebener IP-Adresse, zusätzlich Messung der Übertragungszeit, ggf. timeout-Anzeige

nutzt ICMP mit
Typ/Kode=8/0 (echo request) und Typ/Kode =0/0 (echo reply)

TRACEROOT

ermittelt Weg (Router-Folge) zu Rechner mit gegebener IP-Adresse (Implementierung auf UDP bzw. IP)

Senden von ICMP-Nachrichten mit schrittweise steigender TTL

TTL=1 → beim nächsten Router Nachricht 11/0 (TTL expired),
TTL=2 …,
TTL=n → Nachricht 3/3 (port unreachable) oder 0/0 (echo reply)

RIP (Routing Information Protocol)

klassisches AS-internes Routingprotokoll (1982 Vorläufer in BSD-UNIX)

RFC 1058, RFC 2453

Distanzvektoralgorithmus

- Distanzvektoren enthalten bis zu 25 Tabelleneinträge (Netze)
- Metrik ist Anzahl der Teilstrecken eines Weges (Hops) (jede Streckenbewertung gleich 1)
- Maximale Distanz 15 Hops

Implementierung

- UDP, Port 520
- keine Unterstützung von CIDR
  → **RIPv2** mit Unterstützung von VLSM und CIDR

RIP-Informationen (Advertisements)

- Austausch aller 30 s mit sämtlichen Nachbarn
- Timeout 180 s, danach Ziel unerreichbar, Hopzähler auf 16

OSPF (Open Shortest Path First)

AS-internes Routingprotokoll

RFC 2328

Link State Algorithmus nach Dijkstra

- Hierarchische Unterteilung eines AS möglich (Backbone-Bereich und untergeordnete lokale Bereiche)
- Metrik wird manuell von Administratoren festgelegt

Implementierung

- kein Transportprotokoll, Übertragung direkt über IP
- Unterstützung von VLSM und CIDR

OSPF-Advertisements

- Austausch per Broadcast im gesamten AS nach Änderungen bzw. aller 30 min

BGP (Border Gateway Protocol)

Standardprotokoll für Inter-AS-Routing, **RFC 1163, RFC 4271**

- Pfadvektorprotokoll

  vollständige Pfade wegen Schleifenvermeidung
  (ASx → Asy → … →Zielnetz)

- **BGP-Session**

OPEN	Verbindungsaufbau Austausch der Routingtabellen mit Nachbarn (u.U. mehrere 100000 Einträge)
UPDATE	danach nur Pfadaktualisierungen
KEEPALIVE	Meldung Arbeitsbereitschaft, z.B. aller 60 sec
NOTIFICATION	Fehlermeldungen, Verbindungsabbau

BGP (Border Gateway Protocol)

- Implementierung TCP-basiert
  Unterstützung von VLSM, CIDR und MPLS
- komplexe Metrik (Berücksichtigung kommerzieller/politischer Vorgaben)

  Peering bilaterale Abkommen zwischen Netzbetreibern
  Policies Regelung des erlaubten Nachrichtenverkehrs
- manuell verwaltete Tabelleneinträge

- Routingstrategien
  - Routing-Algorithmus nach Bellmann-Ford → "best effort"
  - Lastminimierung, z.B. durch „Hot Potatoe"-Algorithmus
  - QOS-Optimierung,
    z.B. Garantien für Paketverlustrate, Verzögerung, Jitter

    → Gefährdung der Grundphilosophie des Internet
    (Gleichbehandlung der Verkehrsströme)

Auszug aus einer Routingtabelle (ZIH/INF-Router)

ip route

D – EIGRP (Enhanced Interior Gateway Routing Protocol), EX – External,
O – OSPF (Open Shortest Path First), …, C – Connected, S – Static

Statische Routen

S 141.76.75.0/24 [1/0] via 141.76.29.65
S 141.76.40.0/22 [1/0] via 141.76.29.33

Direkt verbundene Routen

C 141.76.8.0/24 is directly connected, Vlan8
C 141.76.29.80/28 is directly connected, Vlan800

Dynamische Routen über EIGRP-Routingprotokoll mit ZIH-Router

D EX 192.168.168.96/28 [170/3072] via 141.30.1.181, 1w0d, Vlan1959
D EX 192.168.168.112/28 [170/3072] via 141.30.1.181, 1w0d, Vlan1959

Router im ZIH ist **Default-Router** für alle anderen Adressen

D*EX 0.0.0.0/0 [170/768] via 141.30.1.181, 1w0d, Vlan1959

IP-Paket-Fragmentierung

Problem

IP-Paketgröße: max. 64 kByte zulässig

für viele Netzwerke ist dies zu groß
z.B. Ethernet-Frame: max. 1500 Byte Payload

Lösung

jedem Netzwerk wird eine **MTU** (Maximal Transfer Unit) zugeordnet
(gemessen in Byte, z.B. Ethernet-MTU = 1500)

IP-Schicht muß ggf. IP-Pakete in mehrere Fragmente zerlegen,
bzw. wieder aus Fragmenten zusammensetzen

Header enthält Flags mit Kennzeichen und Fragment-Offsetadresse
Flags: (fragmentiert ja/nein) und (letztes Fragment ja/nein)

Fragmente dürfen ggf. weiter fragmentiert werden !

