1. Az OSI referenciamodell és a TCP/IP

Az ISO/OSI modell

- A számítógép hálózatok - a megvalósításuk bonyolultsága miatt - rétegekre osztódnak.
- Több világcég megalkotta a saját elképzelései alapján a saját hálózati architektúráját, de az eltérések miatt egységesíteni kellett, amit csak nemzetközi szinten lehetett megoldani. Ez a szerep az ISO-ra (International Standards Organization -Nemzetközi Szabványügyi Szervezet) hárult.
- A hálózatokra vonatkozó rétegmodellt 1980-ban fogalmazta meg OSI (Open System Interconnection) néven. Ez viszont nem szabvány, hanem csak egy ajánlás. Mindössze csak azt mondja meg, hogy milyen rétegekre kell osztani egy hálózatot és ezen rétegeknek mi legyen a feladatuk. Nem kötelező betartani. A megvalósított rendszerekben egyes rétegei szinte teljesen üresek, másokat tovább kellett osztani zsúfoltságuk miatt. Sok hiányossága ellenére a mai napig alapnak tekintik a gyártók.
- a TCP/IP megjelenéséig csak kevés, gyártófüggő megvalósítás, pl.:
  - IBM: SNA
  - Digital: Decnet
  - Siemens: Transdata
- Az OSI referencia modell szerint egy hálózatot 7 rétegre osztunk:

Alkalmazási réteg
Megjelenítési réteg
Viszonyréteg
Szállítási réteg
Hálózati réteg
Adatkapcsolati réteg
Fizikai réteg

- egy réteg a többivel *protokollok* segítségével kommunikál (-> *protokoll stack*), így a rétegek egyenként változtathatók (pl. a fizikai rétegben a sodrott érpárt üvegszálra cseréljük)
- egy réteg az alatta levő szolgáltatásait veszi igénybe, s a felette levőnek szolgáltatást nyújt

Az adatátvitellel foglalkozó rétegek:

- A fizikai réteg (physical layer)
  A bitek kommunikációs csatornára való kibocsátásáért felelős. Biztosítania kell, hogy az adó által küldött jeleket a vevő is azonosként értelmezze. Tipikus villamosmérnöki feladat a tervezése.

- Az adatkapcsolati réteg (data link layer)
  Alapvető feladata a hibamentes átvitel biztosítása a szomszéd gépek között, vagyis a hibás, zavart, tetszőlegesen kezdetleges átviteli vonalat hibamentessé transzformálja az összeköttetés fennállása alatt. Az adatokat adatkeretekké (data frame) tördeli, továbbítja, a nyugtát fogadja, hibajavítást és forgalomszabályozást végez.
- A hálózati réteg (network layer)
  A kommunikációs alhálózatok működését vezérli, feladata az útvonalválasztás forrás és célállomás között. Ha az útvonalban eltérő hálózatok vannak, akkor fregmentálást, protokoll átalakítást is végez. Az utolsó olyan réteg, amely ismeri a hálózat topológiáját.
- A szállítási réteg (transport layer)
  Feladata a végpontok közötti hibamentes adatátvitel biztosítása. Már nem ismeri a topológiát, csak a két végpontban van rá szükség. Feladata az összeköttetések felépítése, bontása,

csomagok sorrendbe állítása.

A logikai összeköttetéssel foglalkozó rétegek:

- A viszonyréteg (session layer)

Lehetővé teszi, hogy két számítógép felhasználói kapcsolatot létesítsenek egymással. Jellegzetes feladata a logikai kapcsolat felépítése és bontása, párbeszéd szervezése. Szinkronizációs feladatokat is ellát, ellenőrzési pontok beépítésével.

- A megjelenítési réteg (presentation layer)

Az egyetlen olyan réteg, amely megváltoztathatja az üzenet tartalmát. Tömörít, rejtjelez (adatvédelem és adatbiztonság miatt), kódcserét (pl.: ASCII - EBCDIC) végez el.

- Az alkalmazási réteg (application layer)

Széles körben igényelt szolgáltatásokat tartalmaz. Pl.: fájlok gépek közötti másolása.

A TCP/IP

- nyílt rendszer (szabadon hozzáférhető és implementálható szabványok)
- etalon implementáció: BSD (Berkeley Software Distribution) Unix
- A TCP/IP által használt rétegek és a hozzájuk tartozó protokollok:

SMTP	HTTP	FTP	DNS	Alkalmazási
TCP			UDP	Szállítási
IP				Hálózati
Ethernet	PPP	AX.25		Adatkapcsolati
réz, üveg	Telefon	Rádió		Fizikai

Fogalmak:

- unicast/multicast/broadcast – egy címzett/sok címzett/mindenkinek szól
- simplex/half duplex/full duplex – egyirányú/váltott kétirányú/egyidejű kétirányú kommunikáció
- CRC (Cyclic Redundancy Check) – hibaellenőrző eljárás ill. annak eredménye
- flow control (folyamatvezérlés) – a fogadó szabályozza a bejövő adatfolyam ütemét-
- little/big endian – egy bájt bitjei közül a kisebb/nagyobb helyiértékű az első a vezetékben (pl. az Ethernet CRC és IP csomagok big endian sorrendűek)
- PDU (Protocol Data Unit) – egy kommunikációs protokoll által értelmezett adategység
- SDU (Service Data Unit) – egy réteg ilyen adategységet küld a fölötte levő rétegnek

2. RFC-k, Internet-szervezetek

RFC-k

- **R**equest **F**or **C**omment
- az internetközösség alapdokumentumainak (szabványok, protokollok, javaslatok) általános megnevezése
- nyilvántartójuk az IETF
- egy munkacsoport egy bizonyos probléma megoldására keletkezik és a probléma megoldásával feloszlik

Internet-szervezetek

- IEEE (Institute of Electrical and Electronics Engineers): az Internetes, számítógépes és elektronikai szabványokért felelős szervezet
- IETF (Internet Engineering Task Force): a legfőbb Internetes szabványügyi szervezet, minden érdeklődő számára nyílt.
- ISOC (Internet Society): egy hivatásosokból álló non-profit tagsági szervezet, amely az Internetes technológiák fejlődését próbálja elősegíteni, valamint oktató, népszerűsítő és ösztönző szerepet is vállal az Internetes technológia és alkalmazások terén. A tagság ingyenes. Magyar képviselete a MIT (Magyar Internet Társaság).
- IANA (Internet Assigned Numbers Authority): a különböző Internetes paraméterek, számok és nevek (portok, egyedi IP-címek, domain nevek stb.) kiosztásáért és protokollok paramétereinek számontartásáért felelős szervezet.
- ISC (Internet Systems Consortium): egy non-profit közhasznú szervezet, amely alapvető protokollok, szoftverek és műveletek kifejlesztésével és karbantartásával járul hozzá az Internet infrastruktúrájához.
- stb.

3. Klasszikus ethernet

- A Xerox fejlesztése a 70-es évekből, azóta rengeteg újítása van
- Az OSI-modell két legalsó szintjét valósítja meg
- Eredetileg busztopológia (ma csillag és fa)

Működés

- az adatátvitel keretekben (frame) történik
- egy keret felépítése:

Preambulum (7-byte)	Start Frame Delimiter (1-byte)	Cél MAC cím (6-byte)	Forrás MAC cím (6-byte)	Hossz (2-byte)	**Adat**	Pad (tömítés, ha kell)	**CRC** (4-byte)

  - Preambulum: szinkronizálásra való, váltakozó 0/1 sorozat
  - SFD: „10101011”
  - Cél:
    - lehet *unicast* (állomásnak), *multicast* (tartománynak) vagy *broadcast* (mindenkinek: „FFFFFFFFFFFF”)
    - MAC (Media Access Control) cím:
      - az első 3 bájt gyártónként fix (az IEEE osztja), a maradék (2^{24} bit) a gyártóé
      - elsősorban a küldő és a címzett eszköz azonosítására való
  - Hossz: következő bájttól a CRC-ig (az újabb Ethernet II-ben típus van helyette)
  - Pad: az adat minimális mérete 46 bájt, ha nincs annyi, kiegészítjük
  - CRC: célcímtől az adat (pad) végéig számolandó; ha a fogadónak más jön ki, eldobja a keretet
- Tulajdonságok:
- CS (Carrier Sense)
  - az egyes állomások érzik, ha a csatorna foglalt
- MA (Multiple Access)
  - a másnak szóló csomagokat csak promiscous módban fogadják (snoop - szimatolás)
- CD (Collision Detection)
  - ha adás közben észreveszik, hogy ütközés (collision) van, megállnak, de előbb még 32 zajbitet kikürtölnek
  - véletlen ideig várnak: *backoff delay*, majd újra próbálkoznak
  - ha újra ütközés van, újra véletlen, de hosszabb ideig várnak: *exponential backoff* egyetlen állomás kisajátíthatja az erőforrásokat (capture effect)
- 1 persistent
  - ha szabad a csatorna, és van adnivaló, 1 valószínűséggel adnak

Fizikai réteg

- eredetileg 10Base5 vastag koax kábel, később 10Base2 vékony koax T alakú BNC csatlakozókkal
- (újabb ethernet verziók csavart érpárokat, optikai kábelt ill. wireless kapcsolatot használnak)

4. Point to point ethernet, fast ethernet, full duplex ethernet, VLAN-ok

Point to point ethernet

SLIP:(**Serial Line IP) = (Soros vonaLi IP)**

Egy protokoll, melyet az IP soros vonalon (mint pl. telefon vonal, RS-232-es kábel) keresztüli használatára készítettek, két rendszer összeköttetésére. A SLIP meghatározása az 1055-ös RFC-ben található.

Pontól-Pontig-Protokoll. A PPP, a SLIP utódja, a routertől-routerig és hoszttól-hálózatig kapcsolat megteremtésére alkalmas szinkron és asszinkron vonalakon egyaránt. 3 részből áll:
- lehetővé teszi IP-csomagok keretté alakítását és soros vonalon való továbbítását
- tartalmaz egy kapcsolat vezérlő protokollt (link controll protocoll, LCP), ami az adatkapcsolat létrehozására, konfigurálására és ellenőrzésére szolgál.
- tartalmaz hálózatvezérlő protokollt (network controll protocoll, NCP), amely például lehetővé teszi egy állomás számára, hogy megadja, képes-e fejrésztömörítésre. (Ez azért lehet hasznos, mert a TCP és az IP protokollok elég nagy mennyiségű fejrész jellegű adattal bővítik az adatokat.)

flag 7E	cím FF	vez. 03	protokoll	információ	CRC	flag 7E
1	1	1	2	max 1500 byte	2	1

Fast ethernet

- IEEE 802.3u, 1995
- 100 Mbit/s

Full duplex ethernet

- IEEE 802.3x
- Csak pont-pont összeköttetéseken működik
    - Például switch <-> végberendezés
    - Nem jelenti, hogy csak két ethernet cím!
- Nincs CSMA/CD, bármikor lehet adni
- Az átviteli kapacitás duplájára nő
    - Például 100 Mb/s összeköttetésen 200Mb/s forgalom lehetséges
- Nem kell ütközések miatt késleltetni
- Kábel hossz nőhet, slot time csökkenhet

Ethernet flow control

- A full duplex ethernet kellemetelen mellékhatása miatt kell
- Az adó állomást meg kell állítani, ha nem tudjuk tovább adni a frame-eket
- PAUSE frame-et küldhetünk
    - Speciális frame típus
    - A destination cím lehet az adó állomás ethernet címe, vagy a speciális 01:80:C2:00:00:01 multicast cím
    - A type mező: 0x8808
    - Adat:
        - MAC opcode: 0x0001
        - MAC paraméter: 2 byte idő, az egysége 512 bitnyi
        - 42 db 0 byte (pad, hogy a 46 byte meglegyen)
    - Egy újabb PAUSE az előzőt érvényteleníti
        - 0-val érkező PAUSE hatása: continue
    - A két állomás egymástól függetlenül lehet képes adni/fogadni PAUSE-t
    - PAUSE alatt a másik állomás csal MAC control frame-et küldhet (pl. pause-t)

Gigabit ethernet

- IEEE 802.1z - optika, IEEE 802.1ab UTP
- 1998
- Extension field
    - Gigabit ethernetnél a time slot 512 bit
    - Ezt úgy töltjük ki, hogy a CRC *után* kiegészítjük a frame-et
    - Csak half duplex módban van értelme
- Burst (ömlesztett) mód
    - Több frame-et küldünk egy lélegzetre
    - Az első frame-ben jelezzük, hogy burst következik
    - Frame-ek közt gap-eket
    - A burst maximális hossza 8192 byte
    - Csak half duplex módban van értelme

Frame, jelzi a burst-öt	Gap	Frame	Gap	...	Frame

VLAN-ok

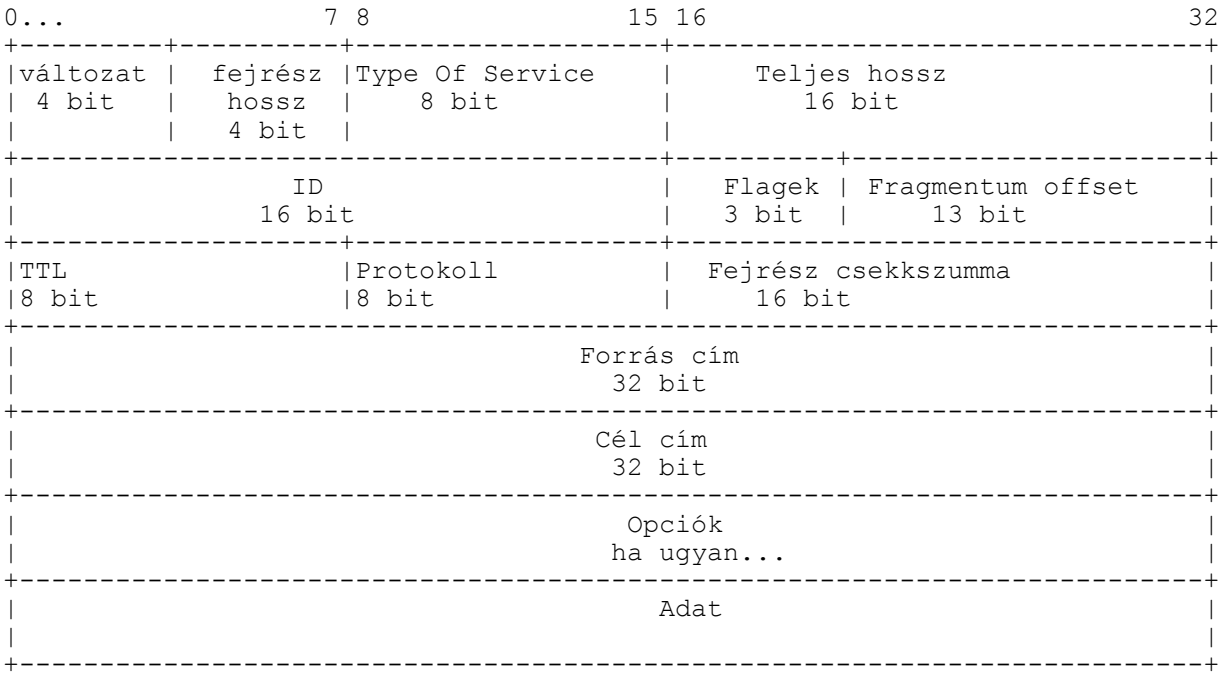
- IEEE 802.1Q
- Broadcast domain-eket = VLAN definiálhatunk
- Egy fizikai ethernetet logikailag több részre osztunk
- Csak valamilyen magasabb réteg (router) által lehet kommunikálni a VLAN-ok közt
- Előnyök
    - Biztonság
    - Kezelhetőség
    - Erőforrás takarékosság
- VLAN tag-ek: az ethernet frame-et újra ethernetbe csomagoljuk
    - A cím mezők után 4 extra byte
    - 2 byte: 802.1Q típus
    - 12 bit: VLAN id

5. IP

RFC791 Postel, 1981

Tulajdonságok

- Nem megbízható
  - Nem biztos, hogy a célba ér a csomag
  - Nem biztos, hogy nem érkezik többször
  - Nem biztos, hogy egymás utáni csomagok sorrendje megmarad
  - Nem biztos, hogy nem sérülnek benne bitek
- Nem kapcsolatorientált
  - Az egyes csomagokat a hálózati réteg egymástól függetlenül kezeli
  - Nincs kapcsolat felépítés, kapcsolat bontás
- A felsőbb rétegek gondoskod(hat)nak megbízható, kapcsolat-orientált kommunikációról



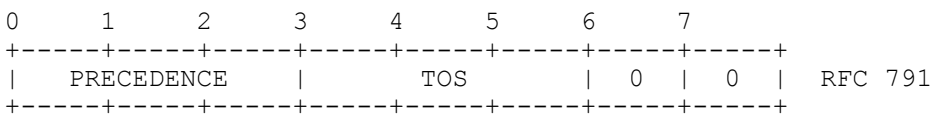
- A 0. bit a legnagyobb helyiértékű (big endian)
- Az egyes sorokban álló byte-ok balról jobbra mennek át (big endian)
- A változat (version) klasszikus esetben 4, manapság lehet 6

Fejrész hossz

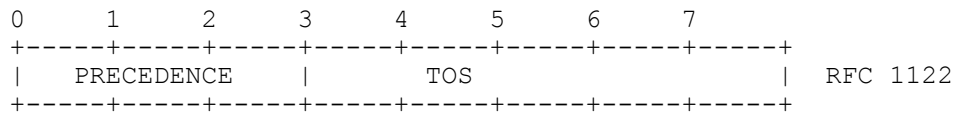
- 4 bit, a mértékegység 4 byte.

