Digitale Kommunikation und Internetdienste 1

Wintersemester 2004/2005 – Teil 7

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

Übersicht

- Internet Protocol
  - IPv4 (cont'd)
- Subnetting, Supernetting
- Network Address Translation / Masquerading
- ICMP
- IGMP
- Internet Protocol
  - IPv6 (Geschichte)

Erinnerung

- IPv4-Adresse besteht aus 32 Bit
- Schreibweisen:
  - einzelne Bits: 10001100 10110011 11011100 11001000
  - einzelne Bytes dezimal: 140.179.220.200
  - einzelne Bytes hexadezimal: 8C B3 DC C8
- enthält sowohl Netzwerk- als auch Hostadresse
- Problem: wo trennen?

Erinnerung: Adreßklassen

- Einordnung nach Adreßklassen
- erstes Byte im Bereich 1 .. 126: Klasse A
- erstes Byte im Bereich 128 .. 191: Klasse B
- erstes Byte im Bereich 192 .. 223: Klasse C
- erstes Byte ist 0 und 127: reserviert (eigenes Netz, Loopback)
- Probleme: inflexibel, Adreßverschwendung

Subnetting (1)

- benötigt: Möglichkeit „passendere” Adreßgruppen zu anzusprechen
- und Untergruppen dieser Gruppen zu bezeichnen
- Lösung hierfür: *Subnetzbildung (subnetting)*
- Konvention wird in RFC 950 (STD-5) beschrieben
- Ermöglicht routing innerhalb von großen Teilnetzen

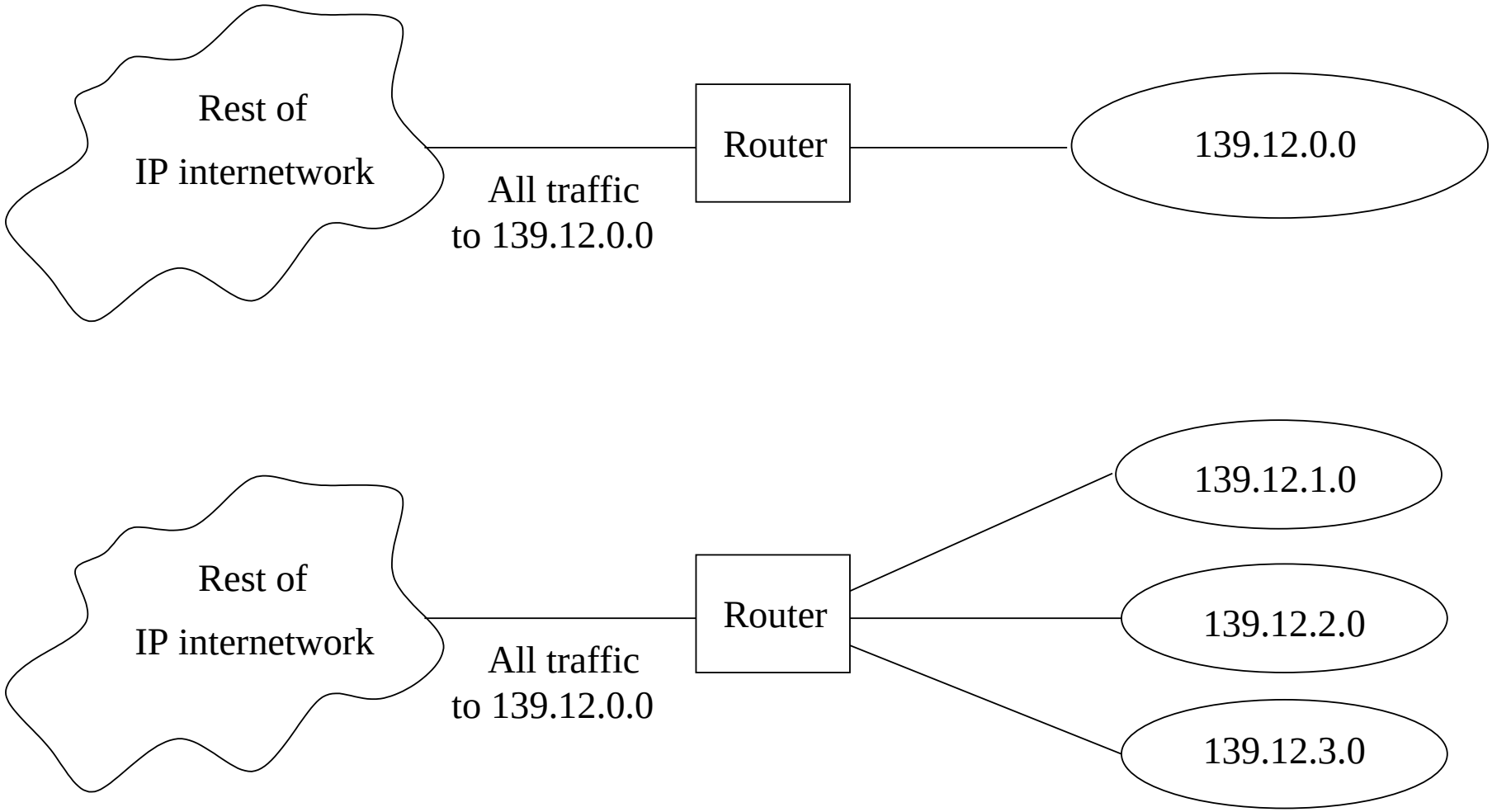
Subnetting (2)