IP – Routing in einfachen Netzwerken, z.B. LAN

Kommunikationsniveau zu Stationen

- innerhalb Netzwerk **direkt** auf Schicht-2-Niveau
- außerhalb Netzwerk **indirekt** über Gateway

Gateway = Default-Router (Vermittlung zu allen anderen Netzen)

Trivialrouting Test { Rechner im eigenen Netzwerk? }
bzw. { eigene IP Δ Maske = Partner-IP Δ Maske ? }

→ ja: Empfänger direkt erreichbar
→ nein: Weg über Gateway wählen

Erforderliche Informationen → TCP/IP-Konfigurationsdaten

- eigene IP-Adresse
- Netzmaske
- IP-Adresse Gateway, evtl. mehrere
- zusätzlich IP-Adresse DNS-Server

ARP (Address Resolution Protocol) und RARP (inverse ..)

Problem

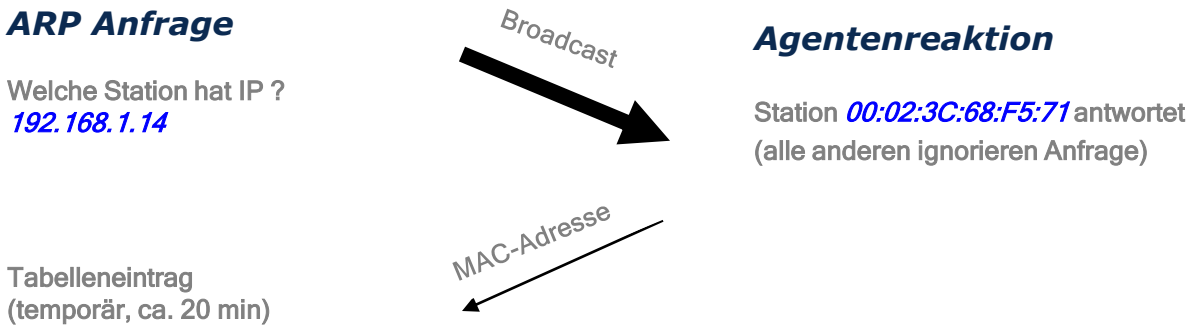
direkte Kommunikation in Netzwerken
erfordert Kenntnis der Partner-MAC-Adresse

IP: Zieladresse gibt Nutzer vor
MAC: eigene Adr. Ist bekannt; Ziel ?

IP-Adresse	MAC-Adresse
…	…
192.168.1.14	?

Lösung in LAN

Auf jeder Station läuft ein ARP-Agent,
dieser reagiert auf Broadcast-Anfragen



DHCP (Dynamic Host Configuration Protocol)

RFC 2131

manuelle Eingabe der TCP/IP-Konfigurationsdaten

- sinnvoll für stationäre Computer
- problematisch für mobile Nutzer
  (häufiger Wechsel, keine Administrationsrechte, evtl. Fehleingaben)

DHCP ← Automatisierung der Konfiguration

- Client-/Serverarchitektur
- Server kennt Netz-Standarddaten
  und manuell zugeteilten IP-Adreß-Pool
- Client fordert Konfigurationsdaten per Broadcast an;
  Server teilt Daten zu
- Reservierung der Datenzuordnung (leasing time)

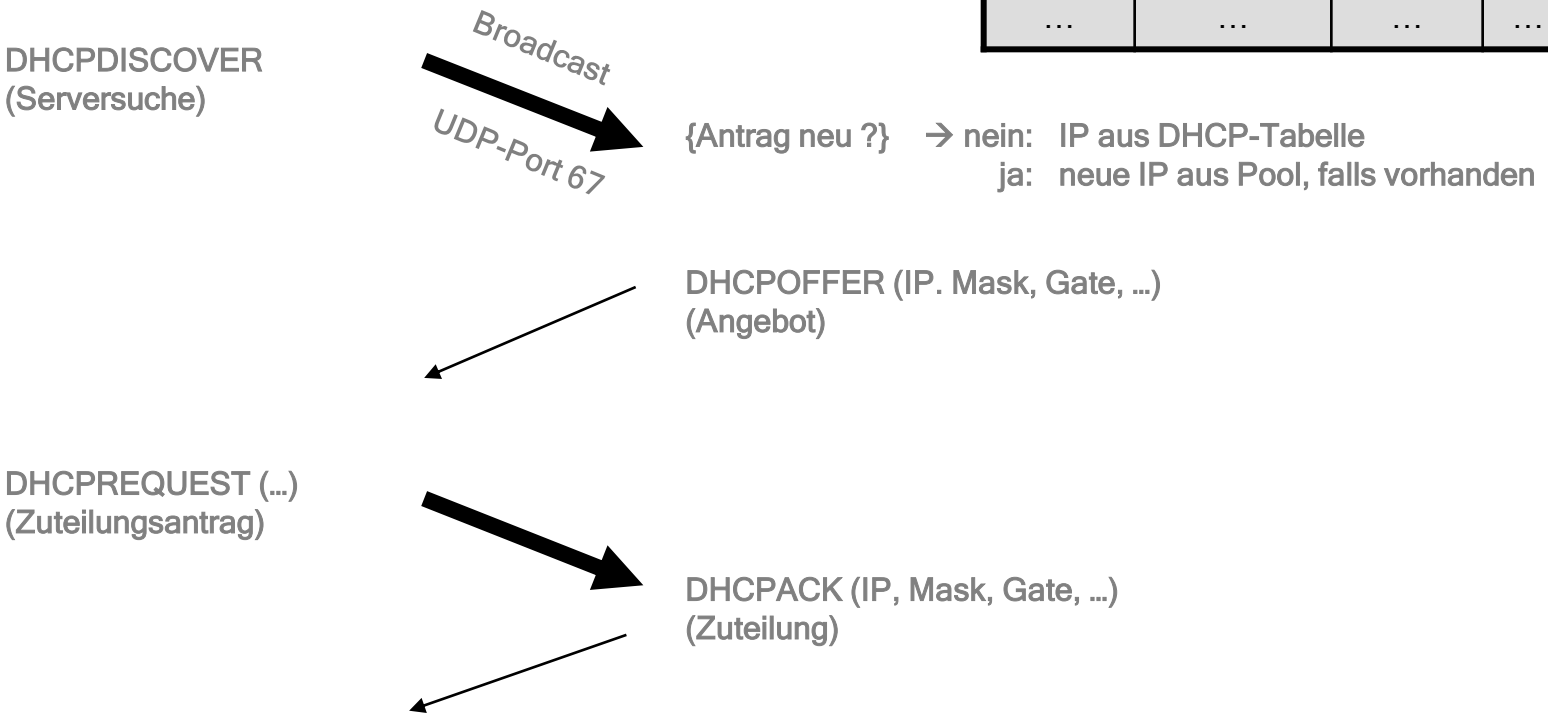
- keine Sicherheitsmechanismen
- kein Informationsaustausch zwischen DHCP-Servern