TOS

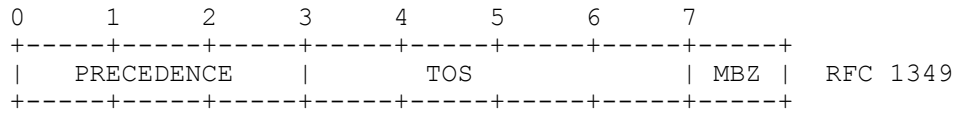
- RFC 791



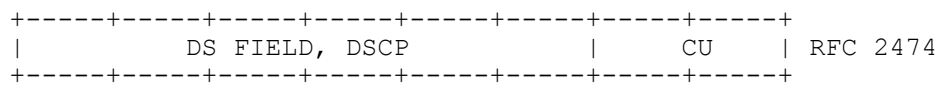
- RFC1122



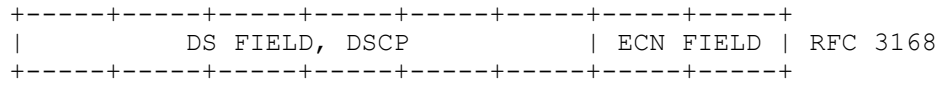
- RFC 1349



  - MBZ: Must Be Zero
- RFC 2474



  - CU: Currently Unused
- RFC 3168



    - DSCP: differentiated services codepoint
    - ECN: Explicit Congestion Notification
- A csomag hálózat belsejében való továbbítását befolyásoló mező
- Sokszor átdolgozták
- Nincs garancia arra, hogy a hálózati partnerek kezelik
    - Többnyire nincs baj abból, ha nem
- Hálózati átjárók, tűzfalak. routerek manipulálhatják

Hossz

- Az egész IP csomag hossza byte-ban
- Max 65535
- Egy interfészen az MTU korlátozza
- Közbülső eszközök fragmentálhatnak
- Ethernetnél mutathat kevesebbet mint az ethernet csomag
    - Az ethernet csomag legalább 46 byte

ID

- A csomagra jellemző egyedi szám
- Hagyományosan eggyel több mint a előző
    - Megjósolható -> visszaélésre ad módot
- Tűzfalak randomizálhatják

Flag-ek

- Az IP darabolásnál (fragmentálás) van szerepük
    - Don't Fragment (DF)
    - More Fragments (MF)

Fragmentum offszet

- Ha az MF bit áll, azt mutatja, hogy *ez* a csomag hova illik
- Mértékegység: 8 byte

TTL - Time To Live

- Az időt hop-ban méri
- Felső korlátot ad közbülső routerekre
- Minden router dekrementálja
- Ha 0, eldobja a csomagot, ICMP üzenetet küld vissza a feladónak
- Ha nem lenne végtele ideig keringhetnének csomagok
- A traceroute program épp ezt használja
    - Egy sikertelen traceroute: tracroute-fennakad.captracroute-fennakad.cap
    - Egy sikeres (tcptraceroute): tcptraceroute.cap

Protocol

- Az IP feletti protokollt adja meg
- IANA (Internet Assigned Numbers Authority) osztja
    - ICMP -1
    - TCP - 6
    - UDP - 17
    - Összesen: IP-protocol-numbers.txt

Csekszumma

- Nem ethernet CRC, sima bit összeadás
    - RFC1071, RFC1624 kiegészíti
    - ones' complement összeadás, a csekkszumma az eredmény egyes komplemense
    - Az ellenőrző oldalon 0 jön ki, ha minden jó
- Csak a fejrészre
- A csomag többi részét a felsőbb szint ellenőrizheti
- Hop-ról hop-ra változik. Miért?

Forrás és célcím

- Az IP réteg szempontjából a legfontosabb: ennek alapján továbbítja (route-olja) a csomagot
- Tűzfalak még ezt is módosíthatják

Opciók

- Type (1 byte), Hossz (1 byte), Adat
- Elvben több is lehet
- A mindennapi gyakorlatban ritka
- Pl: source routing
    - Manapság tiltják a routerek, tűzfalak

6. PPP, PPPoE

PPP - Point to Point Protocol

- RFC1661: pont-pont kapcsolatra használatos adatkapcsolati réteg
- IP alatt gyakran használt, másra is használható
- Leggyakrabban telefon vonalon
- LCP: Link Control Protocol
    - A kapcsolat felépítésére
    - A kapcsolat konfigurálására
    - A kapcsolat vezérlésére
- NCP: Network Control Protocol
    - A magasabb réteg(ek) konfigurálására

PPP frame szerkezet - ISO HDLC (High level Data Link Control) alapú

Flag (1 byte)	Address (1 byte)	Control (1 byte)	Protokoll (2 byte)	Adat	CRC (2 byte)	Flag (1 byte)

- Flag: a frame kezdetét és végét jelzi: 0x7E
- Address: PPP-nél mindig 0xFF
- Control: PPP-nél mindig 0x03
- Protocol: ethernet „type"-hoz hasonló
- Adat
    - Általában legfeljebb 1500 byte. LCP-vel másban is megállapodhatnak
    - Az adatfolyamban a flag byte-ot escape-elni kell
    - Alapértelmezésben minden 0x20-nál kisebb byte-ot is escape-elnek

LCP

- Kapcsolat felépítés
- Autentikálás
- Paraméterek egyeztetése
- Tömörítés
- Sallangok elhagyása
- Kapcsolat figyelés

NCP

- A magasabb réteg konfigurálására szolgál
- IP: IPCP, IP Control Protocol, RFC 1332
    - IP cím
    - Van Jacobson TCP/IP fejrész tömörítés
        - A rendszerint 40 byte-os TCP/IP fejrészből 3 lehet
    - DNS szerverek címe (RFC1877)

PPPoE

Ver. (4 bit)	Típus (4 bit)	Kód (1 byte)	Session ID (2 byte)	Hossz (2 byte)	Adat

- PPP over Ethernet, RFC2516
- Ethernet type mező:
    - 0x8863: discovery stage

        - 0x8864: PPP session stage
- Discovery
    - A partnerek megtudják az ethernet címeket és egy session ID-t
    - Négy lépcsős folyamat
        - Broadcast ethernet csomaggal indul (PADI, Initiation)
        - Több állomás (Access Concentrator) is válaszolhat (PADO, Offer)
        - A kliens egy unicast csomaggal dönt (PADR, Request)
        - A koncentrátor egy session id-t küld (PADS, Session-confirmation)
- Session
    - Unicast csomagok, mindben ott a session id, hossz
    - PADT (Terminate) lezárhatja

7. ARP, RARP

Hogyan tudja meg a link layer (ethernet) címet egy hálózati állomás?

- ARP - Address Resolution Protocol
- RFC826
- Elsősorban etherneten

ARP csomag formátum

6	6	2	2	2	1	1	2	6	4	6	4
Ethernet cél	Ethernet forrás	típus	hardware típus	protokoll típus	hardware méret	protokoll méret	Opkód	Ethernet forrás	IP forrás	Ethernet cél	IP cél

- Az ethernet type: 0x806
- A kérés broadcast, a válasz unicast
- Hardware type: ethernetnél 1
- Protocol type: IP-nél 0x800
- Hardware size: 6 (ethernet cím hossza)
- Protocol size: 4 (IP cím hossza)
- Opcode: 1 ha kérés, 2 ha válasz, 3 ha RARP request, 4 ha RARP válasz

```
22:25:58.477643 0:b:cd:8:24:22 Broadcast arp 42: arp who-has 10.1.30.1 tell
10.1.30.200
22:25:58.477818 0:7:e9:2e:74:99 0:b:cd:8:24:22 arp 60: arp reply 10.1.30.1
is-at 0:7:e9:2e:74:99
```

ARP kérés nem létező hostra

- A broadcast-ra nem jön válasz
- A kérdező timeout után újra próbálkozik
    - Implementáció függő
- Egy idő után feladja, értesíti a felsőbb réteget a hibáról
    - Implementáció függő

ARP cache

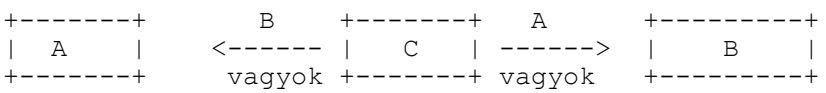
- A megtanult megfeleltetéseket az állomások megjegyzik
- Timout: néhány perc
    - Implementáció függő
- Erővel is lehet ARP entryket betenni, kivenni
    - Unixokon arp parancs

ARP cache poisoning

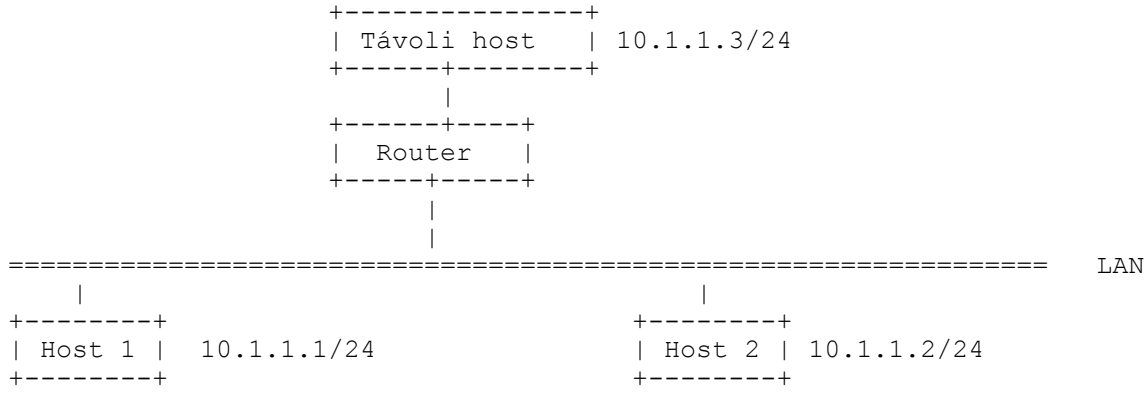
- Egy rosszindulatú támadó hamis ARP adatokkal mérgezhet
- Elterelhet forgalmat
- Túlcsordíthat ARP cache-t: DoS (Denial of Service)
- WiFi növeli a veszélyét !
- Védekezés: ARP táblákat be lehet vasalni

- Fáradságos, de nagy biztonságot követelő helyeken szükséges

Man in the Middle támadás ARP cache poisoninggal

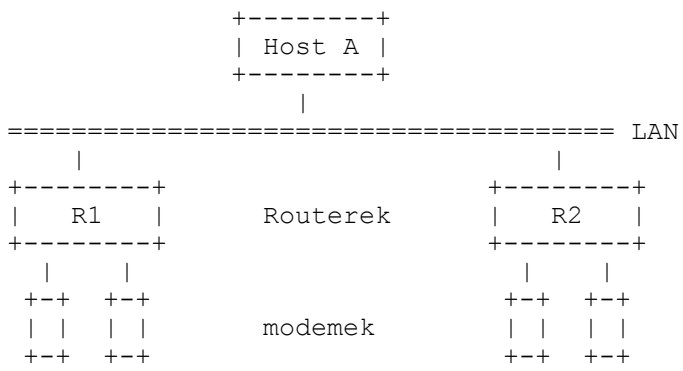


Proxy ARP



- A router a távoli host-ot a LAN-on jelenlevőnek mutatja
- Válaszol a 10.1.1.3 IP címre érkező ARP kérésekre

UNARP



UNARP probléma

- Modemeken át jönnek állomások
- A LAN-on használt IP címekből használnak
- R1 és R2 proxy arp-t ad nekik
- Egyik pillanatról a másikra átkerülhetnek R1-ről R2-re
- Host A rossz ARP információt hihet !

UNARP megoldás

- RFC 1868
- Ethernet ARP broadcast
- Reply, a source ethernet cím üres!

Gratuitous ARP

- A saját IP címemre adok ki ARP kérést
- Lehet pl. egy ping következménye

Mi van, ha nem tudjuk a saját IP címünket?

RARP

- RFC 903
- Külön ethernet frame type: 0x8035
    - Lehetne ARP is
- Broadcast kérés
- A formátum megegyezik az ARP-vel
- A válaszban a címzett IP címe és ethernet címe
- A válasz már unicast

RARP hátrányok

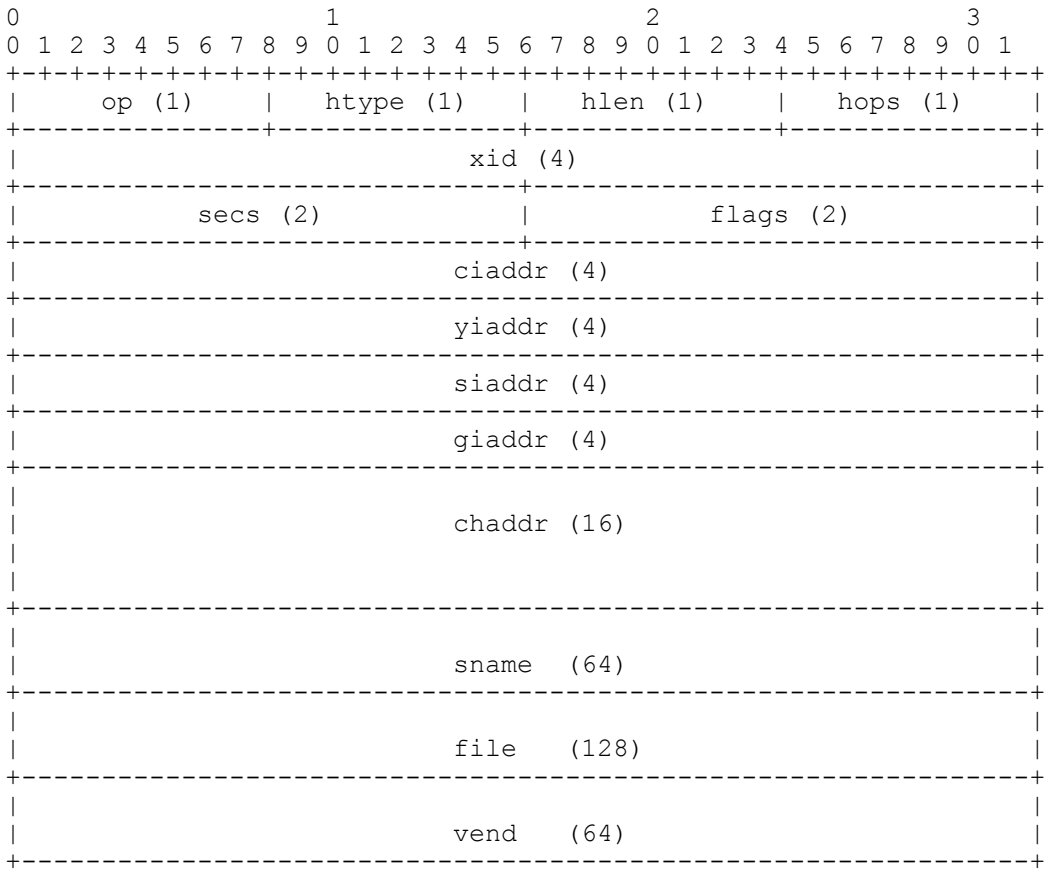
- Csak egy puszta IP címet ad vissza (routert, netmaskot nem)
- Nem route-olható

8. BOOTP, DHCP

BOOTP

- Kliens gépek automatikus konfigurálása, indítása
  - Pl. X terminálok
- RFC 951, RFC 1542
- IP címen kívül visszaad
  - Netmaskot
  - Router címet
  - Boot server host címet
  - Boot fájlnevet
  - ...
- UDP csomagok
  - server port: 67
  - kliens port: 68
- A kérés ethernet broadcast, a válasz unicast

BOOTP csomagformátum



- Opcode: 1 - request, 2 reply
- Hop count: proxy szerverek eggyel növelik
- xid: tracnsaction id. Ezzel derül ki, hogy mi mire válasz
- secs: az első bootp kérés óta eltelt idő

- ciaddr: Client IP address. Kérésben a kliens kitöltheti: ezt kérem
- yiaddr: Your IP address. Ezt a címet kapja a kliens
- siaddr: Server IP address. A válaszoló IP címe
- giaddr: Gateway IP address. Ha proxy szerver van, annak a címe
- chaddr: a kliens ethernet címe (redundáns)
- sname: server host name
- file: boot fájlnév
- vend: különböző kiegészítések
    - tag kód, hossz, érték
    - Netmask
    - Router(ek) IP címe
        - Több is lehet
    - DNS szerver IP címe

DHCP

- Dynamic Host Configuration Protocol, RFC2131
- BOOTP kompatibilis
    - A ‚vend' helyett az ‚options' szó használatos
    - 312 byte hosszú lehet (64 volt)
    - DHCP „message type" opció
- Rugalmasan lehet IP címeket kiosztani
    - Permanens IP cím - nem jár le
    - Dinamikusan - egy poolból véletlenszerűen, meghatározott időre
    - Manuálisan - ugyanaz a kliens mindig ugyanazt kapja
- Nem lehet DHCP-vel
    - DNS bejegyzést eszközölni
    - Egy routert konfigurálni
- Message types
    - DHCPDISCOVER
    - DHCPOFFER
    - DHCPREQUEST
    - DHCPDECLINE
    - DHCPACK
    - DHCPNAK
    - DHCPRELEASE
    - DHCPINFORM
- 4 lépcsős kezdeti handshake
    - DISCOVER (broadcast) -->
    - OFFER <--
    - REQUEST -->
    - ACK <--
- Lease
    - Egy kiosztott IP cím/ethernet cím pár, a hozzá tartozó lejárati idővel
    - A szerver az összes kiosztott lease-ről nyilvántartást vezet

Leases adatbázis részlet

```
lease 10.2.19.105 {
        starts 3 2004/10/13 11:07:41;
        ends 3 2004/10/13 23:07:41;
        hardware ethernet 00:0d:60:8d:49:34;
```

```
        uid 01:00:0d:60:8d:49:34;
        client-hostname "Pingvin";
}
lease 10.2.19.110 {
        starts 2 2004/10/12 13:12:05;
        ends 2 2004/10/12 13:14:05;
        hardware ethernet 00:11:92:fa:ee:00;
}
```

DHCP szerver konfiguráció fájl részlet

```
subnet 10.18.48.0 netmask 255.255.255.0{
  range 10.18.48.150 10.18.48.200;
  option subnet-mask 255.255.255.0;
  option broadcast-address 10.18.48.255;
  option domain-name-servers 193.224.194.225;
  option routers 10.18.48.254;
  option domain-name "sapientia.hu";
}


host eva {hardware ethernet 00:b0:d0:08:02:79; fixed-address 193.224.194.26;}
```

9. ICMP, ICMP hibaüzenetek

Általánosan összefoglalva:

Az TCP/IP hálózatok működésében kulcsfontosságú szereppel járó vezérlőprotokoll, amelynek feladata a hálózaton fellépő hibákról történő értesítések küldése, illetve azok kezelése.
Az ICMP legismertebb, felhasználó által is látható alkalmazásai a ping ill. traceroute segédeszközök, amik az ICMP ECHO műveletének segítségével teszik lehetővé a megfordulási, illetve utazási idő mérését, valamint a köztes hálózati csomópontok felderítését két gép között.

Az IP fejléc után, az adatrészben van az ICMP fejléce, és a hozzá tartozó adatok.

Egy ICMP csomagnak van típusa. A típus attól függ, hogy az üzenet mire vonatkozik.

Az alapvető típusok:

Szám	Név	Mire való
0	echo-reply	Az echo-request -re válasz. Ping is használja.
3	destination-unreachable	'A cél elérhetelen' hibaüzenet. Bármilyen TCP/UDP forgalomhoz kell, enélkül a végtelenségig várnánk a kapcsolatra.
5	redirect	útválasztás, ha nem fut útválasztó démon
8	echo-request	Visszhang kérés. Ilyet küld a ping.
11	time-exceeded	Jelzi, hogy lejárt a csomag ideje. Traceroute használja.