- Host-Adreßteil einer IP-Adresse wird weiter unterteilt
- in eine Subnetznummer und
- in die eigentliche Host-Adresse
- Dadurch entstehen sogenannte *Subnetzwerke (subnets)*
- Beispiel: Aufteilung eines Klasse-B-Netzes

Subnetting-Beispiel (1)

- Beispiel: Klasse-B-Netz 139.12.0.0
- Subnetting: Die ersten 8 Bit der Host-Adresse werden für die neue Subnetz-Adresse verwendet
- es entstehen separate Subnetze mit eigener Subnetz-Adresse
- diese werden dem Router für 139.12.0.0 mitgeteilt
- dieser leitet IP-Pakete an das jeweilige Subnetz weiter

Subnetting-Beispiel (2)



Subnetting

- Aufteilung in Subnetze ist für Rest des Internets unsichtbar
- die Hosts der einzelnen Subnetze werden weiterhin von externen Hosts und Routern als Teile des Klasse-B-Netzes 139.12.0.0 betrachtet
- andere Router müssen daher nicht umkonfiguriert werden
- Wie kann der lokale Router aber wissen, wie das Netz 139.12.0.0 unterteilt ist?
- Bezeichnung durch *Subnetz-Maske (subnet mask)* (RFC 950)

Subnetzmaske (1)

- Die Subnetzmaske ist eine 32 Bit lange „Adresse”
- dient dazu Netzwerk- von Hostadresse zu unterscheiden
- die Bits der Subnetzmaske sind so definiert:
  - Alle Bits, die zur Netzwerk-ID gehören, haben den Wert 1
  - Alle Bits, die zur Host-ID gehören, haben den Wert 0
- Kurzschreibweise: Angabe der Anzahl von auf 1 gesetzten Bits, dem *Netzwerkpräfix (network prefix)*
- jeder Host benötigt eine Subnetzmaske: entweder eine *default*-Subnetzmaske oder eine *custom*-Subnetzmaske

Subnetzmaske (2)

Default Subnet-Masks:

- Class A: 11111111 00000000 00000000 00000000
  255.0.0.0
  Network Prefix /8

- Class B: 11111111 11111111 00000000 00000000
  255.255.0.0
  Network Prefix /16

- Class C: 11111111 11111111 11111111 00000000
  255.255.255.0
  Network Prefix /24

Subnetzmaske (3)

- Einfache Berechenbarkeit:
- um die Netzwerkadresse aus einer IP-Adresse mit bekannter Subnetzmaske zu bekommen
- wird ein logisches UND zwischen 32-Bit-IP-Adresse und 32-Bit-Subnetzmaske berechnet
- Beispiel: Adresse
  10001100.10110011.11110000.11001000 = 140.179.240.200
  Default Class B Subnet-Mask
  11111111.11111111.00000000.00000000 = 255.255.0.0
  (UND) ergibt Netzwerk-Adresse:
  10001100.10110011.00000000.00000000 = 140.179.0.0
  (kurz: 140.179.240.200/16)

Subnetz erstellen

- Subnetting ist eine Drei-Schritte-Prozedur:
- 1. Bestimmen der Anzahl Host-Bits für das subnetting
- 2. Aufzählen der neuen Subnetzwerk-IDs
- 3. Aufzählen der IP-Adressen für jede Subnetzwerk-ID

Subnetting-Beispiel (1)

Beispiel: Subnetting eines Klasse-C-Netzes

Subnet	#Masken-Bits	Subnetmask	kurz	#Hosts
0	0	255.255.255.0	oder /24	256
1-2	1	255.255.255.128	oder /25	126
3-4	2	255.255.255.192	oder /26	62
5-8	3	255.255.255.224	oder /27	30
9-16	4	255.255.255.240	oder /28	14
17-32	5	255.255.255.248	oder /29	6
33-64	6	255.255.255.252	oder /30	2