DHCP

Client **Server**

DHCP-Tabelle
(zugeteilte IP-Adressen)

IP-Adr.	MAC-Adr.	Leased time	…
…	…	…	…
…	…	…	…



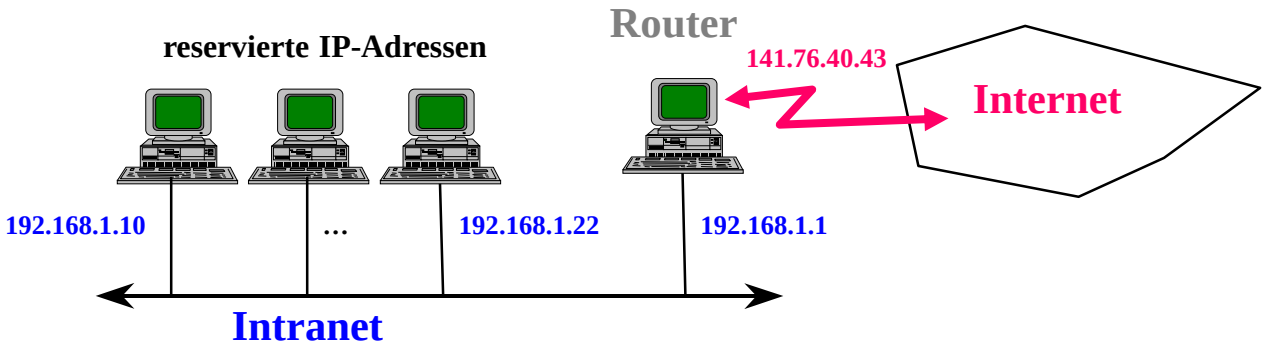
IP – NAT/PAT (Network Address and Port Translation)

Problem: Internet-Anschluß für Intranet

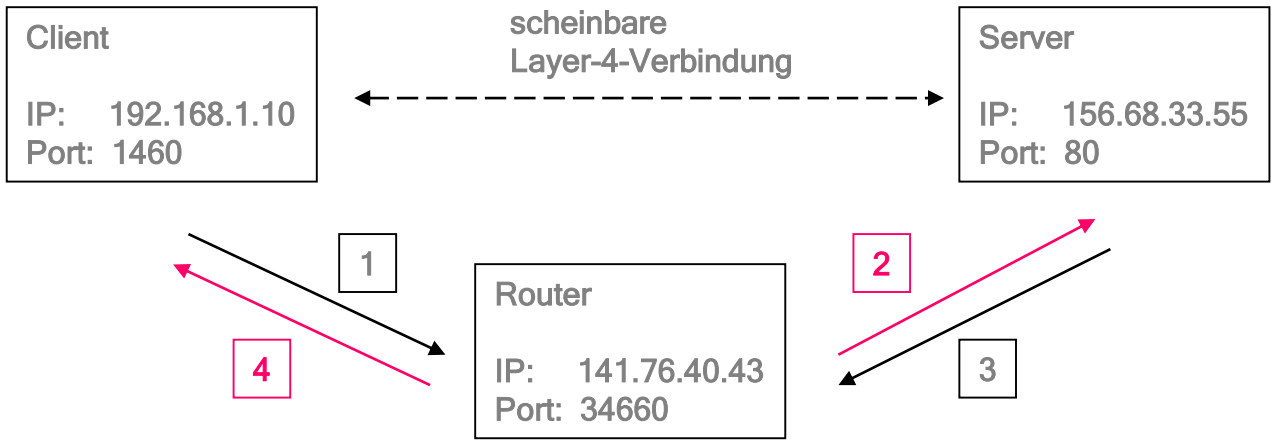
private Adressen (wegen Adreßmangel, Kosten) nicht im Internet routbar !

