IP/DNS/DHCP-Versuch

1 Protokollstack

- Schicht 3

2 IP-Header

Bild

- Version: Protokollversion (aktuell = 4, neu = 6)
- Header length: Länge in 32-bit Einheiten (5 = normal = ohne Optionen = 20 Bytes)
- ToS: Priorität und Qualität (geringe Verzögerung, hoher Durchsatz, hohe Zuverlässigkeit). Kaum verwendet. Von neueren Standards umdefiniert (diffserv).
- total length: Länge des Datagramms in Oktetten.
- Identifikation: Fragmentierung für alle Fragmente gleich, sonst eindeutig. KEINE Sequenznummer.
- Flags: 1. Bit: 1 = nicht fragmentieren (falls erforderlich $\Longrightarrow$ verwerfen und ICMP-Fehlermeldung, 1. Bit: 0 = letztes Fragment, 3. Bit: nicht verwendet.
- Fragment Offset: Position des Fragments in original Datagramm (Einheit 8 Oktette). Man vermeidet Fragmentierung, weil sie Aufwand verursacht, indem man die maximale unfragmentierte Paketgröße ermittelt und verwendet.
- TTL: In jedem Router dekrementiert und bei 0 verworfen. (verhindert Endlosschleife)
- Protocol: 1 = ICMP, 6 = TCP, 17 = UDP (Protokoll in Data-Feld)
- checksum: nur für Header. In jedem Router neu berechnet (wegen TTL)
- Source IP Address: 4 Bytes
- Destination IP-Address
- Options: Durch Füllzeichen Header auf Vielfaches von 4 Bytes gebracht. Kaum verwendet (Sicherheit, Timestamp, Source Routing, Aufzeichung der Route)
- Data: Nutzdaten. Bis zu 65536 Oktette.

3 IP-Adressen

Bild

- Zweiteilung: Netzmaske — Hostanteil
- übliche Schreibweise: 4 Bytes dezimal durch „.“ getrennt.
- Klassen
  - A: 0-x, Defaultnetzmaske 8 Bit
  - B: 10-x, Defaultnetzmaske 16 Bit
  - C: 110-x, Defaultnetzmaske 24 Bit (192 = 11000000)
  - Multicast: 1110-x
- Classless Inter-Domain Routing (CIDR): Netzmaske wird zur Steigerung der Flexibilität im Umgang mit dem knappen Adreßraum individuell festgelegt (wir haben 28-Bit Netzmaske = 240)
- Maske manchmal auch als Breite in Bits angegeben
- besondere Adressen

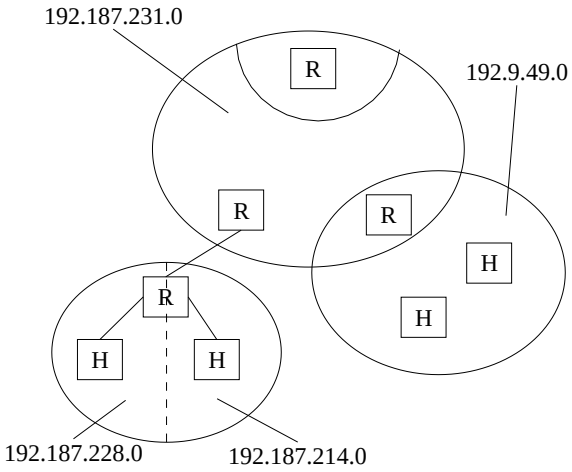
  - Netzadresse: Hostanteil = 0
  - direct Broadcast: binärer Hostanteil lauter 1er
  - limited Broadcast: 32 1er

- Beispielrechnung
  - Rechner: 192.168.215.5/255.255.255.240=28
  - Netzadresse: 192.168.215.0
  - direct Broadcast Adresse: 192.168.215.15
  - limited Broadcast Adresse: 255.255.255.255

4 Wegewahl - Routingtabelle

Host/Netz	Netzmaske	Gateway	Kommentar
127.0.0.0	255.0.0.0	0.0.0.0	localhost
192.168.215.0	255.255.255.240	192.168.215.3	eigenes Subnetz (direkt erreichbar)
default	0.0.0.0	192.168.215.14	Defaultroute

- Hostrouten und Netzrouten (ggf. extra Parameter bei route Kommando)
- Netzmasken immer angeben, sonst default Class A/B/C-Maske, die bei uns falsch ist.
- Funktion
  - Rechner will zu 192.168.215.9 senden
  - geht Tabelle durch und sucht Tabelleneintrag, bei dem Netzanteil (maskiert durch Netzmaske) gleich ist
  - gefunden $\Longrightarrow$ damit Gateway bekannt
  - nicht gefunden $\Longrightarrow$ Defaultroute (Interface zu GW $\Longrightarrow$ Routingtabelle)
- Problem: Woher weiß ein Router mit überlicherweise mehreren Interfaces, welches er nehmen muß $\Longrightarrow$ Bei Konfiguration des Interfaces gibt man (eine oder mehrere) IP-Adressen an
- nötige Konfiguration für Client
  - IP-Adresse
  - Netzmaske
  - Defaultgateway
  - $\Longrightarrow$ skaliert
- Problem: Router hat mehrere Anschlüsse an große Netze, deren Subnetze der Administrator nicht kennt
  - damit Default Route nutzlos, da vielleicht die falsche Richtung
  - manuelle Konfiguration nicht möglich $\Longrightarrow$ Routingprotokoll: Hello, RIP, OSPF,...
  - Prinzip: Router tauschen ihre Routingtabellen untereinander aus $\Longrightarrow$ Router lernen in welche Richung ein Subnetz zu finden ist