Subnetting-Beispiel (2)

Aufteilen des Klasse-C-Netzes 200.133.175.0 in 14 Subnetze mit je 14 Hosts

Subnetz	Subnetz-Bits	Netz-ID	Host-IPs	Broadcast-Adresse
1	0000	200.133.175.0	reserviert	keine
2	0001	200.133.175.16	.17 bis .30	200.133.175.31
3	0010	200.133.175.32	.33 bis .46	200.133.175.47
4	0011	200.133.175.48	.49 bis .62	200.133.175.63
5	0100	200.133.175.64	.65 bis .78	200.133.175.79
6	0101	200.133.175.80	.81 bis .94	200.133.175.95
7	0110	200.133.175.96	.97 bis .110	200.133.175.111
8	0111	200.133.175.112	.113 bis .126	200.133.175.127
15	1110	200.133.175.224	.225 bis .238	200.133.175.239
16	1111	200.133.175.240	reserviert	keine

Supernetting

- Prinzip des Subnetting mit IP-Adresse und Netzmaske nicht nur in Subnetzen verwendet
- „Supernetting”: Zusammenfassen mehrerer kleinerer Netze
- Anwendung: z.B. bei CIDR
- generell: IP-Adresse und Netzwerkmaske als Routing-Einträge
- d.h. Abkehr von klassenbasiertem Routing

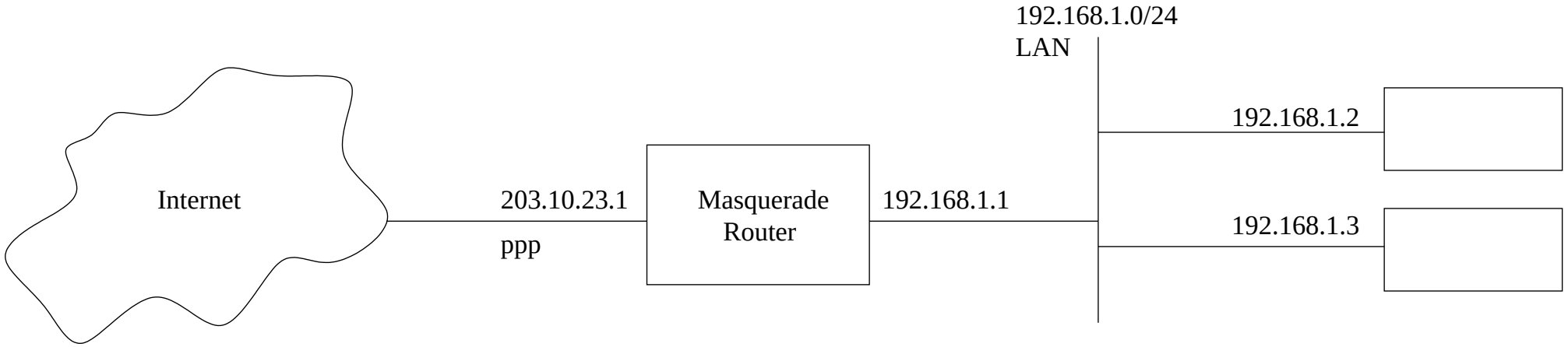
Private und öffentliche Netze

- Erinnerung: Es gibt öffentliche und private IP-Adreßbereiche
- 10.0.0.0/8, 172.16.0.0/16 - 172.31.0.0/16, 192.168.1.0/24 - 192.168.255.0/24
- private IP-Adreßbereiche werden im Internet nicht weitergeleitet
- Wunsch: mit privaten IP-Adreßbereichen dennoch aufs Internet zugreifen können
- Umgehung durch Umrechnung: Maskieren und Übersetzen

Masquerading

- *Maskieren (masquerading)* eines Netzbereichs vor einem anderen
- Maskieren: „verbergen und als etwas anderes ausgeben”
- kann von speziellen Routern durchgeführt werden
- nötig: mindestens eine öffentliche IP-Adresse

Masquerading-Beispiel



Masqueraded request
From 203.10.23.1 port 1085

Original request
From 192.168.1.3 port 1243

Translated by masquerade router at 203.10.23.1

Original reply
To 203.10.23.1 port 1085

Demasqueraded reply
To 192.168.1.3 port 1243

Masquerading-Probleme

Probleme:

- Knoten im privaten Netz können zwar eine Verbindung initiieren
- jedoch können Knoten von außerhalb dies nicht zu einen Knoten im privaten Netz
- entspricht nicht dem ursprünglichen Zweck
- „lügen” über Paketherkunft
- u.U. recht große Übersetzungstabelle die oft geändert werden muß
- braucht relativ viel Rechenzeit und führt zu höherer Latenz
- → entspricht daher „Notlösung”

Masquerading-Möglichkeiten

- Unterscheidung: *source NAT/destination NAT* (Umschreiben von Quell- bzw. Zieladresse)
- zudem verschiedene Stufen des Maskierens:
- *Basic Network Adress Translation (basic NAT)*: 1:1-Zuordnung
- *Masquerading (MASQ)*: n:1-Zuordnung
- *Network Adress Translation (NAT)*: n:m-Zuordnung
- *Port Adress Translation (PAT)*

Port

- *Port*: Bezeichnung für eine logische Kanalnummer
- Knoten können mehr als eine Verbindung gleichzeitig haben
- die Verbindungen werden dann nach Kanalnummer unterschieden
- Kanalnummern eindeutig je Transportprotokoll (z.B. TCP, UDP)
- häufige Schreibweise: 192.168.1.3:1023/tcp

Basic NAT

- Beispiel: Öffentliche verfügbare Adressen: 205.0.0.0/24

- Source NAT:
  Quelle 192.168.0.2, Ziel 170.0.0.1 → Quelle 205.0.0.2, Ziel 170.0.0.1
  Quelle 192.168.0.3, Ziel 170.0.0.1 → Quelle 205.0.0.3, Ziel 170.0.0.1

- Destination NAT:
  Quelle 170.0.0.1, Ziel 205.0.0.2 → Quelle 170.0.0.1, Ziel 192.168.0.2
  Quelle 170.0.0.1, Ziel 205.0.0.3 → Quelle 170.0.0.1, Ziel 192.168.0.3

Masquerading und PAT

- Source Adress Translation:
  Quelle 192.168.0.2:5000, Ziel 170.0.0.1:80
  → Quelle 205.0.0.2:6000, Ziel 170.0.0.1:80

  Quelle 192.168.0.3:5000, Ziel 170.0.0.1:80
  → Quelle 205.0.0.2:6001, Ziel 170.0.0.1:80

- Destination Adress Translation:
  Quelle 170.0.0.1:80, Ziel 205.0.0.2:6000
  → Quelle 170.0.0.1:80, Ziel 192.168.0.2:5000

  Quelle 170.0.0.1:80, Ziel 205.0.0.2:6001
  → Quelle 170.0.0.1:80, Ziel 192.168.0.3:5000

Port Forwarding

- *Portweiterleitung (port forwarding)* erlaubt es, Verbindungen über frei wählbare Ports zu Knoten innerhalb eines Netzes weiterzuleiten
- eingehende Datenpakete werden per Destination NAT maskiert
- ausgehende Pakete werden per Source NAT maskiert
- durch die Festlegung eines Ports keine dynamische Vergabe wie bei MASQ
- Verbindungen können auch von „außen” initiiert werden
- somit möglich, daß interner Rechner auch nach außen hin als Server dient

IPv4-Fragmentierung (1)

- Damit Datengramme über jede Art von Netzwerk verschickt werden können muß das Internet Protokoll dazu in der Lage sein, die Größe der Datengramme dem jeweiligen Netz anzupassen
- jedes Netzwerk besitzt eine sogenannte *maximale Paketgröße (Maximum Transfer Unit - MTU)*
- sie bezeichnet, daß nur Pakete bis zu dieser Größe über das Netz verschickt werden können
- z.B. X.25-Netz: 128 Byte
- z.B. Ethernet: 1500 Byte

IPv4-Fragmentierung (2)

- falls die MTU eines Übertragungsmediums kleiner ist als die Größe eines versendeten Pakets,
- so muß dieses Paket in kleinere Pakete aufgeteilt werden
- es genügt nicht die MTU des direkt angeschlossenen Netzes zu prüfen
- oder daß die Transportprotokolle einfach kleinere Pakete verschicken
- ein Paket kann auf dem Weg vom Quell- zum Zielhost mehrere unterschiedliche Netzwerke mit unterschiedlichen MTUs durchlaufen

IPv4-Fragmentierung (3)

- Beispiel: Ethernet-Heimnetzwerk ist über einen Router und DSL-Anschluß ans Internet angeschlossen
- Verbindung Router-DSL-Modem über PPPoE
- MTU Ethernet: 1500 Bytes
- Rechner im Netz senden mit MTU 1500
- DSL-Router fügt für PPPoE 8 Byte hinzu: 1508 Byte
- diese können nicht mehr am Stück zum DSL-Modem übertragen werden → Pakete müssen fragmentiert werden