Lösung: Abbildung der Intranetadressen auf „vollwertige" IP-Adressen

- Static NAT: feste Zuordnung, z. B. 192.168.1.10 => 141.76.40.43
- Dynamic NAT: dynamische Zuordnung einer Adresse aus einem Adresspool
  → Probleme bei Verbindungsaufnahme von außen
- NAT/PAT: erweitertes Dynamic NAT
  Abbildung von Socketadressen (IP-Adresse + Port-Nummer)



NAT/PAT - Beispiel



1. Client sendet IP-Paket an Router erstmals von 191.168.1.10:1460
2. Router fordert vom Betriebssystem einen freien Port und erhalt Port 34660, manipuliert Absender-Socket-Adresse in 141.76.40:434660 und sendet an Server

   Tabelleneintrag: Router-Port 34660 — Socketadresse im Intranet 192.168.1.10 : 1460

3. Server erhält IP-Paket (mit Quell-IP des Routers 141.77.40.43) und antwortet
4. Rückmanipulation von IP und Port durch Router

Löschung der Routertabelleneinträge bei Layer-4-Verbindungsabbau

NAT/PAT - Beurteilung

- eine IP-Adresse ermöglicht mehr als 60000 Verbindungen!
  (16-bit-Portnummern)

  → mittelfristige Lösung des Adreßmangelproblems bei IPv4
  hohe Akzeptanz

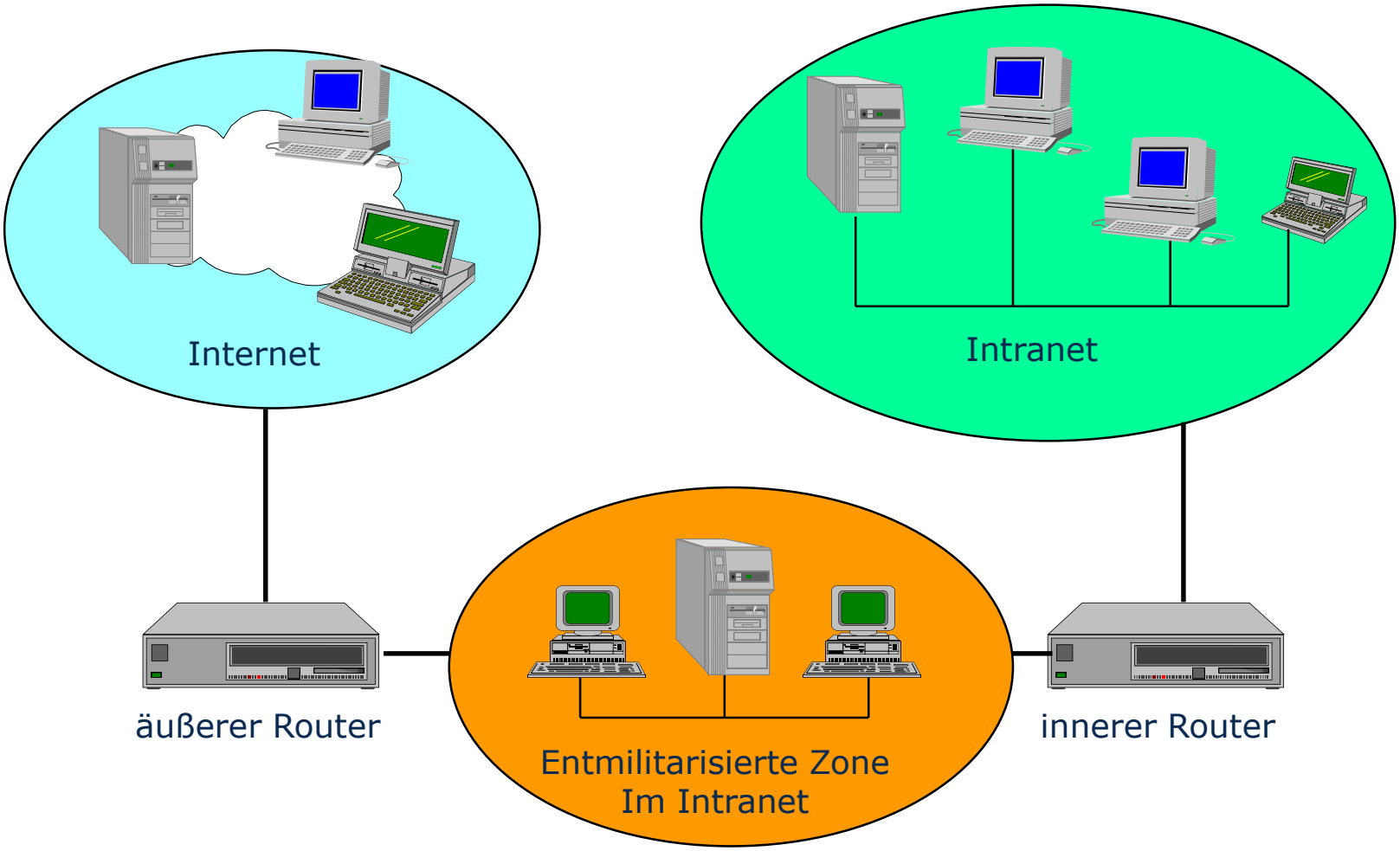
- eigentlich Realisierung eines Socket-Proxy-Servers
  arbeitet auf Schicht 3, nutzt aber Eigenschaften der Schicht 4 (unsauber)

- Manipulation der IP-Adresse,
  evtl. Probleme bei Anwendungen (P2P-Netze, …)

- Serverbereitstellung für Internet erschwert
  erfordert Freischaltung eines Routerports mit permanenter Zuordnung
  Port < -- > Intranet-Socketadresse

- Intranetrechner nur über freigeschaltete Routerports angreifbar.
  → einfache Firewall-Realisierung