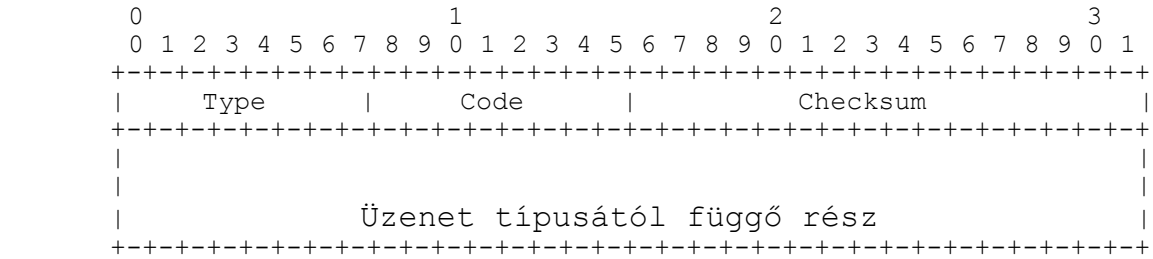
Az ICMP csomag általános felépítése

0...7 (bit)	8...15 (bit)	16...23 (bit)	24...31 (bit)
Típus	Kód	Checksum	
Típus függő adatok			

Előadásból:

- ICMP - RFC792
- STD 5 (IP-vel együtt)
- To control = Vezérel
- Bizonyos szempontból az IP fölött levő réteg
  - IP csomagokat használ
  - IP protocol mező: 1
- Bizonyos szempontból az IP alatti réteg
  - Az IP viselkedését (is) befolyásolja
    - Hibaüzenetek
    - Szolgálati üzenetek: routing, netmask stb.

ICMP formátum



- Type: az elsődleges információ, az üzenet típusát határozza meg

- Code: bizonyos üzeneteknél az üzenet altípusa
- Checksum: egyszerű összeadás, mint az IP-nél, az egész ICMP üzenetre, 16 bites darabokban

ICMP üzenet-típusok

Két fő csoport

- Hibaüzenetek
- Vezérlő, konfiguráló üzenetek

A hibaüzenetekre nagyon szigorú szabályok vonatkoznak
Sose eredményezhet hibaüzenetet:

- ICMP hibaüzenet
- IP broadcast, vagy multicast
- Alacsonyabb (link layer) broadcast, vagy multicast
- Egy IP csomag többedik (nem első) fragmentuma
- Olyan IP csomag, aminek forráscíme nem egy host IP címe
- IGMP (Internet Group Management) üzenetek

A hibaüzenet mindig tartalmazza a kiváltó IP csomag lényeges részét

- A teljes IP fejrészt (20-60 byte)
- Az első 8 byte-ját az IP adat résznek
- TCP és UDP esetén ez tartalmazza a portokat
    - A vevő tudja, hogy melyik programot érinti a hiba

10. ICMP vezérlő (nem hiba) üzenetek

Általánosan összefoglalva:

Órák szinkronizálása és csomag haladási idejének becslése (Timestamp request and reply)

A gépek lekérdezhetik a másik gép óráját, amely korrigálható a csomagok haladási idejével. Ezzel a mechanizmussal egyúttal a csomag haladási ideje is becsülhető az adott útvonalon.

Subnet maszk kérés és válasz (Address mask request and reply)

Subnet használata esetén a Subnet Mask határozza meg, hogy az adott osztályú cím hosztokat azonosító részéből mely bitek határozzák meg az adott fizikai hálózatot. Ez a maszk beállítható a hoszt hálózati konfigurálásakor, de ezzel az ICMP kéréssel be is szerezhető a router-től. A kérés a router-nek közvetlenül is elküldhető, vagy broadcast-tal is kérhető.

Ping: Az egyik legismertebb a „ping”, amely egy „ICMP_ECHO_REQUEST” parancs, amellyel a távoli gépet arra kényszerítjük, hogy egy „ICMP_ECHO_REPLY” választ küldjön. Így ellenőrizhetjük, hogy a címzett gép működik. Mivel az ICMP üzenetek nem tartalmaznak PORT számokat, mivel nem egy alkalmazásnak szólnak.

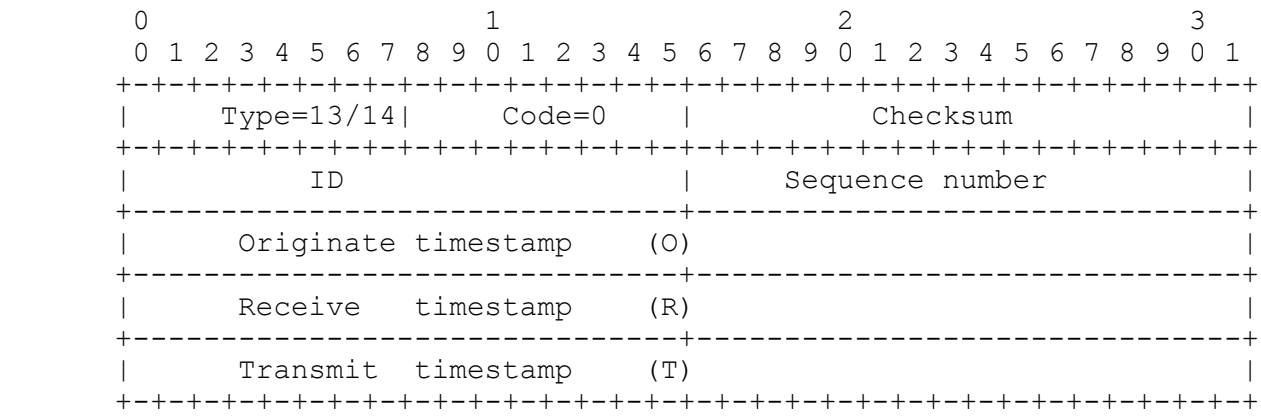
Előadásból:

ICMP address mask request/reply

- Request típus: 17, broadcast
- Reply típus: 18. unicast
- RARP-pal kapcsolatban használatos
- Több válasz is érkezhet
  - A RARP csak egy puszta címet ad
  - Nekem is az legyen a netmaskom, ami a többinek
- Kiment a divatból
  - DHCP betölti ezt a funkciót is

ICMP Timestamp request/reply

- Request típus: 13
- Reply típus:14
- Mindkettő unicast



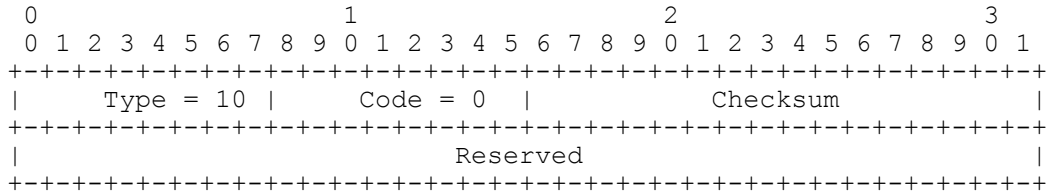
- Időegység: az UTC (Coordinated Universal Time) éjfél óta eltelt milliszekundumok
  - Ha a felső bit 0
  - Ha a felső bit 1, akkor más idő is lehet - implementáció függő
- A gyakorlatban - ma már - a receive és a transmit timestamp egyezik

- Óra beállításra alkalmas
  - Legyen RTT a request küldés és a reply fogadás közti idő
  - Ha egyformán jár az óránk, akkor O+RTT/2=R
  - Ha R ennél nagyobb, akkor a mi óránk késik, ha kisebb, akkor siet
  - Alternatívák:
  - Ravaszabb órabeállító protokoll: NTP (Network Time Protocol)
  - 13. TCP/UDP port: daytime. A helyi időt mutatja olvasható formában

Router advertisement/router solicitation

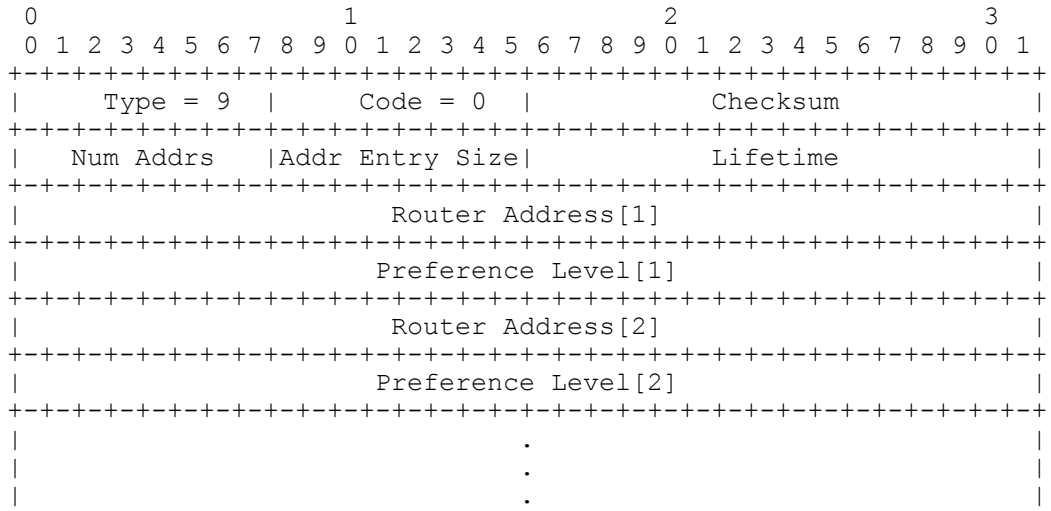
- RFC1256
- Dinamikusan, ICMP üzenetek által állít route-okat

Router Solicitation



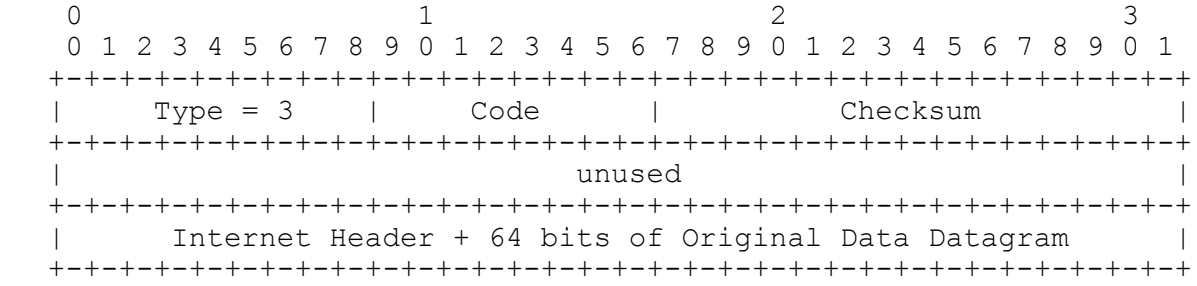
- Type = 10
- Multicast: 224.0.0.2 = all routers
- A routerek unicast-tal válaszolnak: router advertisement ICMP csomaggal

Router Advertisement



- A **default** router címét hirdeti
- Num Addrs: ennyi router címet hirdetek
- Addr entry size: ennyi 4 byte-os érték egy entry (=2)
- Lifetime: ennyi másodpercig érvényes ez a hirdetés
- Router Address: a router IP címe
- Preference Level: előjeles szám, minél nagyobb, annál jobban preferáld
- Nem csak solicite-ra válaszul, unicasttal, hanem multicasttal is
  - 8-10 percenként, véletlent belekeverve küldik
  - A 224.0.0.1 címre = all hosts
- Nem lehet hálózat/router vagy host/router hozzárendelést megadni

Destination unreachable

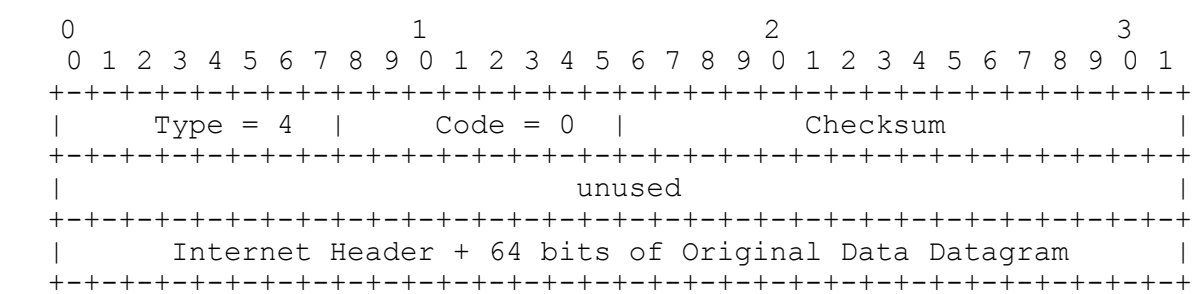


- Gyakori hibaüzenet
- Küldheti a címzett, vagy egy közbülső router
- Tűzfalak is küldhetik előbb, akár a címzettet mímelve
- A code mező mutatja a finomabb okot
    - Network unreachable: router küldi, ha zavarba jön, nem találja a címzettet
    - Host unreachable: az utolsó router küldi, aki úgy érzi, látnia kéne, de ilyen nincs
    - Protocol unreachable: többnyire nem UDP/TCP-vel kapcsolatos hiba
    - Port Unreachable
        - Jellemzően a címzettől jön
        - TCP-re nem jellemző
            - Ott RESET-tel bomlik a kapcsolat
    - Fragmentation needed but DF set = Túl nagy csomag
        - Egy közbülső router küldi
        - A következő MTU kisebb mint a csomag
        - A csomagban kérték, hogy DF: Don't Fragment
        - A második 4 byte-os szóban elküldheti a bajt okozó MTU-t
            - Ez eredetileg használatlan
            - RFC 1191 vezette be

Path MTU discovery

- Különösen TCP kapcsolatoknál fontos
- Lehetőleg nagy csomagokat akarunk küldeni
- De nem akarjuk, hogy közben fragmentumokra szedjék a csomagokat
- A cél címig mekkora a legkisebb MTU?
    - Időben akár változhat is !
- RFC 1191
    - A küldőnek van egy változója a célcím-hez tartozó MTU-ról
    - Kezdetben a célcím route-jához tartozó MTU
    - Csökkenti, ha „túl nagy csomag" üzenetet kap

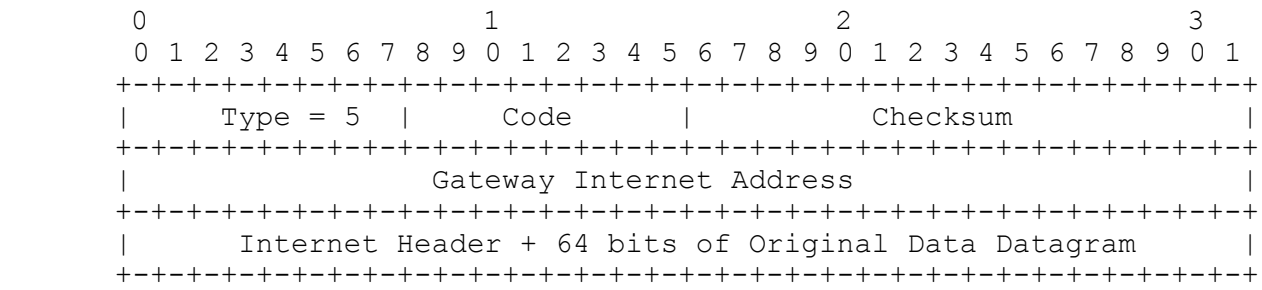
Source quench - Lassíts!



- Egy közbülső router, vagy a célállomás küldheti

- Ha eldobta a csomagot, mert nem tudta már továbbítani
- Ha már közel van ehhez az állapothoz
- A küldő állomás visszafogja magát
    - A visszaküldött csomagból látszik, hogy hol is!
    - Egy idő után dönthet úgy, hogy újra erőteljesebben küld
- Tűzfalak kiszűrhetik
    - hiba
    - furcsa jelenségeket okozhat
    - esetleg sokáig nem vesszük észre

Redirect - A célállomást rövidebb úton is eléred, ha erre keresed

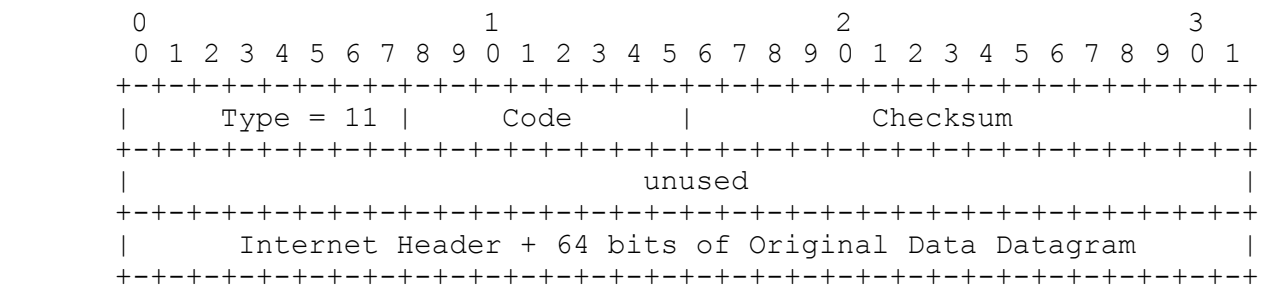


- Router küldi
- Megjelöl egy másik routert, ami kedvezőbb
- Csak akkor, ha látja, hogy a küldő és a kedvezőbb router egy hálózaton van
- A küldő állomás módosítja a routing tábláját
- Visszaélésre ad módot
    - A default routeren kívül mástól nem szabad elfogadni
    - Lehet, hogy még a default router-től sem!

Redirect Code

- 0 = Redirect datagrams for the Network.
- 1 = Redirect datagrams for the Host.
- 2 = Redirect datagrams for the Type of Service and Network.
- 3 = Redirect datagrams for the Type of Service and Host.