IPv4-Fragmentierung (4)

- Wie kann man die MTU des gesamten Pfades herausfinden?
- Vorab: Berechnung, dann Test z.B. mit „ping”
- PPPoE: 8 Byte, läßt MTU 1492 übrig
- normaler IP- und ICMP-Overhead: 28 Byte → ping-Paketgröße: 1464 Byte „Nutzlast”
- ping mit Option „don’t fragment” testet den gesamten Pfad

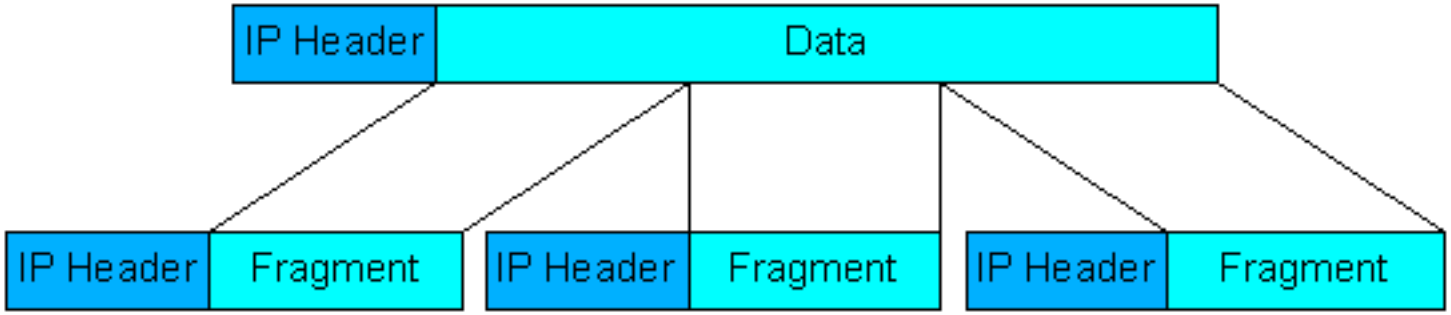
IPv4-Fragmentierung (5)

- benötigt: flexibleres Verfahren
- dieses muß bereits auf der Internet-Schicht kleiner Pakete erzeugen können
- das Verfahren wird *Fragmentierung* genannt
- das IP-Protokoll eines jeden Netzwerkknotens sollte in der Lage sein empfangene Pakete gegebenenfalls zu zerteilen
- jeder empfangende Knoten muß dazu in der Lage sein, diese Fragmente wieder zum ursprünglichen Paket zusammenzusetzen

IPv4-Fragmentierung (6)

- jedes Fragment eines zerteilten Pakets erhält einen eigenen, vollständigen IP-Header
- über das Identifikationsfeld im Header können alle Fragmente eines Pakets wiedererkannt werden
- die Lage der Daten eines Fragments innerhalb der Gesamtnachricht wird mit Hilfe des Fragment Offset-Feldes ermittelt
- einzelne Fragmente eines Pakets können durchaus unterschiedliche Wege auf dem Weg zum Zielhost nehmen

IPv4-Fragmentierung (7)



ICMP (1)

- das *Internet Control Message Protocol (ICMP)* ist Bestandteil jeder IP-Implementierung
- Aufgabe: Fehler- und Diagnoseinformationen für IP zu transportieren
- spezifiziert in RFC 792
- oft wird ICMP auch für Testzwecke verwendet
- z.B. um zu ermitteln, ob ein Host derzeit empfangsbereit ist

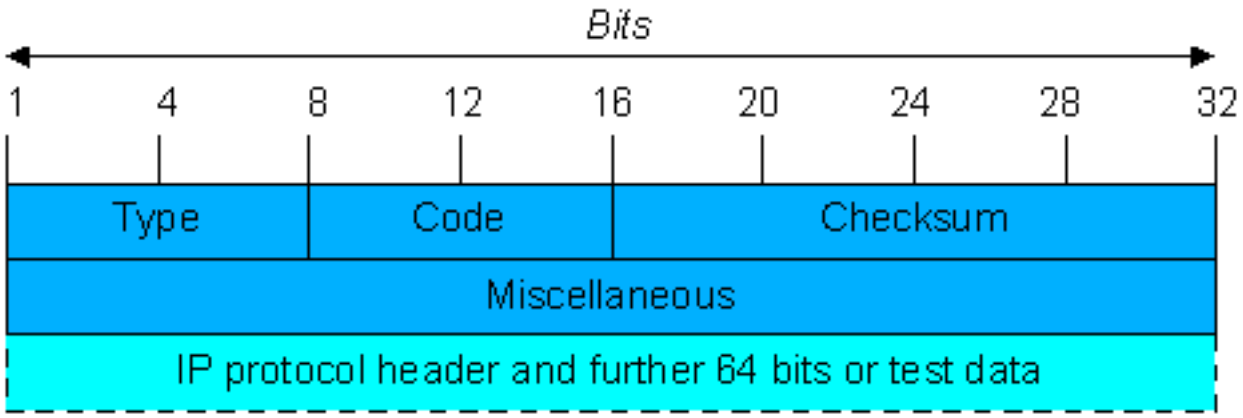
ICMP (2)