Paketfilter als Firewalls

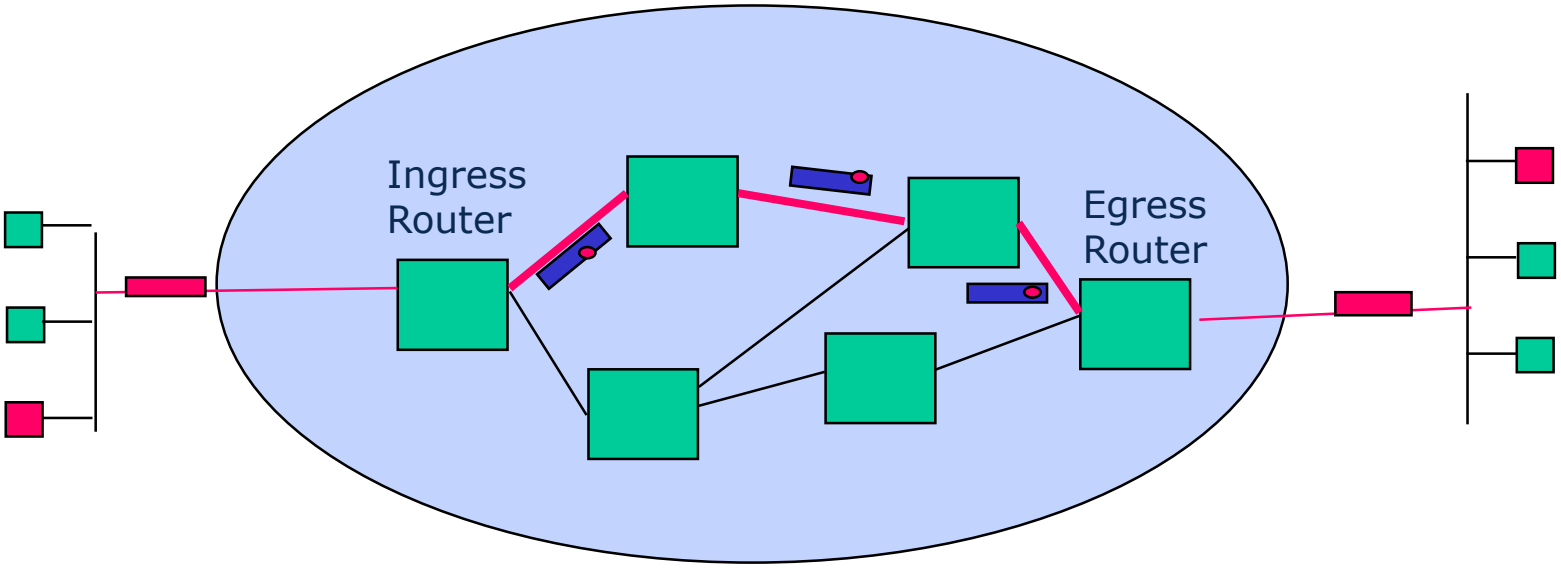


Multiprotocol Label Switching (MPLS)

- Source Routing
- effiziente Weiterleitung von Paketen entlang vordefinierter Pfade gemäß ihrer sog. Forward Equivalence Class (FEC),
- gleiche Behandlung aller Pakete eines Datenstroms
  (z. B. Videokonferenz)
- gesteuert durch Labels
  (Kennzeichnung für aufeinander folgende Pfadabschnitte)
  Gewährleistung von Dienstqualität
- größere Pakete als bei ATM (z. B. 1500 Byte)
- Basis für

Layer-3-VLAN	(Virtual LAN)	Anschluß:	Bridge-Port
Layer-3-VPN	(Virtual Private Network)		Router-Port

Multiprotocol Label Switching (MPLS)



1. Pfad („Tunnel“) durch das Gesamtnetz wird ermittelt (z. B. mittels IP-Routing)
2. Label Distribution Protocol (LDP) legt Labels für diesen Pfad fest
3. Ingress Router markiert eingehende Pakete gemäß FEC mit passendem Label
4. Effiziente netzinterne Weiterleitung gemäß vorgegebenem Pfad (vgl. ATM); dabei Umwandlung Eingangslabel -> Ausgangslabel durch jeden Switch
5. Egress Router entfernt Label und leitet Pakete ins Zielsystem weiter

Layer-3-VLAN

Virtuelles IP-Netzwerk

Anpassung an die organisatorische Unternehmensstruktur
(Trennung von physikalischer und logischer Struktur)

- Anbindung entfernter Bürostandorte („Intranet-VPN“)
- Kommunikation mit Kundenstandorten („Extranet-VPN“)
- Einbindung mobiler Mitarbeiter („Remote-Access-VPN“)

für Endgeräte transparent,
d.h. keine Änderung der NIC, Treiber, … erforderlich

IP-Paketverschlüsselung im Internet (PPTP, IPsec, …)

Realisierung z.B. über ATM als Backbone-Technologie – ELAN

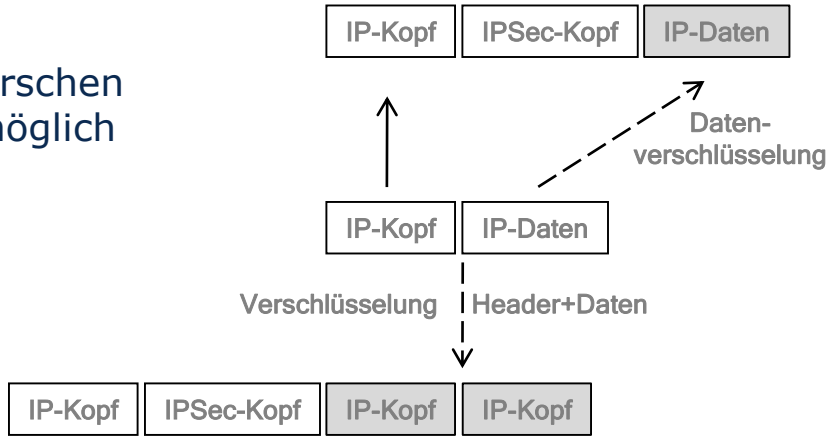
- MPoA v1.0 (Multiprotocol over ATM)
- LANE v2.0 (LAN Emulation)

