Számítógéphálózatok kidolgozott vizsgatételek

Leichner Dávid

2015. január 4.

A kidolgozás szerintem több, mint ami vizsgára kell. De a tételek tartalmazzák a villámkérdéseket is. Ezt lent részletezem. Kellemes felkészülést, és sikeres vizsgát!

1. OSI referenciamodell és TCP/IP

ISO/OSI referenciamodell

- A 70-es évektől
- ISO = International Organisation for Standardization
- OSI = Open System Interconnection

Rétegek

- Egymásra épülő elemek valósítják meg a hálózat működését
- Minden szinten létezik egy protokoll, ami szerint kommunikál
- Az alatta levő réteg szolgáltatásait **igénybe veszi**, a felette levő réteg számára szolgáltatást **nyújt**
- Az egymásra épülő rétegek protokoll stacket alkotnak
- Példák:
  - Ethernet alatt a fizikai réteg lehet réz, vagy üveg
  - TCP felett az alkalmazás lehet SMTP, vagy SSH
  - Ethernet és SMTP között lehet IP, vagy Decnet

A rétegek a következők:

1. Alkalmazási
2. Prezentációs
3. Session
4. Transzport
5. Hálózati
6. Adatkapcsolati
7. Fizikai

- A TCP/IP-ig nem volt gyártófüggetlen implementáció
- Ilyen gyártófüggő megoldások:

gyártó	**hálózat**
IBM	SNA
Digital	Decnet
Siemens	Transdata

A TCP/IP nyílt rendszer (Open System):

- A megvalósításhoz szükséges szabványok szabadon hozzáférhetőek
- Az implementációk jórészt szabadon hozzáférhetőek, sőt továbbfejleszthetők
- Az szabványosítási folyamatba bárki bekapcsolódhat
- Etalon implementáció: BSD (Berkeley Software Distribution) unix
  - Klasszikus szabad szoftver
  - Szabadság = ingyenesség (jelentheti, hogy a forráskód szabadon megtekinthető és módosítható)

SMTP	HTTP, FTP	DNS	**Alkalmazási (application)**
TCP		UDP	**Szállítási (transport)**
IP			**Hálózati (network)**
Ethernet	PPP	AX.25	**Adatkapcsolati (link)**
réz, üveg	Telefon	Rádió	**Fizikai (physical)**

Az egyes rétegek egymástól nem függnek, cserélhetők, csak az alattuk és a felettük levő szolgáltatást ismerik.

Internet protokollok és programok

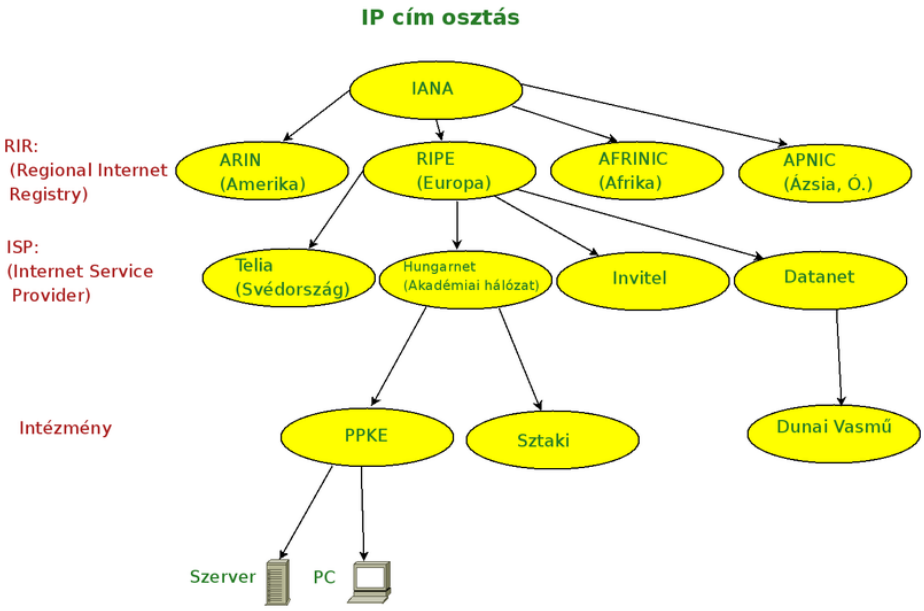
- Fizikai réteg: egy médiumon bitfolyamot visz át egy jel modulálásával
- Adatkapcsolati réteg: csomagokat különít el. Adatátviteli hibát vehet észre, ismételhet. A fizikai réteggel sokszor összenőtt: pl. ethernet kártya
- Hálózati réteg: hálózati címek szerint irányítja a csomagokat
- IP: A végberendezések fele irányít egy-egy csomagot. A végberendezéseket címek azonosítják. Az egyes csomagok egymástól nem függenek. Internet Protocol.
- ICMP: hiba, és szolgáltati üzenetek küldésére szolgál. Internet Control Message Protocol.
- IGMP: Multicast üzenetek küldésénél használatos. Egyetlen csomagot egyszerre több végberendezéshez is eljuttathatunk. Internet Group Management Protocol
- Szállítási réteg: két végberendezés közt kapcsolatot építhet, bonthat, az alkalmazások számára „telefonkapcsolatot” ad.