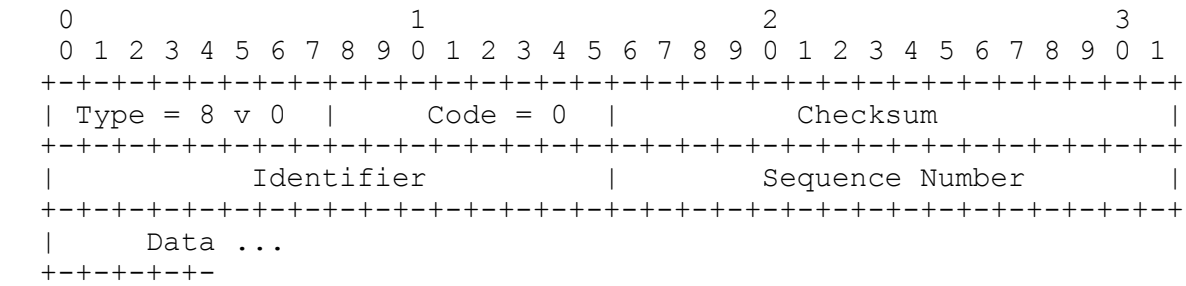
Time exceeded



- Eldobtam a csomagod, mert mire ideért lejárt a TTL
- Code
    - 0 = time to live exceeded in transit;
    - 1 = fragment reassembly time exceeded.
        - Lejárt a timeout, és nem sikerült az egész (részekben küldött) datagram-ot összeállítani

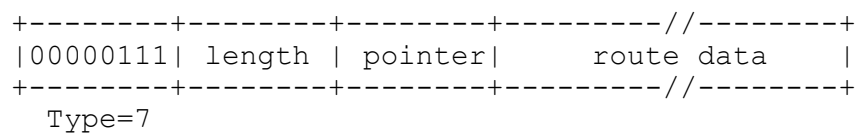
        - Ha az első fragmentum nem érkezett meg, akkor nem!
- A visszaküldött csomagból látszik, hogy melyik kapcsolathoz tartozik

Echo request/reply - a ping program eszközei



- Type: request = 8, reply = 0
- ID: egy ping instanciát azonosít, (= processz ID)
- Sequence Number: egy instancián belül a sorszámot
    - Lehet, hogy más sorrendben kapjuk vissza az echot!
- Adat: a küldött adatot vissza kell kapjuk
- ping program
    - -R kapcsoló
        - Az **IP** record route opcióját kapcsolja be
        - Mindkét irányban láthatjuk a közbülső routereket

IP record route opció



- length: ennyi byte az opció
- a route data length-3 hosszú
- pointer: a következő IP cím helyét mutatja: először 4, legfeljebb 40
    - 40: tele az IP header
- Az adat 4 byte-os IP címekből áll

Traceroute

- Az IP record route opció legfeljebb 9 router címet tárol
- Nem mindenki engedi át
- 1, 2, 3,... TTL-lel küld UDP csomagokat
- Az i-edik menetben küldött csomagot az i-edik hop router utasítja el ICMP time exceeded üzenettel
- UDP 33434-től egyre nagyobb portokra küld csomagokat
    - Ezzel azonosítja a választ
- Tűzfalak korlátozhatják
- Alternatívák
    - traceroute -I: ICMP csomagokat küld
    - mtr (my/Matt's traceroute): ICMP csomagok, szép felület
    - tcptraceroute: TCP csomagok, a 80-as, vagy bármely más portra

kellhet még: Link state/Distance vector (OSPF/RIP) összehasonlítás, lásd következő tétel vége

11. Spanning Tree Protokoll

Spanning Tree Protocol

- DEC találmány, IEEE átvette: 802.1D
- Ethernet típus: BPDU (Bridge Protocol Data Unit)
    - Kissé anakronisztikus a bridge szó...
    - Nem ethernet II, hanem 802.3
    - Használt cél cím: 01:80:c2:00:00:00 - multicast
    - A switch-ek egymás közül root bridge-et választanak
- Minden switch a root-hoz vezető minimális utat veszi be a fába
- Ha több egyenlő súlyú út van, akkor a preferáltabb node-on át vezető nyer
    - Az ehhez tartozó port: designated port, root port
    - A root switch kivételével minden switchnek egy pillanatban csak 1 van
- A BPDU-kat periódikusan küldik a switch-ek
- Az élekhez súlyokat rendelünk, alapértelmezésben a sebességük szerint
    - A root bridge-hez közvetlenül vezető élek 0 súlyúak
    - 10M - 100
    - 100M - 19
    - 155M - 14
    - 1G - 4
- A csomópontokhoz prefenciákat rendelünk: BID, Bridge ID, 8 byte
    - A kisebb BID preferáltabb, root az lesz, ahol a legkisebb a BID
    - Gyári érték, de konfigurálható
        - Célszerű konfigurálni, hogy ne egy perifférikus switch legyen a root!

BPDU szerkezet

- Az általam root-nak tudott BID
- Az root-hoz tartozó teljes út költsége
- Az én BID-em
- Port ID, amin küldöm ezt a BPDU-t
- Maximum age: ennyi idő után felejtsd el ezt az információt
    - Jellemző érték: 20 sec
    - Legalább ennyi időnként jönnek a BPDU-k

Port állapotok

- Blocking
    - Bekapcsolás után, kiküld egy BPDU-t: 'itt vagyok'. Van-e itt valaki ?
    - Tartalék üzemmódra kapcsolt port (nem nyert az algoritmusban)
- Listening - blocking state után, BPDU-kat figyel
- Learning - listening után, fogad és küld BPDU-kat, felépíti a táblázatait
- Forwarding - konfigurálás után úgy látja, hogy ez részt vesz a spanning tree-ben
- Disabled - az algoritmus által kikapcsolt, hibás

12. Distance vector routing protokollok, példa: RIP

Általánosan összefoglalva: (RIP és OSPF összehasonlítás)

a RIP egy kezdetleges, kizárólag Broadcast hálózatok routeolását megoldó protokoll. Ezt akkor érdemes használni, ha hálózatunk nem nagy kiterjedésű, és nem kell a hálózati forgalmat megosztani költséghatékony módon. Nem túl célszerű RIP-et használni, ha az egyes hálózati elemek ponttól-pontig vannak összekapcsolva. Különös figyelemmel kell venni, hogy a RIP esetén a 16-os költségmutató mér elérhetetlen célt jelent. Így nem lehet nagyobb kiterjedésű hálózatot létrehozni. A RIP megvalósítása Linux rendszerekben is a routed daemon. A `routed` működéséhez fontos megadni az egyes hálózatok paramétereit (Hálózatcím, netmaszk, átjáró).

```
root@router:/# route add -net 10.0.1.0 netmask 255.255.255.0 dev eth0

root@router:/# route add gateway 10.0.1.1
```

Az OSPF meglehetősen bonyolult, viszont nagyon sok jó tulajdonsággal rendelkezik. Az OSPF használata a HeniNet esetében lehetőség szerint elég jó választás, hiszen a hálózatterhelés elosztásával jól kezelhetők a csomagok időzítései és a linkek sávszélességei is. Nem mellékes szempont az esetleges pont-pont összeköttetések route-olásának megoldásait is. Az OSPF megvalósítása Linux rendszerekben a zebra nevű daemon segítségével lehetséges. A zebra képes használni a régebbi RIP, RIPv2 RIPng, OSPFv2, OSPFv3, BGP4, BGP4+ routing protokollokat is. Így könnyedén válthatunk például RIP használatáról OSPF-re, illetve a későbbiek során IPv4-ről IPv6-ra. Tehát a zebra daemon használatával képesek vagyunk hálózatunkat IPv6-ra felkészítenünk anyagi megterhelés nélkül.

A DVMRP az MBONE routing protokollja [RFC1075]. Az MBONE egy kísérleti multicast gerinc, mely nem önálló fizikai összeköttetéseken, hanem a rendes unicast Internet fölött létrehozott alagutakban (tunnel) működik. Alapvetően multicast-képes szigetekből áll, a multicast router-ek egymás között pedig egymásnak címzett IP csomagokba csomagolva adják át a továbbítandó csomagokat. Maguk a multicast router-ek a DVMRP-t futtatják, ami leginkább a RIP-hez hasonlít, de nem a célpontok felé vezető legrövidebb utat, hanem a források felől vezető legrövidebb utat számítja ki. Ezt az információt az RPF tisztogatásokkal működő változata használja fel a multicast csomagok továbbítására. A csomagokat azonban nemcsak a router minden interface-én hanem az alagutakon is továbbítják.

Az alagutakat a két router rendszergazdájának manuálisan kell konfigurálnia. Minden alagútnak három paramétere van: a másik végén lévő router, az alagút költsége és egy küszöbérték. Az alagút költségét a distance-vector protokoll használja fel a legrövidebb út számításához, a küszöbérték pedig a forgalom korlátozására szolgál. Ha egy csomag TTL mezője nem nagyobb, mint a küszöb, az adott alagúton nem továbbítják. Megegyezés szerint a szervezetek közötti alagutak 32, a régiók közöttiek 64, a kontinensek közöttiek pedig 128-as küszöbbel rendelkeznek, így egy 128-nál kisebb TTL-lel elküldött csomag biztos nem vesz majd igénybe óceán feletti vonalat.

Az MBONE kísérletnek indult, mára pedig több mint 10000 tagja van. A multicast router-ek feladatait többnyire közönséges munkaállomások látják el, melyeken a régi szép időkhöz hasonlóan egy „mrouted" nevű program végzi a route-olást, a program a kísérlet része és ingyenes. Hiába kísérlet azonban, az MBONE-t ma már üzemszerűen használják, ennek segítségével előbb audio-, majd videoközvetítést hallhatunk-láthatunk az IETF gyűlésekről, ám mindennek véget kell érnie, hisz a DVMRP nem képes ilyen nagy hálózatot a siker esélyével kiszolgálni. Erre jobb protokollok és dedikált router-ek kellenek.

Routing protokollok

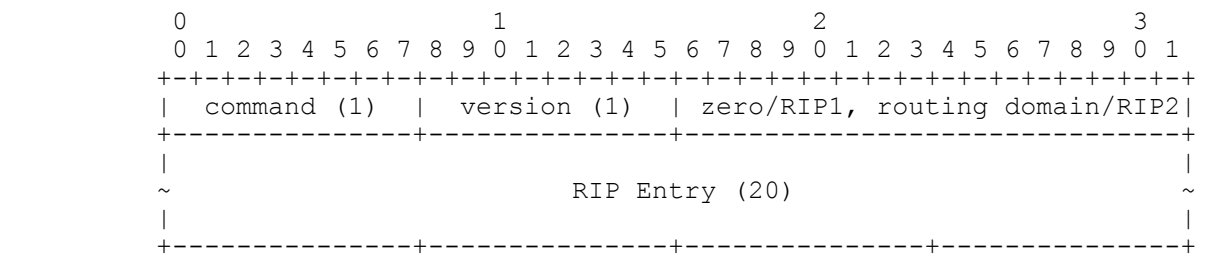
- Egyszerű esetben meg lehet úszni egy default route-tal
- Az internet attól szép, és működőképes, hogy időben és térben változatos és folyton változó a hálózatok/csomópontok összekötése
- Nem elég a statikus routing információ: időben változó *dinamikus* kell
- Nem csak egy utat veszünk figyelembe, hanem többet: *multipath* routing kell

Distance vector protokollok

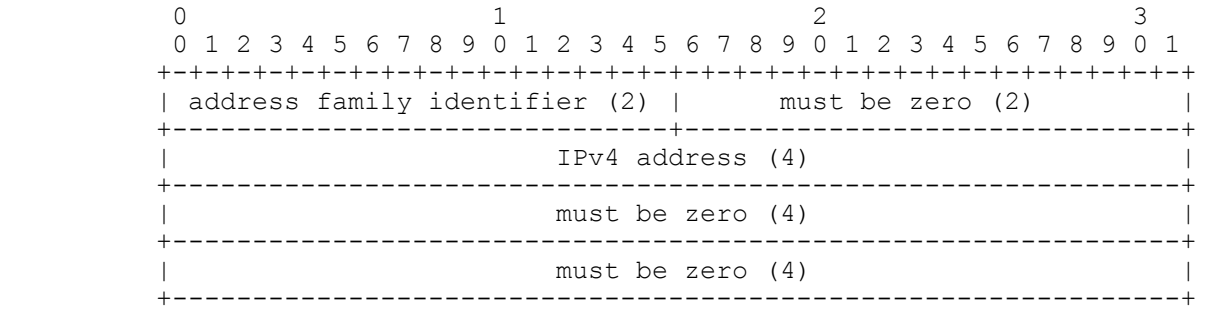
- Minden router, minden célpontról (hálózatról) küld információt
  - Milyen célpontot
  - Milyen messze látok: súly, távolság, költség, metric, (pl. hopcount)
  - Melyik szomszéd router fele (a célpont „tulajdonosa")
- A *szomszédos* routereknek periodikusan elküldi a saját képét a hálózatról
- A szomszédos router 1-et hozzáad minden kapott értékhez
- A régi saját táblázatát, és a most kapottat összefésüli - eldobja a rosszabb utakat
  - Ha egy célpont „tulajdonosától" nagyobbat kap, azt is elfogadja
    - Csak csökkenni tudna a költség
- A saját képét ő is elküldi az összes többinek
- Lassan beáll egy állapot: konvergencia
- Periodikusan kötelező update-et küldeni
- Ha sokáig nem kapunk valahonnan update-et, az ő útjait elfelejtjük
- Triggered update: ha változás van (pl. meghal valami, vagy update-et kap), időn kívül is küld update-et
- Példa: RIP, IGRP

RIP - Routing Information Protocol

- RIP1 - IP broadcast
- RIP2 - IP multicast: 244.0.0.9



- Command: 1 = request, 2 = response
- Version: 1 (RIP1) vagy 2 (RIP2)
- Routing domain: azonosító, egy hálózaton/gépen több RIP instancia is futhat, azok közt választ
- Egy RIP1 entry



```
   |                         metric (4)                            |
   +---------------------------------------------------------------+
```

- Address family: IP-nél 2
- IPv4 cím: hálózat, vagy host cím
- metric: távolság

- Egy RIP2 entry

```
 0                   1                   2                   3 3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Address Family Identifier (2) |        Route Tag (2)          |
+---------------------------------+-------------------------------+
|                         IP Address (4)                        |
+---------------------------------------------------------------+
|                         Subnet Mask (4)                       |
+---------------------------------------------------------------+
|                         Next Hop (4)                          |
+---------------------------------------------------------------+
|                         Metric (4)                            |
+---------------------------------------------------------------+
```

- Subnet mask: a hirdetett címhez/tartományhoz tartozó maszk
- Next hop: én erre az IP címre route-olom ezt

Fontos fogalmak:
Counting to infinity - Split horizon - Poisoned reverse(kapcsolat szakadás esetén a hirdetés)

RIP2 autentikáció

13.Link state routing protokollok, OSPF

Az OSPF (Open Shortest Path First) egy link-state protokoll, melyet az IETF Interior Gateway Protocol munkacsoportja fejlesztett ki elsősorban a RIP hiányosságai miatt. 2. verziója az RFC1247-ben 1991 júliusában jelent meg és körülbelül ötször olyan terjedelmes, mint a RIP leírása. Valóban az OSPF bonyolult, ám sokkal kifinomultabb, kevesebb sávszélességet foglal, hurokmentes és számos más előnnyel rendelkezik a RIP-hez képest.

A link-state protokollok működése két részből áll. Először minden állomás felderíti a hálózat topológiáját, majd a kapott gráfban megkeresi a legrövidebb útvonalat és az ahhoz tartozó első állomást, amely felé továbbítani fogja a csomagot. A hálózat topológiáját a link-ek állapotát leíró rekordok (link-state records) terjesztésével tudatják egymással az állomások. A link-ek állapotát leíró rekordokat időbélyeggel látják el a router-ek, majd minden irányban terjeszteni kezdik.

A multipath routing alatt két lehetőséget értük. Az egyik esetben csak akkor osztjuk meg a forgalmat több útvonal között, ha holtversenyben a legolcsóbbak. A második esetben a forgalom egy részét olyan útvonalra engedjük, amelyik nem a legolcsóbb, de még elfogadható. Mindkét megoldás esetén kisebb lesz a csomagok késleltetésének ingadozása, a több útvonal miatt az effektív sávszélesség is nagyobb és az egyik - például a legolcsóbb - útvonal kiesése esetén a forgalom mintázata nem annyira ugrásszerűen változik meg, hisz a csomagok egy része eddig is más útvonalon haladt.

Az OSPF háromféle link-et különböztet meg.
1. Pont-pont link-ek 2 router között
2. Broadcast jellegű link-ek, mint például Ethernet vagy FDDI, ahol egy csomaggal minden állomáshoz információt juttathatunk el.
3. Nem broadcast jellegű linkek, mint például X.25 vagy ATM.

Az OSPF 3 alprotokollból áll.
1. A Hello protokoll, ami segítségével a router-ek a link-ek állapotát tesztelik, felderítik egymást és meghatározzák a kiválasztott router-t.
2. Az Exchange protokoll, ami segítségével topológiai adatbázisok szinkronizációja folyik.
3. A Flooding protokoll, ami a link-state rekordok terjesztéséért felelős.

Az OSPF egy nyitott, szabvány, mely folyamatosan és nyitott módon fejlődik. Ennek egy nagy előnye van: a gyártófüggetlenség.

Előadás

- Minden router a saját szomszédjairól ad információt
- Az egyes routerek *minden más routernek* elküldik ezt
- Minden router maga kiszámítja, hogy mi merre van optimálisan
- A résztvevő routerek egységesen látják a hálózat topológiáját
- Ha változást észlel egy router, akkor hirdeti
    - Mindenki újra számol

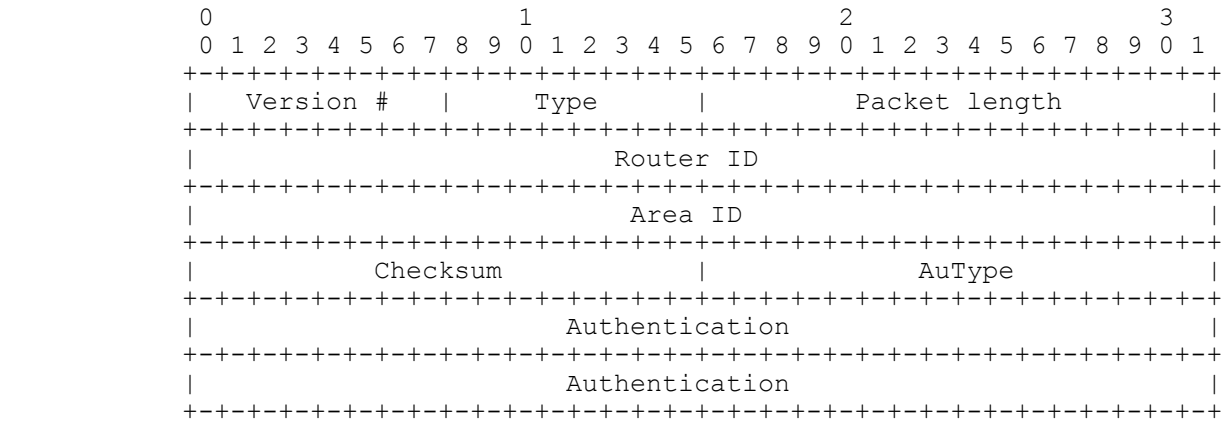
Dijkstra algoritmusa - Open Shortest Path First

- Legyen *G* egy irányított gráf, élei súlyozottak. Legyen *x* és *y* két pont. Válasszunk ki minimális súlyú utat *x*-ből *y*-ba!
- Két segédváltozó: *W* a bevett pontok, *B* a bevett élek. Kezdetben *W={x}, d(x,x)=0*.
- Minden *W*-ben levő *u*-ra és *v*-re ami nincs *W*-ben számoljuk ki *d(x,u)+w(u,v)*-t, és vegyük

a minimumát (ha több van, bármelyiket). **Azt** a *v*-t és (u,v)-t vegyük be, ahol ez minimális.
- Folytassuk, amíg *y* nem lesz *W*-ben.