IPSec (IP Security)

Transportmodus

- IP-Header mit zusätzlichem IPSec-Header
- Verschlüsselter Datenteil des IP-Pakets

→ wenig Overhead
alle Stationen müssen IPSec beherrschen
Verkehrsanalyse durch Angreifer möglich



Tunnelmodus

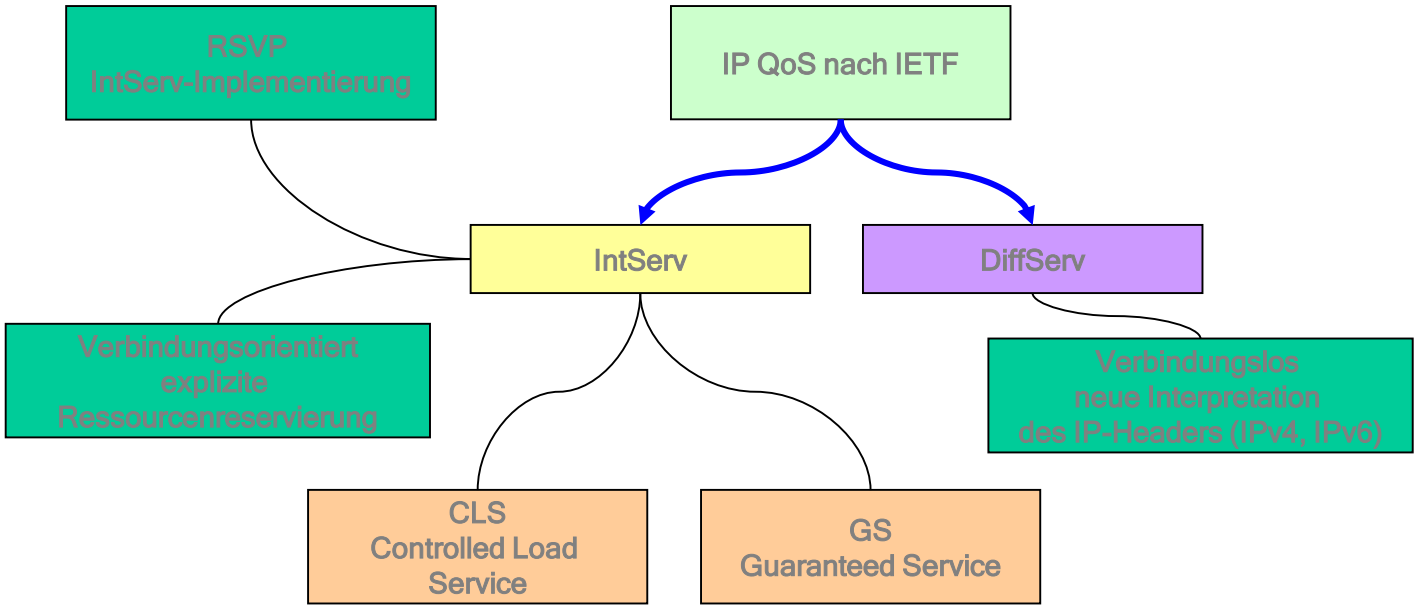
- IP-Paket wird komplett verschlüsselt, neuer IP- und IPSec-Header
- Verkehr über Gateway

→ Overhead größer
nur Gateway muß IPSec beherrschen
Angreifer können nur Anfangs- und Endpunkt des Tunnels feststellen

Quality of Service (QoS) in IP-Netzwerken

IETF

- Integrated Services Networks (IntServ)
- Differentiated Services Networks (DiffServ)



IETF-QoS

IntServ

- reserviert QoS für konkrete Anwendungen, Notation in „QOS-Profiles“ (z.B. Videokonferenz, B=128 Kbit/s, δPacket=10ms) – RFC 1633
- explizite Ressourcenreservierung entlang des Übertragungsweges (virtuelle Verbindungen)

DiffServ

- verbindungslos, keine explizite Ressourcenreservierung, skalierbar
  Aufteilung der Nachrichtenströme in QoS-Klassen
  Spezifizierung im IP-Header
- Dienstklassen → Zuordnung von „Per-Hop- behaviors“ (PHB):
  Regeln, wie mit Paketen einer Dienstklasse zu verfahren ist
- Aggregation von Paketen gleicher Dienstklasse zu Verkehrsströmen möglich

Ressource Reservation Protocol (RSVP)

RFC 2205, Implementierung des IntServ- Ansatzes

RSVP -Traffic wird in IP-Pakete „verpackt"
Pfadbestimmung mittels IP-Routing, danach Reservierungen

RSVP ist für nicht-RSVP-fähige Netzwerkknoten transparent

Robustheit gegen Router-/Leitungsausfälle → „Soft States"

Router behalten Verbindung nicht bis zum expliziten Beenden
Reservierung wird periodisch wiederholt

Komponenten

Packet Classifier	Zuordnung der Pakete zu QoS-Klassen
Admission Control	Beurteilung, ob Ressourcen für QOS ausreichen
Packet Scheduler	Koordination Paketausgabe nach QOS