- ICMP ist ein vielseitiges Protokoll
- bietet dadurch auch die Möglichkeit versteckte Nachrichten zu übermitteln
- Stichwort: ICMP-Tunneling
- dabei wird das Datenfeld eines ICMP-Paketes genutzt, um Informationen zwischen Rechnern auszutauschen
- das ist zwar keine Technik, die das Einbrechen in Rechner ermöglicht
- dennoch kann ein Sicherheitskonzept eines Netzes dadurch unterlaufen werden
- siehe auch Zeitschrift c’t 11/1997

ICMP (3)

- ICMP hat sehr unterschiedliche Informationen zu transportieren
- nur der Grundaufbau des ICMP-Headers ist immer gleich
- die Bedeutung der einzelnen Felder im Protokollkopf wechselt jedoch
- jeder ICMP-Nachrichtentyp wird in einem IP-Datengramm eingekapselt

ICMP (4)


Bits
1
4
8
12
16
20
24
28
32
Type
Code
Checksum
Miscellaneous
IP protocol header and further 64 bits or test data

ICMP (5)

wichtige ICMP-Nachrichtentypen:

- *Destination Unreachable (Ziel nicht erreichbar)*
- diese Nachricht wird verwendet, wenn:
- ein Netzwerk, Host, Protokoll oder Port nicht erreichbar ist
- ein Paket nicht fragmentiert werden kann, weil das DF-Bit gesetz ist
- die Source Route Option nicht erfolgreich ist.

ICMP (6)

- *Source Quench (Quelle löschen)*
- wird ausgesendet, wenn ein Host zu viele Pakete verschickt
- die aus Kapazitätsmangel nicht mehr verarbeitet werden können
- der sendende Host muß dann die Rate zum Aussenden von Nachrichten verringern

ICMP (7)

- *Parameter Problem*
- verständigt den Absender eines Datengramms darüber,
- daß das Paket aufgrund einer fehlerhaften Angabe im IP-Header verworfen werden mußte
- *Redirect*
- wird ausgesendet, wenn ein Router feststellt, daß ein Paket falsch weitergeleitet wurde
- der sendende Host wird damit aufgefordert, die angegebene Route zu ändern

ICMP (8)

- *Time Exceeded (Zeit verstrichen)*
- wird an den Absender eines Datengramms gesendet, dessen Lebensdauer den Wert 0 erreicht hat
- diese Nachricht ist ein Zeichen dafür,
- daß Pakete in einem Zyklus wandern,
- daß Netz überlastet ist oder
- die Lebensdauer für das Paket zu gering eingestellt wurde

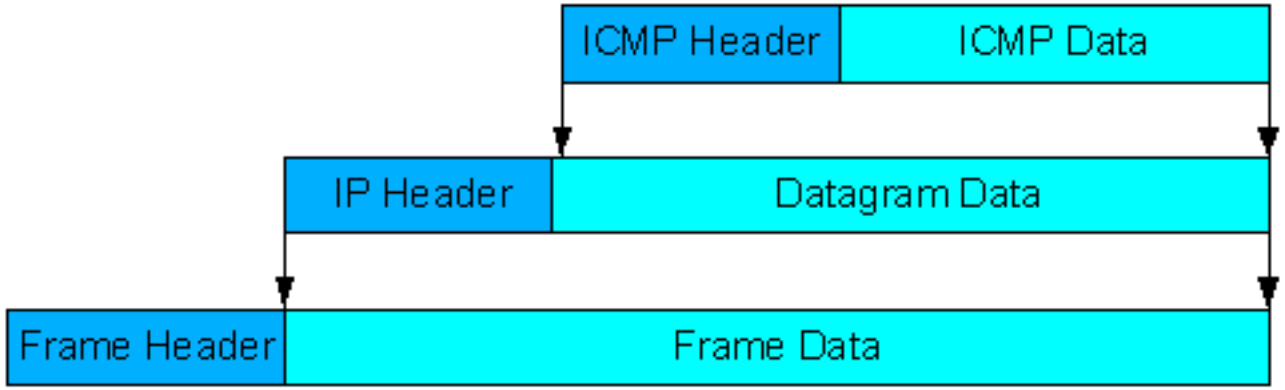
ICMP (9)