OSPF - Open Shortest Path First

- RFC 1583 - pár száz (!) oldal
- Nem UDP, nem TCP: saját IP protokol: 89

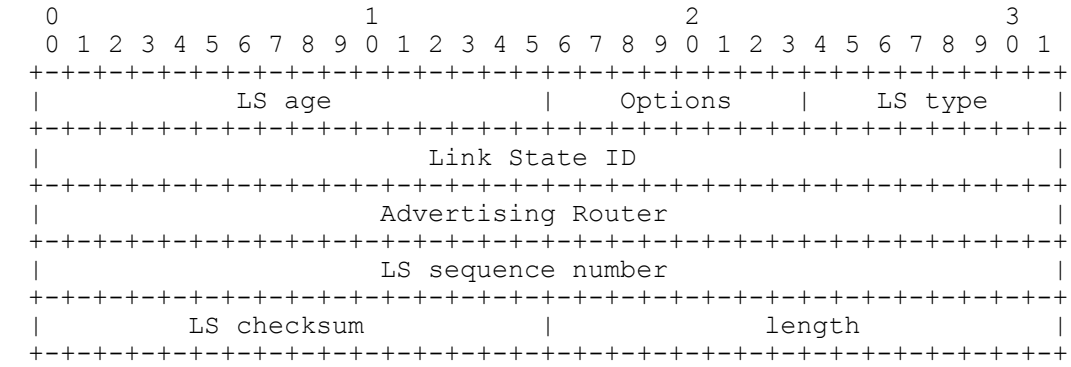


- Version: 2 a kurrens
- Type
  - Hello: szomszédok közti kapcsolatfelvétel/kapcsolattartás
  - Database description: a hálózat topológiájának leírása
  - Link-state request: információt kérek
  - Link-state update
  - Link-state ack: az update nyugtája
- Packet length - az egész OSPF üzenet hossza
- Router ID - az én azonosítóm
- Area ID - erre az area-ra vonatkozik a csomag
- Checksum - az IP -nél szokásos összeadás
- Authtype: különböző area-kban különböző lehet
  - 0: nincs autentikácaó
  - 1: jelszó
  - 2: MD5
    - Az egész csomagból, egy sorszámból és a jelszóból számolva
- Authentication a jogosultsági információ, jelszó MD5 szumma

OSPF

- Hello protokoll
  - A router a szomszédjairól szóló információt az összes szomszédjának elküldi
  - Szomszédok lesznek, ha
    - Azonos az Area ID
    - Azonos az autentikáció
    - Időzítések egyeznek
    - Stub flag egyezik
- DR, Designated Router, BDR, Backup Designated Router
  - Az egy szegmensen levő routerek választják maguk közül
  - A OSPF adatbázist ezek közvetítik
  - Ha *n* router van, akkor nem *n*n*, csak *2*n* tranzakció

    - Szomszédos (Adjecent) router:
        - Saját magam látom az ő hello üzenetében
        - A szomszédos routerek kicserélik a teljes adatbázisukat
        - Update-eket küldenek egymásnak
        - Egy szegmensen a DR-rel és a BDR-rel mindenki szomszédos
- Domain (AS) a hálózat azon része, aminek egyforma képe van a topológiáról
    - ASBR: Autonomous System Border Router - más AS-ekhez interfész
- Area: egy domain önállóan kezelt része, amiben az SPF működik
    - Inter area routing
        - ABR: Area Border Router
        - Nem csak a saját interfészét, hanem az area-k közti interfészeket is számontartja
        - Area 0, backbone area. Az areak összekötésére szolgál.
    - Intra area routing
- Virtual link
    - Lehet, hogy egy area nem kapcsolódik a 0-s area-hoz
    - Lehet, hogy a 0-s area nem összefüggő, több darabból áll
    - Ilyenkor egy virtuális linkkel kell összekötni őket
- Költég - cost: egy router egy link-jének a jellemzője
    - Konfigurálható paraméter
    - Alapértelmezés: az interfész sebességének reciproka * 10^8
- Linkek:
    - Router linkek: részt vesznek egy area kialakításában
    - Summary linkek: inter area linkek egy AS-en belül (ABR dolga)
    - Hálózati linkek: ha több út vezet egy hálózathoz, azokat írja le (DR dolga)
    - Külső linkek: más AS-hez vezetők: ASBR dolga. Az itt megtanult utak információját közli az AS-sel: redisztribució.



- LSA - link state advertisement
    - Egy linkről szóló információ
    - Nyugtázandó
    - Továbbítják minden szomszédnak
        - LSA age: a hirdetés kora másodpercben
    - Update: csak a különbség az előzőhöz képest (inkrementális update)
    - Periódikus update: 30 percenként
        - Ha 1 óráig nem frissül, törlik
    - Típusok:
        1. Router
        2. Network
        3. Summary (IP network)

4. Summary link (ASBR)
5. AS external link

- Kényes, óvatosan konfigurálandó kérdés: a külső utakat milyen költséggel hirdetjük bent?
- Link state id - típustól függ
  - Router id
  - Network cím
- Sequence number: ugyanarra az LS-re vonatkozóan egyre nő
- Checksum: az egész LSA-ra, kivéve az LSA age-et

Link state/Distance vector (OSPF/RIP) összehasonlítás

- OSPF gyorsabban konvergál
- OSPF árnyaltabb: figyelembe vesz TOS-t, sávszélességet stb.
- OSPF nagyobb hálózatban is használható
- OSPF kevesebb hálózati forgalmat generál
- OSPF nem „flat" hierachia van benne (area-k)
- RIP egyszerűbb: könnyebben adminisztrálható
- RIP kisebb erőforrás igényű a router oldalán

14. BGP - Border Gateway Protocol

Bevezetés:

Az IETF kitalált tehát egy közbülső, új megoldást, az út-vektorokat (path vectors).A módszer az, hogy minden terjesztett útvonalban a célpontig vezető teljes útvonalat leírjuk. Így a hurokmentességet minden router könnyen ellenőrizheti:

A BGP 3 listát (Route Information Base, RIB) tart nyilván, egyben azokat az utakat tárolja, melyeket szomszédaitól hallott (RIB-In), a másikban azokat, melyeket terjeszt (RIB-Out), a harmadikban azokat, melyeket az AS használ (local-RIB). Az első két listából minden szomszédos AS-hez tartozik egy-egy, azoknak az utaknak amit onnan hallott és oda terjeszt. A BGP-4 legfontosabb változtatása az, hogy a célpontok immáron prefixek. A BGP implementáció nem IP cím + maszk párosával írja le a prefixeket, hanem egy 1 byte hosszúságú hossz mezővel, mely a prefix bitben mért hosszát adja meg és magának a prefixnek értékes bitjeivel. Maga a BGP kapcsolat TCP fölött épül ki. Ezzel elsősorban a protokoll vált egyszerűbbé, mert nincs szükség annak figyelésére, hogy elküldött üzenetünket nyugtázta-e már a vevő fél. Hátránya viszont, hogy így a titkosítás megoldhatatlan a TCP titkosítása nélkül, hiszen elég egy a TCP kapcsolat bontására szolgáló csomagot küldeni az egyik oldalnak és a kapcsolat lebomlik. Így könnyű zavart okozni. Másik hátrány, hogy a TCP meglehetősen érzékeny a torlódásokra (sok újraküldés), ám ez okos TCP implementációkkal kivédhető. Minthogy a TCP byte-stream jellegű kommunikációt tesz lehetővé, a BGP pedig üzenetekben kommunikál, minden üzenet elején 16 byte szolgál az üzenet elejének azonosítására és a jövendőbeli hitelesítésre. Ezen felül az üzenet tartalmazza saját hosszát, így a vevőnek nincs más dolga, mint bevárni a megadott számú byte-ot, utána rögtön a következő üzenet eleje jön majd.

A BGP router-ek a TCP kapcsolat kiépítése után egy OPEN üzenetet küldenek el

Ha a kapcsolat kiépült, a BGP router-ek a terjesztésre szánt összes útvonalukat közlik egymással, minden útvonalat külön frissítő (UPDATE) üzenetben. Miután ez megtörtént, már csak a változásokról értesítik egymást. A hibákról (például helytelen szintaktikájú üzenet, rossz BGP verziószám, stb.) egy harmadik fajta üzenetben értesítik egymást (NOTIFICATION).

Előadásból:

- RFC 1771 BGP - version 4
- Az interneten ezen alapul a routing, a backbone routerek ezt használják
- AS-eken belül is használják: iBGP - internal BGP
- Hirdetés - advertisement
- Adminisztrációs döntés kérdése:
    - A route-olás: policy based routing
    - A hirdetések elfogadása
        - Pl. ISP-k nem fogadnak el /25-ös hirdetéseket
- TCP alapú, 179-es port
    - BGP peers: konfiguráció kérdése
    - A peer-hez vezető út nem lehet BGP függő
        - Közvetlen szomszéd
        - Statikus út
        - AS-en belüli (IGP) route
    - Kezdetben teljes táblázat - később inkrementális update
    - Keep-alive üzenetek: alapértelmezésben 30 sec

- Distance vector: a célhoz vezető AS-eket tatja számon: *AS Path*
- BGP dampening: a gyakran változó hirdetéseket nem veszik figyelembe
    - Ha lejár egy timeout, visszaveszik
- Route-olás:
    - A specifikusabb (hosszabb netmaszkú) út preferált
    - A lokális (AS-en belüli) út preferált (cold potato)
    - A rövidebb AS-path preferált
    - Végső döntés: kisebb IP cím

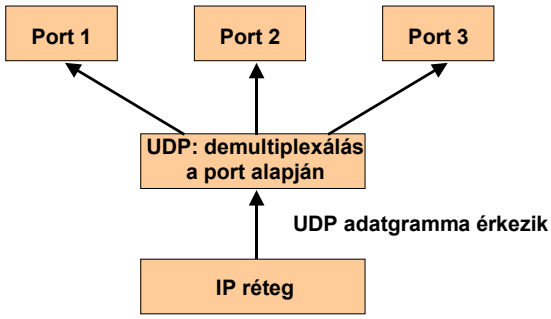
Looking glass: Az interneten elszórva http felületen lekérdezhető routerek; Diagnosztikai eszköz; BGP információ; Traceroute információ

Route szerverek: Az interneten elszórva telnettel elérhető routerek; Diagnosztikai eszköz

15. UDP – User Datagram Protocoll

Bevezetés:

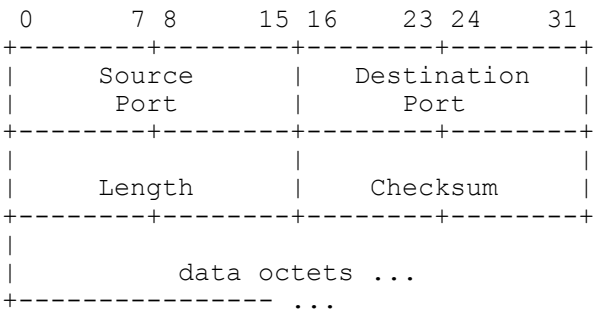
Az UDP szállítási protokoll biztosítja, hogy egy gépen egyidejűleg futó több alkalmazói program egymástól függetlenül küldhessen és fogadhasson csomagokat.
Az UDP csomag fejlécében lévő cél és forrás port szám biztosítja, hogy a csomag a megfelelő process-hez kerüljön feldolgozásra, és hogy a válasz üzenet is a megfelelő helyre érkezzen.
Az UDP az IP protokollt használja az üzenet továbbítására.



IP fölötti réteg demultiplexálása

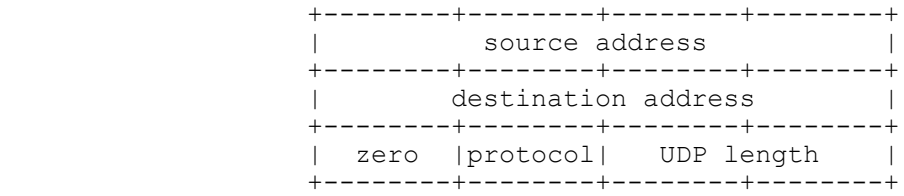
Előadásból:

- RFC 768, J. Postel, 1980
- Egyszerű
- Egy-egy IP csomag küldésére alkalmas
- Nincs kapcsolat felépítés, bontás
- Nincs garancia arra, hogy egyáltalán eljut a címzetthez a csomag
- Mégis, igen gyakran nagyon jó
  - DNS
  - NFS
  - Multimédia alkalmazások
- A magasabb szintek gondoskodhatnak a hiányzó funkciókról



- Forrás/cél port: eszerint demultiplexál az UDP réteg
- Lehet, hogy ugyanaz a sorszám egész mást jelent UDP/TCP portként
  - A gyakorlatban szokás mégis kapcsolatot teremteni: pl. 13 = daytime
- Hossz: byte-ban adja meg a fejrész, és az adatrész hosszának összegét
  - Minimum: 8, ha nincs csak fejrész. (Lehetséges)
  - Redundáns információ: az IP fejrészből kitalálható
- Csekkszumma: a szokásos one's complement összeadás

- Az IP csekkszumma csak az IP fejrészre vonatkozik
- Ha 0, akkor nincs (=>a 0-t FFFF-ként kell ábrázolni)
  - Az UDP csekszumma opcionális
  - Erősen ajánlott, de pl. ha csak ethernet van, nem igazán segít
- Nem csak az UDP csomagot, hanem egy „pszeudó fejrészt" is figyelembe vesz:

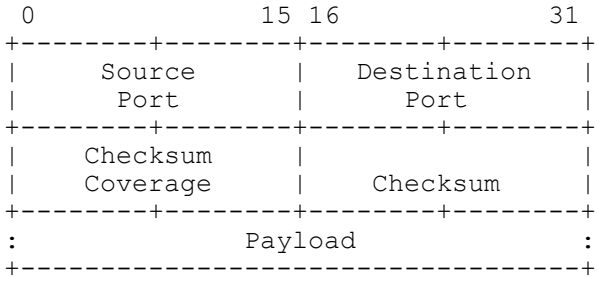


- Az IP fejrészből ismétel meg elemeket
  - A rossz helyre küldött UDP csomagokat lehet így kiszűrni
  - TCP-nél is van ilyen
  - Az UDP csomag hossza kétszer szerepel

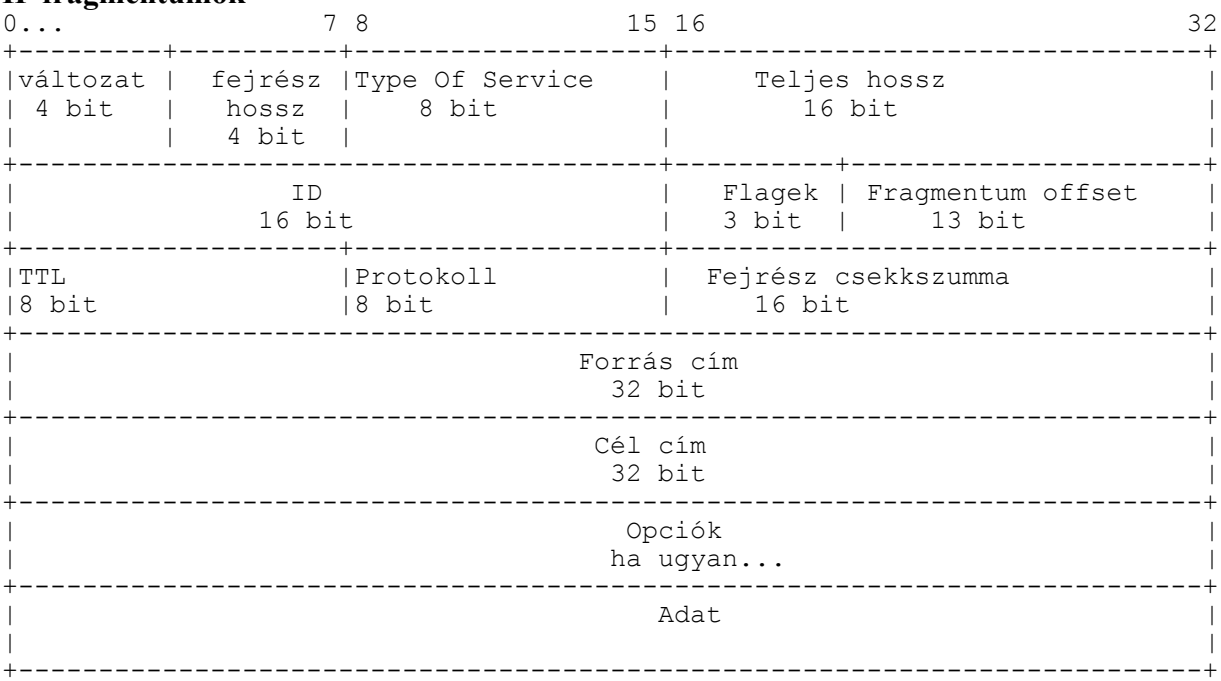
- Ha a csekkszumma hibát jelez, a csomagot egyszerűen eldobjuk, nincs hibaüzenet se
  - Vajon kinek, milyen portra menne ?

UDP lite

- Egyes alkalmazásoknál (pl. hang) lehet, hogy jobb megtartani a sérült csomagot is
- UDP Lite - RFC 3828 (2004 Július)
  - IP protocol id = 136
  - Az ismételt UDP length helyett: checksum coverage
  - Annyi byte-ot fog át a csekkszumma: legalább 8 - az UDP fejrészt mindenképpen tartalmazza



IP fragmentumok



- Lehet, hogy nagyobb az IP datagram, mint az eszközön az MTU
- Ilyenkor darabokra szedi az IP a csomagot: fregmentál
  - *Datagram*-okat több *packet-re*
  - Teheti a küldő oldal
  - Teheti *bármelyik* közbülső router
- A fregmentálás a felső réteg (UDP, TCP) számára transzparens
- Az egyes fragmentumokban külön-külön IP fejrész van
  - Az egyes fragmentumokban ugyan az lesz az ID
    - Ennek segítségével lehet összerakni a csomagot
  - A felsőbb réteg fejrésze nem ismétlődik
    - Következmény: a közbülső darabokból nem lehet tudni, hogy milyen portra mennek
- A „Teljes hossz" mező a fragmentum hosszát mutatja
- Ha nem az utolsó fragmentumról van szó, akkor áll a „More Fragments" bit
- A fragmentum offszet 8 byte-os egységekben a mutatja, hogy hova illik ez a darab
  - Következmény: az utolsó darab kivételével minden fragmentum *adatrészének* hossza 8 többszöröse kell legyen

Példák (értelmezése):

- `17:40:57.874418 lex.jak.ppke.hu > adsl240021.vnet.hu: udp (frag 30811:560@7400)`

- Ha Don't Fragment (DF) bit áll, és kisebb az MTU, akkor
  - Destination unreachable ICMP üzenet
  - Fragmentation needed code-dal
  - A next hop MTU-jával
  - ping -M do -s 1472 adsl-lel.bekapcsolt.valami

Maximális UDP csomagméret

- Elvileg max(IP csomag) - hossz(UDP fejrész+IP fejrész) = 65535 - 28 = 65507
- Az operációs rendszerek (TCP/IP rutin, kernel) korlátozhatják
- Általában 8192 byte (2**12)

- `17:51:42.132622 lex.jak.ppke.hu.59891 > adsl240021.vnet.hu.domain: 18034 updateMA+$ [b2&3=0x6f6d] [24937a] [8301q] [27757n] [24942au][|domain] (frag 30812:24@0+)`

  `17:51:42.156089 lex.jak.ppke.hu > adsl240021.vnet.hu: udp (frag 30812:1456@24+)`

Egyes alkalmazások még tovább mennek: 512 byte-os csomagok : TFTP, BOOTP stb.