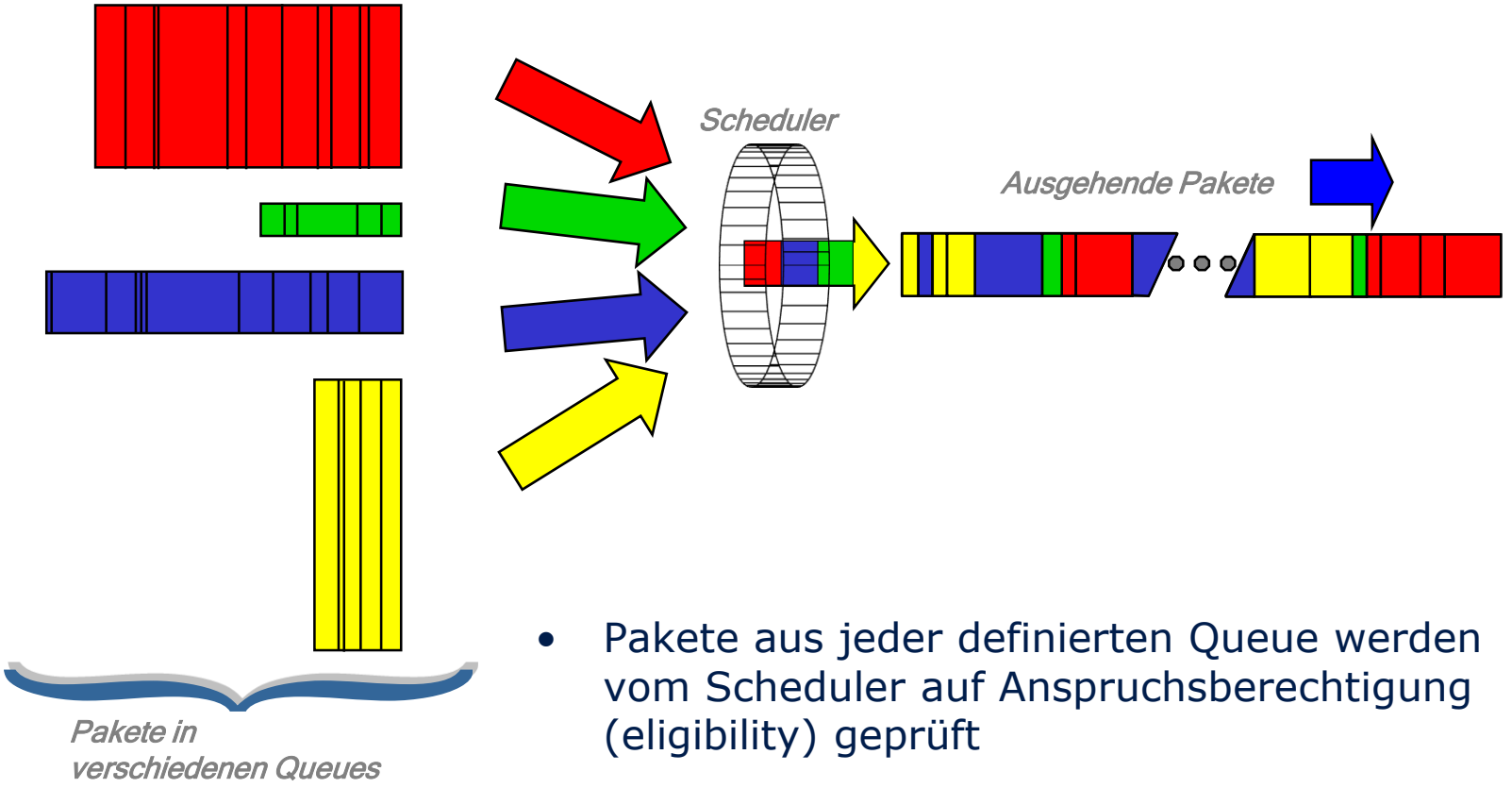
Algorithmen zur Priorisierung der Paketausgabe

Ziele

„fair" geteilte Bandbreite
garantierte Bandbreite und Verzögerungsparameter für jede QoS-Klasse
Jitter-Reduktion

- First-Come, First-Served (FCFS)
- Leacky Bucket
- Priority Queuing
- Round Robin (RR)
- General Processor Sharing (GPS)
- Virtual Clock (VC)
- Weighted Fair Queuing (WFQ, WF2Q, WF2Q+)
- Self-Clocked Fair Queuing (SCFQ)
- Stop-and-Go
- Rate Controlled Service Discipline (RCSD)