- *Echo Reply, Echo Request*
- damit kann festgestellt werden, ob ein bestimmtes Ziel erreichbar ist
- ein Echo Request wird an einen Host gesendet und hat einen Echo Reply zur Folge (falls der Host erreicht wird)
- *Timestamp Request, Timestamp Reply*
- diese Nachrichten sind ähnlich den zuvor beschriebenen Nachrichten
- außer daß die Ankunftszeit der Nachricht und die Sendezeit der Antwort mit erfaßt werden
- mit diesen Nachrichtentypen kann die Netzleistung gemessen werden

ICMP (10)

- IP verwendet ICMP zum versenden von Fehler- und Diagnosemeldungen
- während ICMP zur Übertragung seiner Nachrichten IP benutzt
- d.h. wenn eine ICMP-Nachricht verschickt werden muß, wird ein IP-Datengramm erzeugt
- die ICMP-Meldung wird in den Datenbereich des IP- Datengramms eingekapselt

ICMP (11)


ICMP Header
ICMP Data
IP Header
Datagram Data
Frame Header
Frame Data

ICMP (12)

- das Datengramm wird wie üblich versendet
- eine ICMP-Nachricht wird immer als Antwort auf ein Datengramm verschickt
- entweder ist ein Datengramm auf ein Problem gestoßen
- oder das Datengramm enthält eine ICMP-Anfrage, auf die eine Antwort versendet versendet werden muß
- in beiden Fällen sendet ein Host oder Router eine ICMP-Nachricht an die Quelle des Datengramms zurück

IGMP (1)

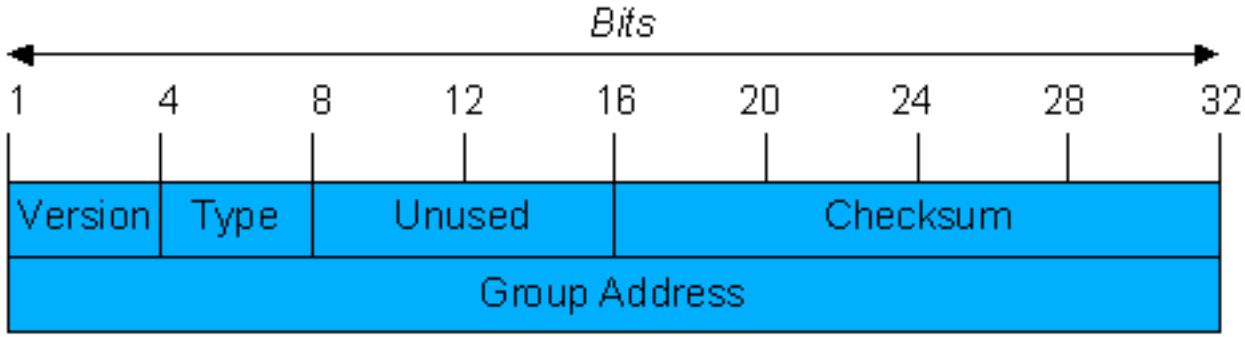
- *Internet Group Management Protocol (IGMP)*
- wird für Multicasting verwendet
- dafür sind IP-Adressen der Klasse D reserviert
- das erste Byte einer Multicast-Adresse hat den Wertebereich von 224 bis 239
- die Übermittlung der Nachricht erfolgt wie üblich nach bestem Bemühen
- d.h. ohne Garantie, daß die Daten auch tatsächlich alle Mitglieder einer Gruppe erreichen

IGMP (2)

- IGMP entspricht grob ICMP
- Unterschied: nur zwei Arten von Paketen
- Anfragen und Antworten
- Anfragen werden dazu verwendet, zu ermitteln welche Hosts Mitglieder einer Gruppe sind
- Antworten informieren darüber, zu welchen Gruppen ein Host gehört

IGMP (3)

- festes Paketformat
- wird in IP-Paket gekapselt



IGMP (4)

- *Version*:
- in RFC1112 ist die aktuelle Version 1 des IGMP Protokolls spezifiziert
- Version 0, die in RFC998 beschrieben wird, ist obsolet
- *Type*
- 1 = Host Membership Query (Anfrage)
- 2 = Host Membership Report (Antwort)

IGMP (5)