= Segement

= explizite Route (ggf + Kabel)

5 Rechnernamen — hosts-Datei

- Datei auf jedem Rechner: IP-Adresse — FQDN — Name
- Probleme
  - Alle Rechner im Internet in einer Datei ? Größe ! Suchaufwand !
  - Wer soll die Datei pflegen ? Aktualität ! Organisationsproblem ! Vertrauen ! Datenvolumen !
- Existenzberechtigung:
  - einige wichtige Rechner eintragen, damit Namensauflösung möglich, wenn DNS nicht geht.
  - Namensauflösung während Booten (vereinfacht Konfiguration)
  - Kleine Netze, für die sich DNS-Server nicht lohnt.

6 Rechnernamen — DNS

- Hierarchischer Verzeichnisdienst
  - „Domain“ = durch „.“ getrennte „Labels“: www.informatik.uni-muenchen.de
  - Top-Level-Domains: org, net, com, edu, gov, mil, int, de (Ländercodes)
  - Grund: Zuständigkeit kann für Teilbereiche an verschiedenen Stellen abgegeben werden. (z.B. de $\Longrightarrow$ DE-NIC)
- verteilter Verzeichnisdienst
  - mehrere Server für Root-Domäne (mit DNS-Server ausgeliefert)
  - ein Server kann mehrere Domänen bedienen (z.B. Provider für alle seine Kunden-Domänen)
  - Zone: alle Domänen, für die ein Nameserver zuständig ist.
  - Sinn: Skalierbarkeit, Robustheit, Geschwindigkeit
  - Verkettung: Root-Server-Adressen mit DNS-Server ausgeliefert
  - Redundanz: Master und Slaves
    - Master hat Daten für Namensauflösung

        * Slave holt sich Daten in regelmäßigen Abständen vom Master
        * Wenn Master ausfällt können Slaves weiter Anfragen beantworten
- Namensauflösung
    - 2 Arten
        * rekursiv: Server fragen rekursiv weitere Server, bis Ergebnis bekannt
        * iterativ: Server beantwortet die Anfrage, falls Information lokal verfügbar, sonst Referenz auf Server, der mehr weiß.
    - Auflösung unbekannter Namen über Root-Server
    - Kann auch anderen Nameserver kennen (forwarding): z.B. wenn Root-Server nicht direkt erreichbar.
- Caching
    - Grund: erhebliche Geschwindigkeitssteigerung
    - Jede Antwort enthält Zeitintervall, in dem ein Cache-Eintrag verwendet werden darf
    - Probleme
        * Inkonsitenz der Datenbasis
        * Fehler läßt sich nicht schnell wieder korrigieren
        * Sicherheitsproblem gefälschte Antworten mit falschen Querreferenzen
        * keine Echtheitszertifikate im praktischen Einsatz
- Server-Konfiguration
    - mehrere Root-Nameserver
    - ggf. Forward-Adresse
    - eigene Domänen
    - Dateien mit Names-/Adreßpaaren für beide Richtungen
        * Name → IP-Adresse: Standardfall
        * IP-Adresse → Name: spezielle Domäne $\Longrightarrow$ <invertierte IP-Adresse>.in-addr.arpa
- Client-Konfiguration
    - Adresse des lokalen Nameservers
    - Besser mehrere, weil DNS sehr wichtig

7 DHCP — Dynamic Host Configuration Protocol

- BOOTstrap Protocol (BOOTP) ist Voräufer von DHCP
- Dynamische Konfiguration der Clients: IP-Adresse, Netzmaske, Router, Nameserver, Root-FS, Boot-Dateiname,...
- Erkennungsmerkmal eines Clients: MAC-Adresse
- Zusätzliche Fähigkeiten von DHCP: dynamische Adreßzuordnung aus Adreßpool
- Arbeitet mit UPD
    - Frage: Ohne IP-Adresse??? Antwort: IP-Broadcast
    - Rückweg:
        * Broadcast: ok
        * direkt
            · Problem: ARP-Request des Servers kann von Client nicht beantwortet werden (kennt ja seine IP-Adresse nicht!)
            · Lösung: DHCP-Server holt MAC-Adresse aus empfangenem Paket und erweitert ARP-Cache entsprechend (geht nur, wenn vom Betriebssystem ermöglicht). Client erkennt Paket an der MAC-Adresse.