UDP szerver kérdések

- UDP-nél is lehet bizonyos értelemben kapcsolatról beszélni
- A forrás/cél IP cím/port négyes azonosítja
- Eszerint demultiplexál az UDP réteg
- Ezért tud egy UDP szerver több klienst kiszolgálni

Több interfész, több IP cím

- Egy gépnek több IP címe lehet
  - Akár egyetlen fizikai interfészen is
- Alapértelmezésben minden IP címen figyelnek a szerver programok
- Az operációs rendszer módot adhat arra, hogy csak egy-egy IP címen figyeljenek
- Módot adhat arra is, hogy csak bizonyos IP címről/címekről fogadjanak el klienst

16. IP multicast ethernet hálózaton

Általános: Broadcast, Multicast

- TCP-nél, mivel az kapcsolat-orientált, nincs értelme
- Rendszerint UDP-vel használatos
- Az ethernet kártya rendszerint csak azokat a csomagokat adja fel, amik:
  - Az ő MAC címére érkeznek
  - Broadcast címre érkeznek
- Minden csomagot felad, ha *pormiscous* módba tesszük
- Lehet kérni, hogy bizonyos multicast címekre érkező csomagokat feladjon
  - Ethernet multicast címek: amiknek 1 byte-ja páratlan.
- A multicast csomagokkal csökkenteni lehet a hálózaton „felébresztett", feleslegesen zavart állomások számát

Demultiplexálás, szűrés: Az egyes rétegek a csomag tartalma alapján döntenek

- Eldobják a csomagot
- Van valaki, feljebb a hierarchiában, aki kéri tőlük
- Az UDP réteg az IP cím, cél port, és esetleg a forrás cím alapján dönt

Limited broadcast

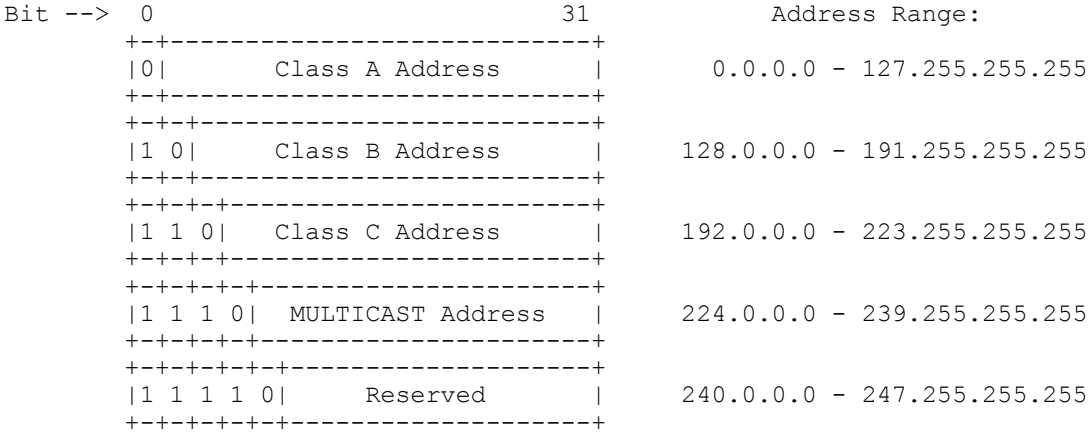
- 255.255.255.255 címre menő csomag
- Csak bekapcsoláskor használják pl. BOOTP/DHCP induláskor
- Sosem forwardolják a routerek

Multicast indokai

- Broadcast feleslegesen terheli a hálózati eszközöket
- Nem egyszer éppen a *legnagyobb* forgalmat bonyolító protokollok küldenek több címzettnek: pl. live audio/video
- Szükség van arra, hogy routereken át haladjon a forgalom

IP multicast - ethernet multicast címek

- A IANA az IEEE-től kapott egy mulitcast mezőt: 00:00:5e:00:00:00 - 00:00:5e:ff:ff:ff
- Ennek felső felét 00:00:5e:01:00:00 -től IP multicastra használjuk: ez *23 bit*
- Az IP címeknél multicast tartomány:



- Azt is mondják, hogy a 224.0.0.0-239.255.255.255 címek D osztályúak
- Már láttunk alkalmazást:
    - 224.0.0.1: all hosts
    - 224.0.0.3: all routers
- Itt *28 bit* áll rendelkezésre
- Mappelés: az IP multicast felső 5 bitjét nem vesszük figyelembe
    - Következmény: az IP rétegnek is figyelni, szűrni kell

Csatlakozás egy multicast csoporthoz / Csoport elhagyás

- Az alkalmazás utasítja az IP réteget, az IP réteg ethernet drivert:
    - Kér/lemond egy csoportot
    - Egy bizonyos interfészen !
    - Egy hoston, egy interfészen több alkalmazás is kérheti ugyanazt !

Küldés: Az alkalmazás a megfelelő IP/ethernet multicast címre küld

17. IGMP

Bevezetés:

Az egyik legrégebbi és sokáig egyetlen a multicast-tal foglalkozó Internet szabvány, megvalósítása mára kötelező az állomásokban, az IPv6-ban pedig az ICMP szerves része. Nem broadcast link-ek esetén a multicast úgy valósítható meg, hogy minden a csoporthoz tartozó állomás számára egyesével elküldjük a multicast csomagot. Broadcast link-ek esetén azonban fontos, hogy a csomag csak egyszer haladjon végig a link-en. Az állomásoknak itt fel kell készülniük arra, hogy meghallják az összes D osztályú címmel feladott csomagot és kiszűrjék azokat, melyek nekik nem szólnak. Valamilyen módon azonban a router-ek tudomására kell hozni, hogy az adott link-en van-e valaki, aki tagja valamely multicast csoportnak, hogy azok egyáltalán körbeadják itt az annak a csoportnak szóló csomagokat. Erre való az IGMP.

Két IGMP üzenet létezik, az egyik segítségével a router-ek kérdezik le a csoporttagságot, a másikkal pedig az állomások válaszolnak, egy-egy választ küldve minden csoporthoz, melynek tagjai. Minthogy broadcast link-en mindegy, hogy hány tagja van az adott csoportnak, csak az fontos, hogy van-e vagy nincs, ha valamely állomás hallja, hogy más már jelezte tagságát a router-nek, egy olyan csoportra, aminek ő is tagja, akkor ő már nem jelez külön.

Előadásból:

- Nem a multicast üzenetek küldése, hanem „szervezés", vezérlés a feladata
- Akkor van szerepe, ha nem csak egy LAN-on akarunk multicast forgalmat
- A routereknek tudni kell, hogy melyik interfészükre milyen multicast forgalmat kell továbbítani
- Külön IP protocol ID: 2
- RFC1112 - IGMPv1, STD 5 része

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Type  |    Unused     |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Group Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- RFC2236 - IGMPv2

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     | Max Resp Time |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Group Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Üzenet típusok:
  - 0x11 = Membership Query
    - General Qeury
    - Group-Specific Query
  - 0x16 = Version 2 Membership Report
  - 0x17 = Leave Group
  - 0x12 = Version 1 Membership Report
  - Érdekes, hogy szigorú értelemben a *v2* üzenetek speciális *v1* üzenetek!
- Max response time: a query üzenetekben ennyi ideig vár a válaszra a router. Mértékegység 1/10 másodperc
- Group Address: ha 0, a minden csoportot jelent, general query

- Csekkszumma: a szokásos one's complement összeadás
- Milyen IP címre kell küldeni az IGMP üzeneteket?
    - A query üzeneteket az all-host multicast csoportba: 224.0.0.1
        - Nem igényel belépést/kilépést
    - Más üzeneteket *abba* a csoportba, ahova az tartozik
- Ha egy host csatlakozik egy csoporhoz egy interfészen, akkor küld egy megfelelő riportot
- A routerek időről-időre érdeklődő (general queey) üzeneteket küldenek minden interfészükön
    - A 224.0.0.1 csoportba
    - Az állomások véletlen ideig várnak, és aztán küldenek egy riportot
        - 1. következmény: csökken a kollízió esélye
        - 2. következmény: törölhetik a várakozó riportot, ha látják, hogy ezen a hálózaton már van más partner a multicast csoportban
        - Minden csoportról külön riport megy, külön időzítéssel
- IGMPv1 szerint ‚leave' üzenetet nem küld, ha már egy processze se figyeli az interfészen a csoportot, csak a következő érdeklődésre nem felel
- Switch-ek és multicast
    - Lehet, hogy a switch a multicast csomagokat minden interfészére kikürtöli
    - Lehet, hogy hallgatózik, és megtanulja, hogy hol milyen csoportok vannak
        - Ez az ethernet címek szintjén is lehetséges
        - A routereket, és a mögöttül levő hálózatokat így el lehet veszíteni
    - Lehet, hogy a rendszergazda erővel beállít egy működési módot a switchen
    - A routerek segíthetnek a switchnek, a megfelelő információ elküldésével: CGMP, Cisco Group Management Protocol

IGMPv3

- Vegyük észre, hogy *küldeni* bárki bármikor küldhet multicast csoportba
- Multicast spam elleni védekezés
- Meg lehet adni, hogy honnan akarok forgalmat elfogadni

RFC 3376

18. DNS működés

Nem IP címekkel azonosítjuk a gépeket, hanem nevekkel. DNS: Domain Name System/Service szolgál a megfeleltetésre. A DNS nevek hierarchikus struktúrában vannak. Minden csúcsnak 63 karakter hosszú azonosítója van.

A tartománynév csúcsok listája egymástól ponttal elválasztva. Az első szintű (csúcs) tartományokat 3 csoportra osztjuk:

1.) arpa speciális tartomány az Ipcím tartománynév átalakításához.

2.) 3 karakterből álló tartományok az általános tartományok. 7 db van:

- com (üzleti szervezetek)
- edu (oktatási intézmények)
- gov (USA kormányzati szervek, más szervezetek)
- int (nemzetközi szervezetek)
- mil (USA katonai szervek)
- net (hálózatok)

3.) 2 karakterből álló tartományok az ISO szebvány által meghatározott országkódok. Ez az ország, vagy földrajzi tartomány

Fontos eleme a tartományneveknek a zóna, ami a DNS-fa egy önállóan adminisztrált részfája. Ilyen például egy intézmény tanszékekre való osztása. (itk.ppke, btk.ppke)

Ha egy zónát megjelölünk, akkor a zónáért felelős rendszeradminisztrátor névszolgáltatókat (name server) állít üzembe az adott zónára vonatkozóan. Kétféle névszolgáltató van: elsődleges és másodlagos. Ezek függetlenül működnek egymástól, így meghibásodás esetén nem lesz baj. Az elsődleges lemezről olvassa be az információkat, a másodlagos pedig az elsődlegestől kapja. Ez a zónatranszfer.

A névszolgáltató a beérkezett kérést megpróbálja feloldani. Ha nem sikerül, akkor továbbküldi egy másikhoz. Így előbb-utóbb megtalálják a kívánt címet.

Az ékezetes domain nevek problémásak. Ráadásul nincs is sok értelmük.

Cache, TTL

- A DNS szerverek a megtudott nevekre emlékeznek
- Haszon
    - gyorsabb feloldás
    - hálózat kímélése
- Hogy meddig, azt a név gazdája dönti el
- Minden rekordhoz tartozik egy ilyen idő: TTL (Time To Live).

DNS cache poisoning

- Az authority vagy az additional szekcióban visszaadott hamis érték
- Visszaélésre ad módot
    - Pl. a www.optbank.hu kéréseket máshova téríthetik
- Védekezés: csak az autoritatív név szerverektől szabad elfogadni adatot

Primary (master), secondary (slave) szerverek

- Egy zónát több szerver is szolgáltat(hat)
- Mind autoritás
- Egy primary, a többi secondary
- Nem összekeverendő a resolvernél használt primary/secondary fogalommal!

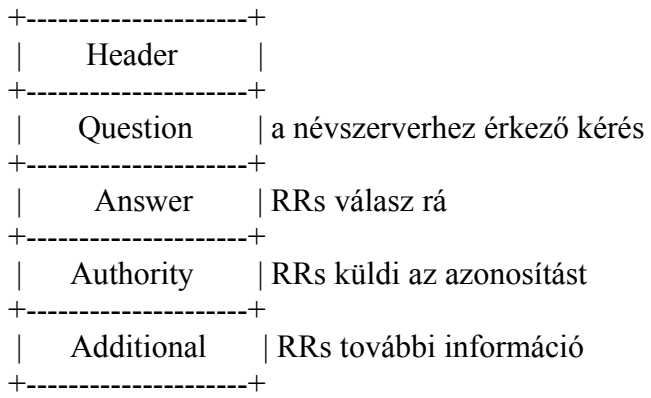
- A secondaryk időről időre tükrözik a primary adatait
    - csak akkor töltenek le adatot, ha van változás
    - konfigurálciós (SOA rekordban eldöntött) paraméterek

Idő értékek a SOA rekordban

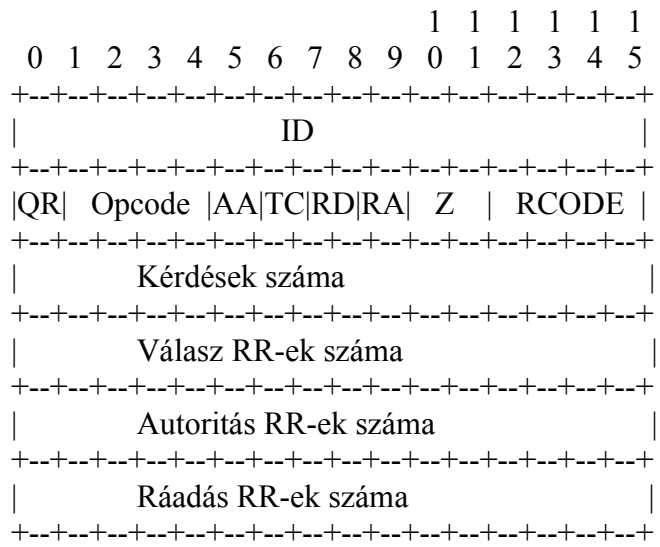
- >8.x Bind-nál W (hét), D (nap), H (óra) is megadható
- Refresh
    - ennyi időnként néz a secondary a primaryra
    - 8.x Bind-nál azonnali letöltés: notify
- Retry
    - Ha nem sikerül ennyi idő múlva próbálkozik a secondary
- Expiration
    - ennyi ideig tartja az adatokat a secondary, ha nem talál a primary-val kapcsolatot
- Nem megfelelő értékek
    - Értelmetlen
    - refresh > expire
    - retry > refresh
    - ttl > expire

  Célszerűtlen érték
    - kicsi TTL (De változás előtt!!)
    - kicsi vagy túl nagy refresh
- TTL a SOA rekordban
    - A 8.2 változatú Bind-ig: default
    - 8.2 óta: minimum
    - 8.2 -nél a default-ra új derektíva: $TTL
    - A TTL-t rekordonként később felülbírálhatjuk

19. DNS rekordok

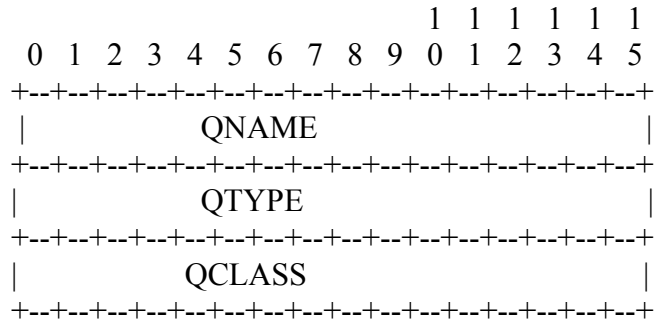


A header rész:



ID: ennek segítségével lehet a kérdést és a választ párosítani
QR: 0 ha kérdés, 1 ha válasz
AA: A válasz autoritatív (Authoritative Answer)
TC: A válasz csonkolt (Truncated)
RD: Rekurziót kérek (Recursion Desired)
RA: Rekurziót adok (Recursion Available)
Rcode: a visszatérési érték: ha 0, siker

Kérdés formátum:



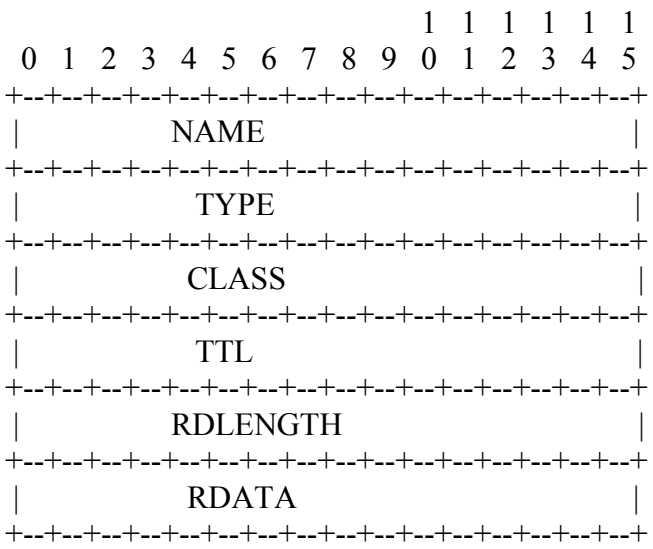
QNAME: a kérdéses domain name. Label-enként. A label-ek maximális hossza: 63
A 0 hosszú label a gyökér domaint jelenti
QTYPE: a kérdés típusa. Például

- o A: Address record
- o NS: DNS rekord
- o AXFR: zóna kérés

QCLASS: majdnem mindig 1, azaz IN, Internet Class

|7|d|i|g|i|t|u|s|3|i|t|k|4|p|p|k|e|2|h|u|0|

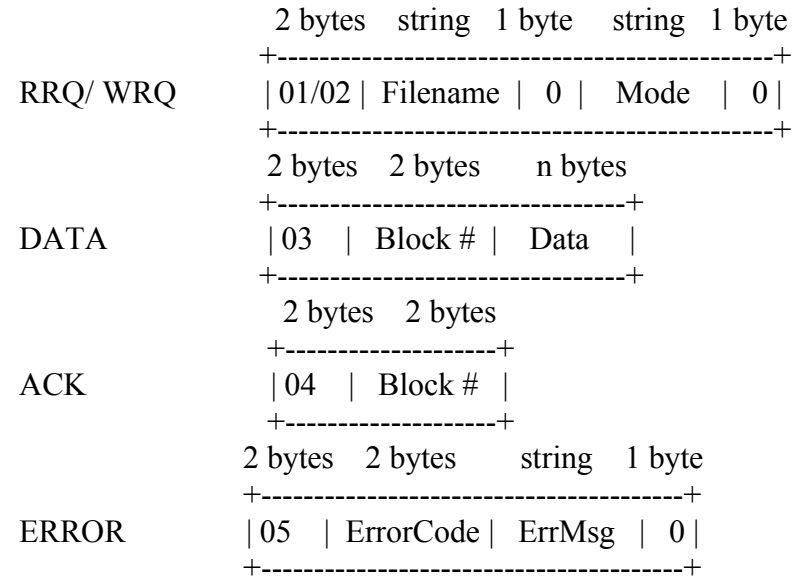
RR, Resource Record formátum



NAME, TYPE, CLASS: mint a kérdésnél
TTL: Time To Live. Ennyi másodpercig kell a cache-ben tartani a rekordot
RDLENGTH: a rekordhoz tartozó adat hossza byte-ban
RDATA: a rekordhoz tartozó adat. Formátuma függ a rekord típusától.