- *Checksum*
- der Algorithmus zur Berechnung der Checksumme entspricht dem des IP-Protokolls
- *Group Address*
- bei einer Anfrage zur Gruppenzugehörigkeit wird das Gruppenadressenfeld mit Nullen gefüllt
- ein Host, der eine Anfrage erhält, ignoriert dieses Feld
- bei einer IGMP-Antwort enthält das Gruppenadressenfeld die Adresse der Gruppe, zu der der sendende Host gehört

Internetentwicklung (1)

- bis vor einiger Zeit wurde das Internet größtenteils nur von Universitäten, Regierungsbehörden und einigen Firmen genutzt
- seit der Einführung des World Wide Web (WWW) ist das Internet aber auch zunehmend für Privatpersonen, kleinere Firmen etc. interessant
- das Internet wandelt sich von einem „Spielplatz für Akademiker” zu einem weltweiten Informations- und Unterhaltungssystem
- mit der ständig steigenden Anzahl von Benutzern des Internet werden sich auch die Anforderungen an das Netz ändern bzw. haben sich bereits geändert
- z.B. das angestrebte Zusammenwachsen der Computer-, Unterhaltungs- und Telekommunikationsbranchen

Internetentwicklung (2)

- den Anforderungen, die z.B. Video-on-demand stellt, ist das Internet bzw. das Internet Protokoll in der Version 4 nicht gewachsen
- Vinton Cerf (der 'Vater' des Internet) bezeichnet das Internet „(...) als die wichtigste Infrastruktur für alle Arten von Kommunikation"[1]
- „Am spannendsten finde ich es, die ganzen Haushaltsgeräte ans Netz anzuschließen. Ich denke dabei nicht nur daran, daß der Kühlschrank sich in Zukunft mit der Heizung austauscht, ob es in der Küche zu warm ist."
- . . .„natürlich muß die Sicherheit derartiger Systeme garantiert sein. Schließlich möchte ich nicht, daß die Nachbarkinder mein Haus programmieren"
- → völlig neue Anforderungen für Internetprotokolle

[1] c't 3/98 S. 44ff: Das Internet bleibt spannend!

- Internet Protokoll Version 6 – IPv6
- auch: IP Next Generation (IPnG)
- vorrangige Grund: begrenzter Adreßraum
- klassenlose Netzwerkeinteilung und CIDR schaffen zwar etwas Luft
- dennoch ist klar absehbar, daß auch diese Maßnahmen nicht ausreichen

IPv6-Ziele

- Die *IETF (Internet Engineering Task Force)* begann deshalb 1990 mit der Arbeit an einer neuen Version von IP

- Hauptziele:
  - Unterstützung von Milliarden von Hosts, auch bei ineffizienter Nutzung des Adreßraums
  - Reduzierung des Umfangs der Routing-Tabellen
  - Vereinfachung des Protokolls, damit Router Pakete schneller abwickeln können
  - Höhere Sicherheit (Authentifikation und Datenschutz) als das heutige IP
  - Mehr Gewicht auf Dienstarten, insbesondere für Echtzeitanwendungen
  - Unterstützung von Multicasting durch die Möglichkeit, den Umfang zu definieren
  - Möglichkeit für Hosts, ohne Adreßänderung auf Reise zu gehen
  - Möglichkeit für das Protokoll, sich zukünftig weiterzuentwickeln
  - Unterstützung der alten und neuen Protokolle in Koexistenz für Jahre

IPv6-Geschichte (1)

- im Dezember 1993 forderte die IETF mit RFC 1550 zu Vorschlägen auf
- es gab eine Vielzahl von Vorschlägen
- diese reichten von nur geringfügigen Änderungen am bestehenden IPv4
- bis zur vollständigen Ablösung durch ein neues Protokoll
- die drei besten Vorschläge wurden im *IEEE Network Magazine* veröffentlicht

IPv6-Geschichte (2)

- aus diesen Vorschlägen wurde von der IETF das *Simple Internet Protocol Plus (SIPP)* ausgewählt
- als Grundlage für die neue IP-Version
- es ist eine Kombination aus zweien der drei besten Vorschläge
- Arbeitsname wurde *IP - Next Generation (IPnG)*
- schließlich offizielle Versionsnummer: IP Version 6 oder kurz IPv6
- die Protokollnummer 5 (IPv5) wurde bereits für ein experimentelles Protokoll verwendet

Aussicht

Themenübersicht für die kommende Vorlesung:

- IPv6 (cont'd)
- Transportschicht
- UDP
- TCP

Ende Teil 7. Danke für die Aufmerksamkeit.