Packet Scheduling/Servicing

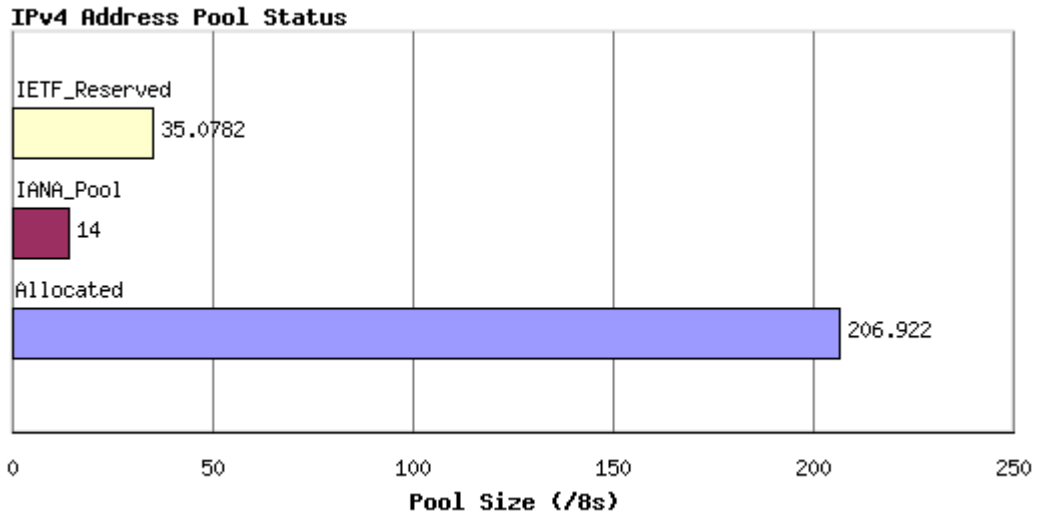


- Pakete aus jeder definierten Queue werden vom Scheduler auf Anspruchsberechtigung (eligibility) geprüft
- die am besten geeigneten Pakete werden vorgeplant und geliefert (scheduled)

IPv4 – Adreßpool -Prognose

(http://www.potaroo.net/tools/ipv4/index.html)

IANA 4.10.2010
noch 5% Reserve an IPv4-Adressen (/8-er Blöcke)



→ Adreßpool aufgebraucht

IANA 2.6.2011
RIR 27.1.2012

Migration IPv4 → IPv6

Schrittweiser Übergang

1. Implementierung in wichtigen Betriebssystemen ✓
2. isolierte IPv6-Netzwerke mit IPv4-Konvertierung zur Außenwelt ✓
3. DNS-Umstellung (Root-Server!) auf IPv4/IPv6-Doppelstack (✓)
4. Gemischter Betrieb (Kapselung von Paketen: **Tunnel**)

IPv6	→ IPv4 →	IPv6
IPv4	→ IPv6 →	IPv4
IPv4	→ … →	IPv6
IPv6	→ … →	IPv4

5. Einstellung der Unterstützung für IPv4-Netzwerke

IPv6

1996… IETF **RFC 2460** → IPv6 oder IPnG (IP Next Generation)
(diverse Modifikationen erschweren Akzeptanz)

Grund Adreßmangel, Unübersichtlichkeit von IPv4 (IP, IPsec, mobile IP, …)

Eigenschaften

- Verwendung von strukturierten **128-Bit-Adressen** (16 Byte)
  verschiedene Adreß-Typen, Rückwärtskompatibilität zu IPv4
- Flexibles Header-Format
  (40 Byte, bessere Notation von QOS-Parametern)
- Optionen zur
  Verschlüsselung, Authentisierung, Prüfung der Datenintegrität
- Autokonfiguration
- …

IPv6 – automatische Funktionen

RFC 2461 Neighbour Discovery Protocol

Adreßtyp „Solicited-Node Multicast“

Präfix	ff02:0:0:0:0:1:ff/104
+	letzte 3 byte der IPv6-Adresse

nützlich für

- automatische Adreßgenerierung (wie bei DHCP)
- Erkennung doppelter IP-Adressen im Netz
- Routersuche
- Dienstsuche im Netz

IPv6 - Paket

- **Basis-Header**
  - 4 bit Version
  - 8 bit Verkehrsklasse
  - 20 bit Datenstromkennzeichnung
  - 16 bit Nutzdatenlänge (opt. Header + Daten)
  - 8 bit Next-Header-Kodierung für Typ optionaler Header oder Kennzeichen für TCP bzw. UDP
  - 8 bit Hop Limit
  - 16 byte IPv6-Sendeadresse
  - 16 byte IPv6-Empfangsadresse
- evtl. optionale Header
- Daten

IPv6 – optionale Header

- Hop-by-Hop Option
  - 8 bit Next Header
  - 8 bit Länge
  - … Optionen,

    jeweils
    - 8 bit Typ
    - 8 bit Länge
    - … Parameter
- Destination Options — entspricht z.Zt. Hop-by-Hop-Header
- Fragmentation — Fragment Offset und -ID
- Routing — Übertragungsweg
- Authentication — Absender-Authentifizierung
- Encapsulation Security Payload — Sicherheitsparameter

IPv6 – Adreßtypen

Notation: Doppelbytes (4 Hexadezimalziffern)
Blöcke getrennt durch „:“, Abkürzungen erlaubt

Unicast-Adressen

- global gültige Adressen

  64 bit Netz-ID (3 bit `001`, 13 bit TLA, 32 bit NLA, 16 bit SLA)

  TLA + NLA: Top- bzw. Next Level Aggregator (Organisation)
  SLA: Site Level Aggregator (Subnetz)

  64 bit Host-ID (0xfffe + MAC-Adresse)

- eingebettete Adressen, z.B. für IPv4 Adressen

- Anycast Adressen

  Rechnergruppen-Adresse, Antwort vom nächsten Partner
  (formal wie Unicast Adressen)

IPv6 – Adreßtypen (2)

- spezielle Adressen, z.B.

für	Loop Back		0:0:0:0:0:0:0:1
	IPv4-Adressen	(eingebettet in IPv6)	0:0:0:0:0:ffff:a.b.c.d/96

- reservierte Adreßbereiche, z.B.

für	Global Unicast	(global gültig)	2000::/3
	Link Local Unicast	(verbindungseindeutig)	fe80::/10
	Site Local Unicast	(in Subnetzen eindeutig)	fec0::/10

Multicast Adressen

- Zieladressen für Rechnergruppen (RFC 4291, RFC 2375, …)

	Aufbau Präfix, … dauerhaft (well known), dynamisch, …	ff00::/8
z.B.	für alle Rechner eines Netzes	ff02::1
	alle Router eines Netzes	ff05::2