19. TFTP (Trivial File Transfer Protocol)

Egyszerűen implementálható fájl küldés/fogadás, amit megvalósít a protokoll. Boot szervereknél használatos: pl. a suli PC laborban.
UDP-t használ, 19-es porton szólítja meg a kliens a szervert. A tényleges adatforgalom már nem a 19-es portról, hanem a szerver véletlen portja és a kliens kezdeményező portja közt.
Stop-and-wait elvű protokoll: Minden egyes küldött blokkra nyugtát vár
Ha nem jön adat timeout-ig, ismétli az utolsó nyugtát
Ha nem jön nyugta timeout-ig, ismétli az utolsó adatot



Az első két byte: opcode, read=1, write=2, data=3, ack=4, error=5
Írásnál és olvasásnál 0-val terminált fájlnév
Mode: netascii (a sorok CR/LF-ek közt vannak), vagy octet
Az adatokat és a nyugtákat a block nr. rendeli egymáshoz
Az adat legfeljebb 512 byte
A fájl végét az 512-nél rövidebb adatcsomag jelzi
Nincs csekkszumma

Bűvészinas szindróma (Sorceres's apprentice syndrome): A k-adik nyugta késik, de nem vész el, ezért a k-adik adatot újra küldi a küldő. Megérkezik a késett k-adik nyugta és a küldő küldi a k+1-edig adatot. Megérkezik az újraküldött k-adikra küldött nyugta, ezért a küldő küldi a k+1-edik adatot

Ettől kezdve minden csomagot kétszer fog küldeni

Biztonság: a TFTP protokollban nincsenek azonosítók és jelszavak. A szerver implementációkkal lehet némi biztosítékot elérni a hozzáférési jogokkal.

20. TCP Transmission Control Protocol

TCP átvitelvezérlő protokoll, ami az IP adatátviteli protokollal megbízható kapcsolat-orientált adatátvitelt biztosít. A réteges felépítés révén különböző operációs rendszereket használó gépek is tudnak egymással kommunikálni.

A rétegek feladatai:

1.) Kapcsolati réteg: a hálókártyát és az operációs rendszert jelenti, melyek kapcsolódnak az átviteli közeghez.

2.) Hálózati réteg: a csomagokat (szegmens) továbbítja a hálózaton. A csomagok célba juttatásáért felelős.

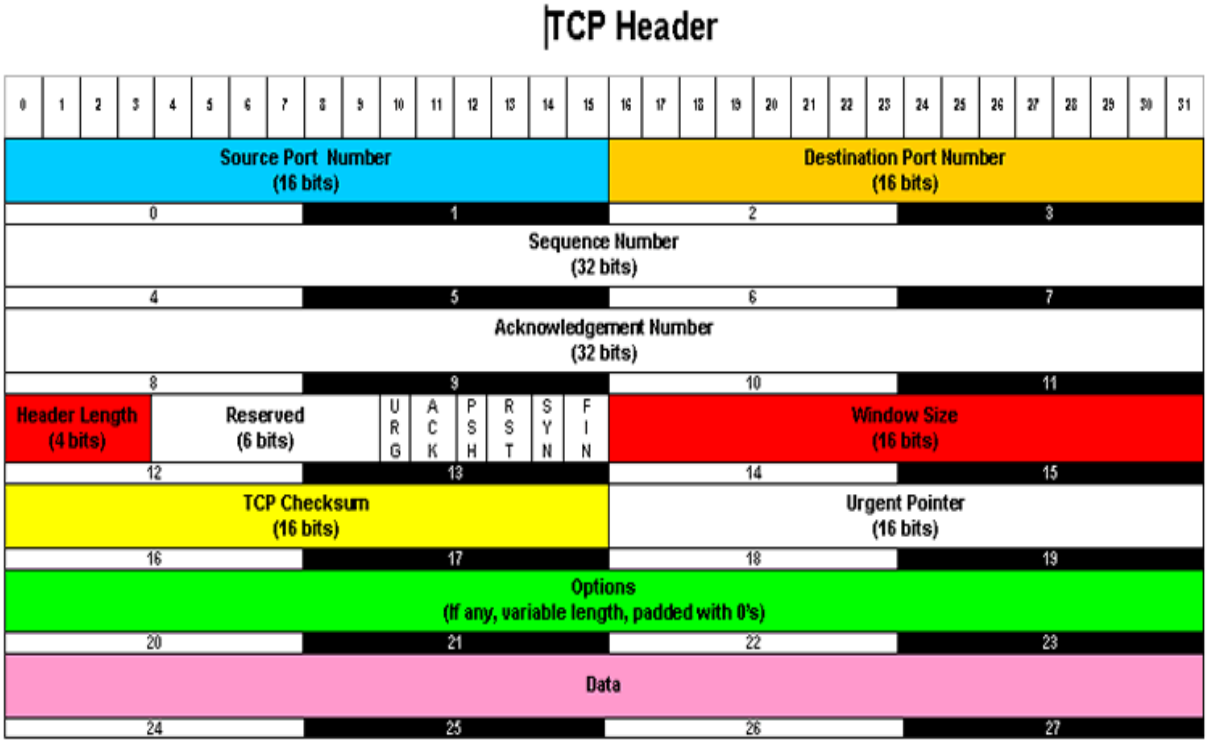
3.) Szállítási réteg: biztosítja a két kommunikáló gép közti kapcsolatot.

4.) Alkalmazási réteg: tartalmazza a távoli bejelentkezéshez, adatállományok átviteléhez és levelezéshez szükséges protokollokat.

A TCP feladata az üzenet-csomagok esetleges darabolása, azok rendbe- és összeállítása a túloldalon, valamint az esetlegesen elveszett rész-csomagok újraküldése, ezáltal két pont közt egy megbízható átviteli csatorna biztosítása. Az adatátvitel során (az OSI modellnek megfelelően) a két TCP szint "beszélget" egymással az alattuk elhelyezkedő réteget (pl. IP) felhasználásával. Az adatkapcsolat kezdetén a két TCP szint megegyezik egymással a legnagyobb, mindkét hálózat által egy egységként feldolgozható rész-csomag (chunk) méretében. A továbbiakban minden átvitelre kerülő üzenet-csomagot a küldő fél maximálisan ekkora méretű darabkákra "szabdal fel".

A csomag vétele után a fogadó TCP réteg egy nyugtázást (acknowledge) küld vissza a már fogadott rész-csomagok megjelölésével. Amennyiben a küldő fél megadott időn belül nem kap ilyen nyugtázást úgy a kérdéses csomagot újraküldi. Mivel nem effektív minden egyes csomag küldése után annak nyugtázását megvárni, ezért a TCP réteg egy ún. sliding window technikát alkalmaz. Ez azt jelenti, hogy egyszerre (az előzőek nyugtázása nélkül is) több csomagot is útjára bocsát a célállomás felé, majd a window limit (az egyszerre úton lévő csomagok maximális száma) elérésekor megvárja a nyugtázásokat.

A TCP a csomagok továbbításakor az alsóbb rétegek felé egy ún. TCP fejlécet illeszt azok elé. Ez a fejléc a küldő- és a célállomás IP címének megjelölése mellett a két pont közti kapcsolatot azonosító egyedi kapukat (port) és egy sorszámot is tartalmaz. A kapu-azonosítók a két azonos csomópont közti párhuzamos üzenet-átviteli csatornák megkülönböztetésére szolgálnak. A sorszám mező kapcsolat folytonosságának, azaz az esetlegesen meg nem kapott csomagok ill. az összeállítási sorrend megállapítására használhatóak. Ez a mező azonban nem a csomag sorszámát (1,2,3...stb.) hanem az adott csomag első byte-jának az üzeneten belüli elhelyezkedését adja meg. Így az 512 byte-os darabokban átvitt üzenetek csomagjainak sorszám mezői rendre a 0, 512, 1024 stb. értékeket fogják kapni. Ezen kívül a fejléc rendelkezik még egy ellenőrző-összeg (checksum) mezővel is, melynek segítségével az átvitel fizikai hibái is kiszűrésre kerülhetnek.

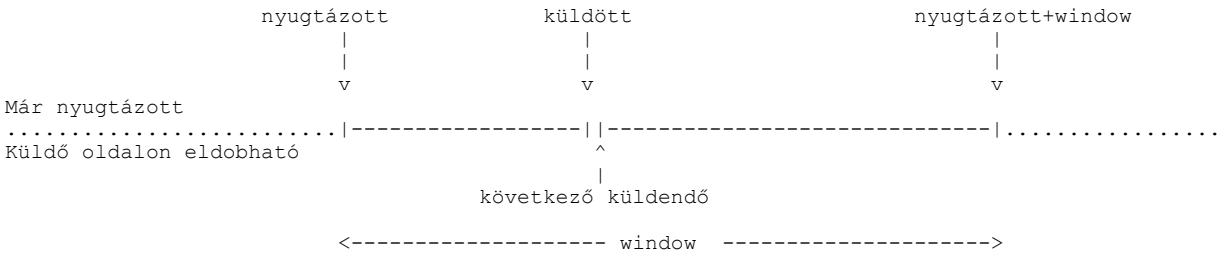


ACK: a nyugtázószám érvénye
SYN: a kapcsolat megnyitása
FIN: a kapcsolat lezárása
PSH: a PUSH funkciót jelzi
RST: súlyos hiba miatt a kapcsolat lezárása. Vétele után a címzett haladéktalanul lezárja a kapcsolatot és felszabadítja a számára lefoglalt erőforrásokat. Mivel a TCP-kapcsolat kétirányú (full duplex), az azonnali lezárás következtében az ellenkező irányú adatforgalom is megszakad, és adatot veszíthet. A TCP-kapcsolat normál lezárására a FIN jelzőbit szolgál
URG: Sürgős: a csomagban lévő adatot azonnal továbbítani kell az alkalmazásnak a vételpufferba való beillesztés nélkül

21. TCP flow control

Az állomások az ablak méretét tudják változtatni. A fogadott nyugták tartalmaznak az ablakra vonatkozó információt (window advertisment), amely lényegében a másik fél szabad puffer méretét tartalmazza. A küldő fél ennek alapján változtatja a saját ablakának méretét. Ha a fogadó pufferei kezdenek megtelni, kisebb ablakot jelölő közleményt küld a feladónak.
Tulajdonképpen változtatja az időegység alatt elküldött csomagszámot.

Sliding window



- A nyugtázott+window sorszámig folyamatosan küldhető adat
  - Rendszerint nem tölti ki az egész window-t
- A nyugtázott+window sorszámnál nagyobb sorszámú oktet nem küldhető
- A window folyamatosan csúszik jobbra az ábrán
- A window-t a fogadó csökkentheti/növelheti
  - Ha nyugtáz és nem akarja növelni, akkor a nyugtában kisebb window-t kell mondania az előzőnél
- A window mérettel a fogadó szabályozhatja a küldés ütemét
  - Flow control: az alkalmazás igényeihez igazodhat
    - Pl. a window lehet az alkalmazás által vételre felkínált buffer mérete
  - Congestion control: torlódás és csomagvesztés elkerülésére
- Egy nyugta mindig addig az *SN-ig* nyugtáz mindent, nem csak az utolsó csomagot
- A window nyílik, ha a jobb széle jobbra mozdul
- A window becsukódik:
  - ha a hirdetett window 0,
  - az adó kimerítette a küldhető adatmennyiséget,
  - a vevő mindent nyugtázott
- A window zsugorodik (shrinks), ha a jobb széle balra mozdul
  - Lehetséges, de ellenjavalt
  - Vannak protokollok, ahol ez lehetetlen
- Újraküldés
  - Ha három ugyanolyan ACK-t kapok, onnan kezdve újraküldök
  - Ha timeout (RTO) lejár, és nem kapok ACK-t

ACK generálás

Esemény	Akció
A nyugtázotthoz, és vett sorszámhoz képest folyamatosan jön egy szegmens	Késleltetés. Ne küldj ACK-t, még 200 ms-ig.
A vett sorszámhoz képest folyamatosan jön egy szegmens, van már nyugtáznivaló (timer ketyeg)	Azonnal küldj ACK-t, csúsztasd a window-t.
A vett sorszámmal összevetve kimaradást jelző sorszámmal jön csomag	Azonnal küldj ACK-t az *előző* csomagot nyugtázva (duplicate ack).
Egy lyukból eddig hiányzó, a lyuk aljához illeszkedő csomag érkezik	Azonnal küldj ACK-t.

Delayed ACK

- A TCP réteg nem feltétlenül küld nyugtát, ha tud
    - Legalább 40 byte-ot kell küldeni!
- Egy timer lejártát megvárja, hátha addig lesz:
    - elküldendő adat, amivel „mellékesen" nyugtáz: piggyback nyugta
    - újabb bejövő adat, amit nyugtázhat.

TCP nyugták

TCP Retransmission timeout (RTO): a nyugtázatlan csomag újraküldésének időzítése

- Ha túl rövid, felesleges újraküldés történik
- Ha túl hosszú, nagy csuklást okoz egy csomag elvesztése
- RTT - Rount Trip Time
    - TCP szegmens újraküldésnél ez határozza meg a timeout-ot
    - Egy szegmens elküldése és a hozzá tartozó ACK megérkezése közti idő
        - Időben változik, sok mindentől függ
        - Átlagot érdemes venni
- Exponential Backoff: a nyugtázatlan csomag újraküldésének idejét mindig duplázzuk, egy határig (jellemző érték: 9 perc)

Hogyan függ az RTO az RTT-től ?

- *Becsült-RTT* = ***a*** ** Becsült-RTT + (1-a) * Mért-RTT*, ahol $0<a<1$
- *RTO = b * Becsült-RTT*, ahol $b> 1$
- Tipikus értékek: $a = 0.9$, $b = 2$
- Ravaszságok
    - *a* nem konstans: ha hirtelen nő *Mért-RTT*, legyen kisebb: gyorsabban vegyük figyelembe a romlást
    - Karn algoritmus:
        - Ha ismételtünk egy csomagot, az arra jövő ack-ból ne számoljunk RTT-t
        - Nem tudhatjuk, hogy az ismételt csomagra jött, vagy az eredetire
        - Ha ismétlünk egy csomagot, az RTO-t ne módosítsuk (most dupláztuk exponential backoff miatt!), míg ismétlés nélkül nem csikarunk ki ack-t

Interaktív adatforgalom

- Kis csomagok
- Lehetőleg azonnali echo
- TOS: minimize delay
- Nagy a csomagokon az overhead: sokszor 1 byte-on 40, ráadásul a nyugta!

Nagle algoritmus

- Probléma: a lassú vonal végén ülő felhasználó gépelésével mindig újabb csomagokat generál
    - Ezzel tetézi a bajt
- RFC 896
- *Nem küldünk újabb csomagot, bufferelünk, míg van nyugtázatlan kinnlevő csomag*
- Timeout után mindenképpen ürítünk
- Önszabályozó: ha gyors a hálózat, nincs is hatása
- Egyes alkalmazásoknál hátrányt jelent
    - X terminálok
    - Több karakteres vezérlőszekvenciák (pl. PF gomb)
    - Ki lehet kapcsolni (klasszikus mondás: TCP_NODELAY)

Torlódás kezelése

- Számítani kell rá, hogy sok eszközön megy át a csomag, amiket túlterhelhetünk, csomagokat dobnak el
- Ha bambán újraküldünk: növeljük a bajt, a túlterheltséget
- Congestion control: mit csináljunk, ha torlódás lett
- Congestion avoidance: mit csináljunk, hogy elkerüljük a torlódást
- Összefoglaló: RFC 2581 TCP Congestion Control
    - Az egyik szerző R.W. Stevens
- Nem csak a technológia tanulságos, hanem a szemléletmód is: ahogy a DNS a szubszidiaritásra, a congestion avoidance a szolidaritásra szép példa

Slow start

- Nem csak a fogadó, a küldő is flow controlt alkalmaz
- **Congestion window**: a hálózat kímélése érdekében bevezetett ablak
    - A küldő becslése, a TCP masina belső változója: *cwnd*
    - A receive window és a congestion window jobb szélének minimuma határozza meg, hogy küldhetek-e csomagot
- Kezdetben a Congestion window 1 MSS méretű
- Minden ack-zott szegmens 1 MSS-sel növeli a *cwnd*-t
    - Ideális esetben a *cwnd* nem korlátoz
    - A vevő képességét, illetve a hálózati kapacitást teljesen kihasználva tömjük a hálózatot
- Slow start alatt az átvitel exponenciálisan nő
- A slow start alkalmazásával elkerüljük, hogy azonnal bajt okozzunk
- Ha torlódást észlelünk, a congestion avoidance algoritmus jut szóhoz

Congestion avoidance

- Congestion avoidance-nál a cwnd-t 1 szegmensnyivel növeljük minden RTT alkalmával
    - lineáris növekedés
- Additive increase, multiplicative decraese
- Bevezetünk egy új változót: slow start threshold, *ssthresh*
    - Kezdetben a receiver window
- Slow startot alkalmazunk, ha *cvnd* < *sshthresh*, congestion avoidance-t különben
- Ha congestion-t észlelünk (RTO, vagy tripla ack), akkor
    - Újraküldjük a szegmenst (fast retransmit)
    - A *nyugtázatlan_byte-ok/2*-re, de legfeljebb 2 MSS-re csökkentjük *ssthresh*-t (multiplicative decrease)
    - Ha RTO történt, akkor slow start-tal indítunk
    - Ha tripla ack, akkor cwnd = sshthresh + 3*MSS

Fast retransmit és fast recovery

- Ha a küldő tripla ACK-kat kap ugyanarra az sequence numberre, akkor congestion-ra következtet
- Nem várja ki az RTO-t, hanem újra küld: ez a fast retransmit
- Ez után nem kezd elölről a Slow Start szerint, hanem a congestion avoidance algoritmust alkalmazza: ez a fast recovery

RED: Random Early Detection

- A TCP congestion control eljárások egészségesebbé tették az internetet
    - A TCP kapcsolatok „reagálnak" a torlódásokra
    - Megszűnt a 80-as években előforduló „congestion collapse"
- A hálózat belsejében is lépéseket kell tenni:

        - Active queue management a routerekben
- RFC 2309: Internet Performance Recommendations
- Első megoldás
    - A router queue-kat kezel egyes interfészeihez
    - Ha a queue-ba már nem fér egy csomag, eldobja
    - Hátrányok:
        - Egyes kapcsolatok monopolizálhatják az erőforrásokat
        - Ha beáll egy telítettség, nehéz kivergődni belőle
        - Újabb lökések tovább rontják a helyzetet
- Arra kell törekedni, hogy tartósan kicsik legyenek a Q-k
    - Úgyis lesznek lökések, amiket el kell viselni
    - Interaktív kapcsolatoknál a hosszú Q-k kibírhatatlanok a nagy késleltetés miatt
- RED működés:
    - Minden csomagot bizonyos valószínűséggel eldobunk
    - A valószínűség annál nagyobb minél nagyobb volt az elmúlt időszakban a Q hossza
    - Egy-egy eldobott csomag nem okoz nagy gondot, de zsinórban eldobott sok csomag igen
    - Le tudunk lassítani minden adatfolyamot *flow*-t
    - Folytonosan mérjük az átlagos Q hosszt
        - A régebbi időket exponenciálisan kisebb súllyal vesszük figyelembe
        - A hossz mértékegysége csomag, de byte is lehet
    - Változók: minimum *(m)* és maximum *(M)* küszöb, $0 < maxp < 1$
    - Ha a Q mért hossza m-nél kisebb, nem dobunk el csomagot, ha M-nél nagyobb minden csomagot eldobunk
    - Ha a kettő közt van akkor a mért hosszzal arányos *0* és *maxp* közti valószínűséggel dobjuk el a csomagot
- WRED - Weighted RED: vannak „egyenlőbb" csomagok: valamilyen szempont szerint bizonyos csomagokat kevésbé valószínűen dobunk el
    - IP címek
    - Protokoll
    - TOS
    - ...

ECN - Explicit Congestion Notification

- RED-del együtt használt eljárás
- Kerüljük a csomagok eldobását, helyette szinezünk
- RFC 2481, RFC 3168
- TCP és IP, végberendezés és router aktív együttműködése
- Az IP és TCP fejrészben új flag-eket használ
    - ECT: ECN Capable Transmission, IP flag(ek)
    - CE: Congestion Experienced, IP flag(ek)
    - ECE ECN Echo, TCP flag
    - CWR: Congestion Window Reduced, TCP flag
- Pontosabban (v.ö. IP fejrész)

```
+-----+-----+
| ECN FIELD |
+-----+-----+
  ECT   CE         [Obsolete] RFC 2481 names for the ECN bits.
   0     0         Not-ECT
   0     1         ECT(1)
   1     0         ECT(0)
   1     1         CE
```

```
Figure 1: The ECN Field in IP.
```

- TCP fejrész, 13. és 14. byte

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               |               | C | E | U | A | P | R | S | F |
| Header Length |    Reserved   | W | C | R | C | S | S | Y | I |
|               |               | R | E | G | K | H | T | N | N |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

- TCP kapcsolat felvétel ECN-nel
  - A kezdeményező SYN csomagban áll ECE+CWR is
  - A SYN/ACK csomagban áll ECE
  - Ez után minden adatcsomagban beállítják az ECT bitet
    - Választhatják, hogy egyes csomagokban nem állítanak ECT-t
- ECN a routerben (middlebox-ban)
  - Ha a sor hossza *m* és *M* közt van, a véletlenszerű eldobás helyett beállítja az adatcsomagban a CE-t, feltéve, hogy a csomagban ECT áll
  - Ha a sor hossza *M*-nél nagyobb, eldob
- ECN a fogadó oldalon
  - Ha CE bites csomagot kap, ECN Echo-t küld
  - Minden ACK ECN echo lesz, míg szembe nem jön egy CWR adat
    - Az ECN echo csomag elveszhetett!
- ECN az adó oldalon
  - Ha ECN Echo csomagot kap, úgy tesz, mintha csomagvesztés által észlelte volna a torlódást
  - Csökkenti (felére) a *cwnd*-t
  - Csökkenti *sstsh*-t
  - A következő kimenő adatban beállítja a CWR bitet
  - A CWR bitet mindig beállítja, ha bármi okból csökkenti *cwnd*-t
- Kompatibilitási probléma
  - Az ECN csupa addig használatlan/rezervált bitet vesz igénybe
  - Egyes TCP stack implementációk nem tolerálják, ha ezek állnak
  - A linux 2.4 kernel bevezetésekor a web helyek túlnyomó része elérhetetlen volt az alapbeállítással, mert ECN-nel vette fel a kapcsolatot

Persist timer

- Ha a fogadó bufferei elfogytak bezárja a window-t
- Ha újra tud fogadni, nyitja: ACK megismételt acknoledgement numberrel, de nem 0 window-val
- Mi van, ha ez az ACK elvész?
  - A fogadó nem tudhatja, hogy van-e még a küldőnek mondandója
  - A küldő próbálkozik: küld egy 1 byte-os csomagot (hite szerint window-n kívül!)
    - Ezt ismétli persist timer-enként
    - A persist timer exponenciálisan nő (Exponential Backoff)
      - Tipikusan 1,5 sec-ről indul, 1 percnél nem lesz nagyobb

Silly window szindróma

- Ha a vevő oldalon kicsi (akár 1 byte) szabadul fel, lehet 1 byte a window
- A küldő 1 byte-ot küld, a vevő megint 1 byte-tal nyit é.i.t.
- Erőforráspocsékolás
- Elkerülésére
  - A vevő nem nyitja a window-t, csak akkor, ha MSS nagyságrendűt nyithat
  - A küldő nem küld, hacsak

        - MSS-nyit küldhet
        - Mindent küldhet, amit az alkalmazás kért
            - Feltéve, hogy Nagle ezt nem tiltja (nem vár ACK-t, és Nagle nincs kikapcsolva)

Keep-alive timer

- TCP kapcsolatok eredendően nem bomlanak forgalom hiányában sem
    - Akár hónapokig „élve" maradnak
    - Közben a middlebox-okat újraindíthatták, átkonfigurálhatták, akár kicserélhették
- Az eredeti szándék szerint az *alkalmazások* használhatnak „AYT" (Are You There) mechanizmust
- Mégis divatba jött a keep-alive mechanizmus
- RFC1122 leírja, bár ellenjavallva
    - Feleslegesen lebonthat később még használható kapcsolatokat
    - Feleslegesen hálózati erőforrásokat használ
    - Kidobott pénz, ha forgalom után fizetünk
- RFC1122 megengedi, hogy opcionálisan használni lehessen
- Keep-alive time: jellemzően 2 óra, általában rendszerparaméter
- Ha egy kapcsolat keep-alive ideig néma, akkor a mechanizmust alkalmazó oldal
    - Küld egy próba csomagot
        - Jellemzően 0 adattal, a küldött utolsó sorszámú SN-nel
        - Ez window-n kívül lesz!
- A másik oldal erre ACK-t küld

22. FTP - File Transfer Protocoll

Az FTP (File Transfer Protocol - Fájl Átviteli Protokoll) az egyik legrégebbi Internet-protokollok egyike - legelső változatát még 1971-ben dolgozták ki. A protokoll feladata a számítógépek közti fájl-csere biztosítása: a különböző platformok fájl-rendszereinek eltéréseit elrejtve, az állományok ellenőrzött és biztos átvitele az Internetre kapcsolt bármely két egység között.

A vezérlő-kapcsolat felépülése után a felhasználó különböző, alapvetően szöveges parancsokat adhat ki a távoli számítógép számára, amelyeket az értelmez és végrehajt. Ebben az üzemmódban minden adatcserét a felhasználó (kliens) kezdeményez, amelyre a szerver a parancsnak megfelelő üzenettel vagy hibakóddal válaszol.

Fájl-átvitel (küldés v. fogadás) kezdeményezése során azonban a parancs hatására a szerver egy második ún. adat-csatornát (data channel) nyit a kliens-számítógép felé, amelyen aztán az adatcsere bonyolítása történik. E csatorna - a vezérlő-kapcsolatban alkalmazottól eltérően - szigorúan csak bináris adatok átvitelére alkalmazható - az átvitelt és a kapcsolat további részeit befolyásoló parancsok továbbra is a vezérlő-kapcsolaton keresztül adhatók/adandók ki. Az átvitel befejeztével a szerver automatikusan lezárja (bontja) az adat-csatornát. Ez az architektúra több szempontból is előnyös: lehetőség van több átvitelre is és a sávszélesség jobban kihasználható.

Főbb parancsok:

TYPE: paranccsal az adatátvitel során a bájtok kódolásának módját határozhatjuk meg, azaz azt, hogy az eredeti gépen tárolt adatfolyamot a továbbítás (küldése) előtt hogyan alakítsa át a szerver, hogy az a kliens által értelmezhető formára kerüljen. A kódolási eljárást a parancs paramétereként megadott kulcsszó határozza meg, mely ASCII, EBCDIC, IMAGE ill. LOCAL.

STRU(CTURE): amellyel az átviendő adathalmaz belső szerkezetét határozhatjuk meg.

MODE: segítségével az adatfolyamban elhelyezendő információkat határozhatjuk meg.

RETRIEVE (RETR): file letöltésére utasít

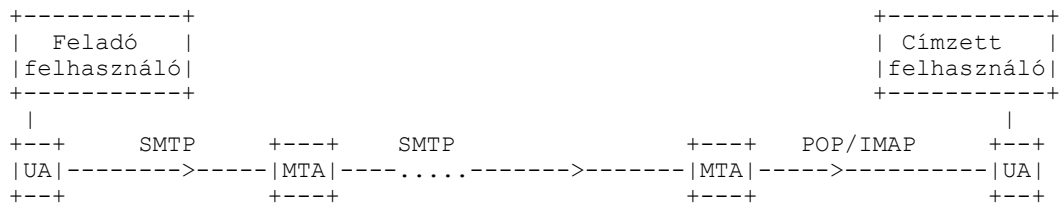
STORE (STOR): feltöltés fogadására utasítja a szervert

Minden FTP kapcsolat megnyitásakor először be kell lépnünk (login) a távoli gépen, azaz azonosítanunk kell magukat. Erre a USER (NAME) parancs használatával nyílik lehetőségünk. A felhasználó-név megadása után általában még egy ahhoz kapcsolódó jelszót is meg kell adnunk, amit a PASS(WORD) parancs kiadásával tehetünk meg. Érdemes megjegyeznünk, hogy a legtöbb publikus FTP-szerverre az "anonymous" felhasználónév és a saját e-mail címünk, mint jelszó megadásával léphetünk be. Ez utóbbi általában csak a felhasználó egyedi azonosítását teszi lehetővé (hiszen maga a user-név ez esetben nem egyedi), de gyakorlatilag bármilyen e-mail címre hasonlító szöveges információt elfogad, hiszen igazából nem állhat módjában az ellenőrizni.

23.SMTP - Simple Mail Transfer Protocol, ESMTP

Szintén klasszikus protokoll

- RFC821, újabban RFC2821 finomította
- User Agent, UA: a felhasználó által kezelt levelezőprogram
- Mail Transfer Agent, MTA: a leveleket továbbító szerver program



- Rendszerint több MTA-n át jut a levél a címzetthez
- Az UA->MTA és az MTA->MTA között rendszerint SMTP szállítja a levelet
- Az célnál rendszerint POP - Post Office Protocol, vagy IMAP - Internet Mail Access Protocol jut szerephez
- Az FTP-nél látott egysoros, CR/LF-fel záródó parancsok, ASCII szöveg
- Válaszok: `xyz valami` alakú válaszok
    - Az xyz számjegyeket a program nézi, a szöveg nekünk, embereknek szól
    - Az első számjegy jelenti a sikert/sikertelenséget, éppen úgy mint az FTP-nél
        - 1 = részleges, előzetes sikeres válasz
        - 2 = siker
        - 3 = részleges, közbülső állaptot jelző sikeres válasz
        - 4 = átmeneti sikertelenség
        - 5 = végleges sikertelenség
- Nincs külön adat csatorna: a levelet is ugyan az a TCP kapcsolat közvetíti, mint a parancsokat
- A felépülő full duplex kapcsolaton half duplex beszélgetést folytat az SMTP

Jellemző SMTP folyamat

- A szerver a 25-ös porton figyel
- A kliens egy ephemeral portról a szerver 25-ös portjára TCP kapcsolatot épít

Parancsok:

HELO: bejelentkezés levélküldéshez.
MAIL: feladó megadása
RCPT: címzett megadása
DATA: levél átvitele
QUIT: kapcsolat bontása
RSET: átvitel megszakítása, eredeti állapot visszaállítása
VRFY: fogadó címének ellenőrzése küldés nélkül
NOOP: üres művelet
Szövegek átvitelére találták ki. Nem támogatja más karakterkészletek és nem szöveges üzenetek átvitelét. Ezek más protokollok feladatai.

A levél tartalmaz fejrészt, melyben általános infókat tartalmaz: küldő, fogadó, téma, hogy érkezett, dátum, megjegyzés, másolatok címei, stb. Más protokollal történhet összeütközés a fejrész kódolását illetően (SMTP-ben ASCII formátumban van). Erre levelezési átjárókat (mail gateway) kell alkalmazni.

A levelező rendszerek gyakran továbbító ügynököt (relay agent) használnak. Ekkor kineveznek egy állomást, mely az összes kifelé menő levél kézbesítéséért felelős.

ESMTP - Extended SMTP

RFC 1425. Több parancsot használhatnak pl. 8 bites kódolással is küldhetnek adatot (ékezetes betűk miatt)

A HELO helyett EHLO parancsot küld a kliens, ezzel jelzi, hogy ESMTP-t ért

A 250-es válasszal a szerver felsorolja azokat a kiterjesztett tulajdonságokat, amiket támogat

24. Levél formátum: RFC2822/822, MIME

A levél formátum szintaxisát írja le. A levél ASCII sorokból áll, amiket CR/LF zár. A levél sorai legfeljebb 988 karakteresek. A levél fejrészből és törzsből (header & body) áll. A fejrész az első üres sorig tart.

RFC822/RFC2822 - levél formátum

- A levél formátum szintaxisát írja le
- A levél ASCII sorokból áll, amiket CR/LF zár
- A levél sorai legfeljebb 988 karakteresek
- A levél fejrészből és törzsből (header & body) áll
- A fejrész az első üres sorig tart
- A fejrész is tartalmaz(hat) küldőt és címzettet, de az a levél „postai" kezelésénél nem kap szerepet
    1. Ott az SMTP envelope számít
- A fejrész mezőket (header fields) tartalmaz
    1. A mező szerkezete: sor elején a név, kettőspont, tartalom
    2. Egy mező általában egy sorból áll
    3. Folytatósort a sor eleji whitespace jelez
- Date:
    1. A feladás idejét jelzi
    2. Példa: Tue, 7 Dec 2004 15:07:54 +0100 (CET)
    3. Fontos, hogy az időzóna is része
- To:
    1. Azok akiknek/amiknek elsősorban szánjuk a levelet
    2. valaki@valahol alakú cím
    3. Több is lehet, vesszővel elválasztva
    4. Kiegészíthető így: "Kiss Pista"
    5. Az UA-n kitöltött To-ból envelope címzett lesz
- CC:
    1. Carbon Copy, indigós másolat
    2. Épp olyan mint a To: envelope címzett lesz
- BCC:
    1. Blind Carbon Copy, titkos, rejtett másolat
    2. Épp olyan mint a To: envelope címzett lesz
    3. A DATA-val átküldött levélből kimarad
- From:
    1. Azt a küldőt jelenti, akinek a nevében megy a levél
- Sender:
    1. A személyt, vagy más entitást, aki ténylegesen küldi
        - From: sales@ceg.hu
        - Sender: kispista@ceg.hu
- Reply-to:
    1. Kérem erre a címre válaszolj
- Message-id:
    1. Egy véletlen szám, ami azonosítja a levelet
    2. Általában @ után a rendszert azonosítja, ahol a levél keletkezett
    3. Pl: Message-ID: <Pine.LNX.4.58.0412082029001.10200@login03.caesar.elte.hu>
- References:
    1. Message-id-k sorozatát tartalmazza, amikre hivatkozik a levél

2. News csoportoknál vezették be
3. Levelezési listáknál is hasznos: thread-ek keletkeznek

- Subject:
  1. Nagy illetlenség kitöltetlenül hagyni
  2. Rövid, és lényegre törő legyen
- Return-Path:
  1. Az envelope sendert mutatja
- Received:
  1. A közvetítő MTA-k által betett mező
  2. Segítségével nyomon lehet követni, hol és mikor járt a levél
  3. Ki lehet szűrni a körbe keringő leveleket
  4. Az egyes MTA-k hop count korlátozást használnak
     - Visszapattanó levelet küldenek ilyenkor
     - Itt nem a feladó, hanem a közbülső rendszer dönti el, hogy mikor elégeli meg!
  5. Az időzóna fontos része a received soroknak

MIME - Multipurpose Internet Mail Extensions

- RFC 2045
- Nem csak 7 bites ascii sorokat akarunk a levélben küldeni
- Úgy küldünk át ékezetes levelet, képet, hangot stb.
  - 7 bites ascii-val kódolunk
  - Fejrész mezőkben vezérlő információt küldünk hozzá
- Egy üzenet több részből állhat
- Minden résznek RFC822 szerinti fejrésze lesz
- Egy MIME üzenet újabb MIME üzeneteket tartalmazhat fa elrendezésben
- Új fejrész mezők
  - MIME-version:
  - Content-type:
    - A törzs típusát mondja meg
    - Leggyakoribb a text típus
    - Paraméter: charset
      - text/plain; charset=ISO-8859-2;
      - text/html
    - multipart/mixed, multipart/alternative
      - Paraméter: boundary
        - Pl. Content-type: multipart/mixed boundary="ezittaz"
        - Az egyes darabokat ilyen sorok választják el:
          - --ezittaz
        - A utolsó darabot ez zárja le:
          - --ezittaz--
      - multipart/mixed keletkezik akkor, ha csatolmányt küldünk egy levéllel
      - Az egyes darabok külön RFC822 szerinti fejrészt tartalmaznak
  - Content-transfer-encoding:
    - 7bit - közönséges ASCII levél, sorokra tördelve
    - 8bit - nem csak 7 bites karakterek, de sorokra tördelve
      - Így lehet tárolni pl. lokális folderben a leveleket
      - SMTP szerverek is átvehetik így: 8bitmime kiterjesztés
    - quoted printable: a 7 bites ascii karakterek változatlanok maradnak, a többit 3 karakteres szekvencia kódolja. Pl. 0xe4-et így: =e4

        - Az egyenlőségjelet is kódolni kell: =3d
    - Base64
        - Az üzenetet 3 byte-os darabokra bontjuk
        - A keletkező 24 bitet 6 bites darabokra bontjuk
        - A 4 darabot egy táblázat szerint kódoljuk (64 jel)
            - Ez a táblázat betüket, számokat és + /-t tartalmaz
        - Visszakódolásnál a karakter indexe szerint összeállítjuk a 3 byte-os darabokat

ESMTP - Extended SMTP

- RFC 1425
- Ha megállapodtak benne
    - Több parancsot használhatnak
    - Pl. 8 bites kódolással is küldhetnek adatot
- A HELO helyett EHLO parancsot küld a kliens, ezzel jelzi, hogy ESMTP-t ért
- A 250-es válasszal a szerver felsorolja azokat az kiterjesztett tulajdonságokat, amiket támogat
- Pl:

```
       ehlo sun.tuc.noao.edu
250-vangogh.CS.Berkeley.EDU Hello sun.tuc.noao.edu [140.252.1.29],
pleased to meet you
250-EXPN
250-8BITMIME
250 HELP
```