- TCP: hiba és veszteségmentes, sorrendtartó átvitelt biztosít. Transport Control Protocol
- UDP: nem garantál hiba és veszteségmentességet, sorrendtartást.
- Alkalmazási réteg:
  - Levelezés: SMTP
  - Telnet, SSH, FTP, HTTP, SCP

2. RFC-k, Internet szervezetek

Ip címek kiosztása

- A föld régiókra van osztva, ezek IP tartományokat adnak
- Európai szervezet: RIPE
- A RIPE szolgáltatóknak ad tovább kisebb tartományokat
  - Az akadémiai hálózatban a Hungarnet a szolgáltató
  - Más szolgáltatók például:
    - Datanet
    - Invitel
- A szolgáltatók egyes szervezeteknek
- Az egyes szervezetekben a hálózati rendszergazdák adnak címeket
  - Rejtett címeket is adhatnak, amik az interneten nem jelennek meg
  - Ezeket átfordíthatják: Network Address Translation (NAT)



Akadémiai hálózat - NREN (National Research and Education Network)

- Felsőoktatási intézmények, közgyűjtemények, kutatóintézetek hálózata illetve szervezete
- Kezdetektől fogva élenjárók az internet használatában, fejleszétésében
- Például:
  - Szervezet: Európában Terena, Magyarországon a Hungarnet, az NIIF Intézet
  - Hálózat: Geant, Magyarországon HBONE

RFC-k

- Request For Comments, a szerény, gyakorlatias hozzáállást mutatja
- Az internet közösség alapdokumentumai
- A protokollok szabványai RFC-k
- Az IETF (Internet Engeeniring Task Force) alkotja meg
- Munkacsoportok (Working Groups)

3. Klasszikus ethernet

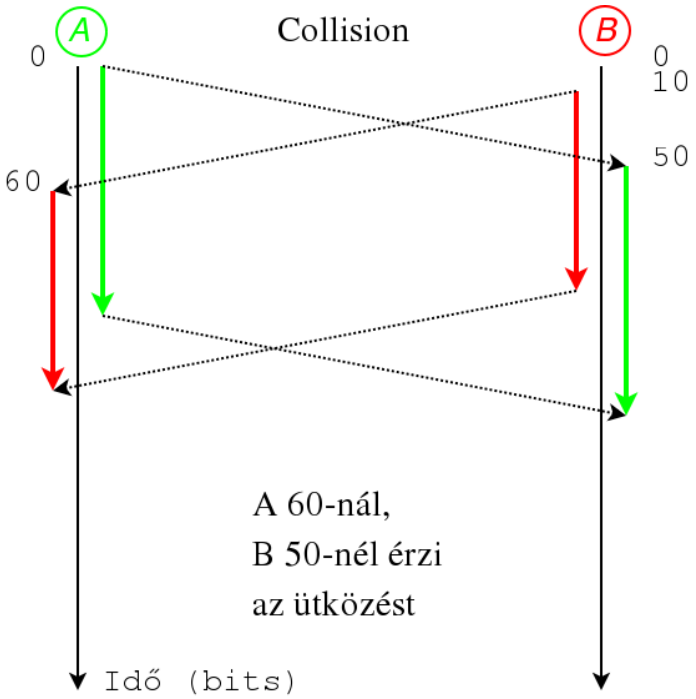
A leggyakrabban használt 2. szintű protokoll

Ether: a közvetítő médium, éter

- Az 1970-es évek elején Xerox Palo Alto Research Center: 2,94 Mbit/s
- 1979-82: Digital, Intel, Xerox (DIX ethernet), Ethernet II, 10 Mbit/s
- 1985, IEEE (Institute of Electrical and Electronic Engineers) szabvány: 802.3
  - 802 working group (1980, februárban indult)
  - Eltér az ethernet II-től
  - Egy hálózaton párhuzamosan többféle frame formátum használható
  - A Novell Ethernet-802.2 és Ethernet-SNAP a 802.3-tól is eltér
- Azóta folyamatosan egyre újabb szabványokat adnak ki az IEEE 802 munkacsoportban
  - Gigabit, 10Gbit, 40Gbit, 100Gbit ethernet
  - Wireless ethernet

CSMA/CD

- CS = Carrier Sense
  - Az egyes állomások "érzik" ha a csatorna foglalt
  - Nem kezdek adni, ha már más ad
- MA = Multiple Access
  - A másnak szóló csomagokat elengedem a fülem mellett
  - "Snoop", jelentése: Fülelek és elkapom a másoknak szoló csomagokat is...
- CD = Collision Detection
  - Ha adás közben észreveszem, hogy ütközés (collision) van, más is ad, megállok, de előbb még 32 zajbitet kikürtölök
  - Véletlen ideig várok: **backoff delay**
  - Újra próbálkozok
  - Ha újra ütközés van, újra véletlen ideig várok, de van felső korlát: *exponential backoff*
  - 1 persistent
    - Ha szabad a csatorna, és van adnivalóm, 1 valószínűséggel adok



1. ábra. Ethernet ütközés

- Capture effect
  - Az exponenciális backoff következménye, ha túlterhelt a hálózat
  - Egyetlen állomás kisajátíthatja az erőforrásokat

Topológia

- Busztopológia
  - Thick ethernet (már nincs)
  - Thin ethernet (koax, BNC, néha)
- Csillag és fa toplógia
  - Manapság ilyenek készülnek
  - UTP és optika

Ethernet fizikai réteg

- 10Base2: vékony (thin) ethernet, koax, T alakú BNC csatlakozók (ritka)
- 10Base-T: 10Mb/s legalább Cat3 csavart érpáron (ez is elavult)

- 100Base-TX: 100Mb/s legalább Cat5 csavart érpáron
  - Ezzel találkozunk leggyakrabban
  - Az UTP kábel 8 eréből csak 4 használatos
    - 2-2 adásra/vételre: send=1,2/receive=3,6
    - Nem feltétlenül „tudja kitalálni” az állomás, hogy melyek ezek
      - lyenkor cross kábel-re van szükség
- 1000Base-T: 1000Mb/s legalább Cat5 csavart érpáron
  - Mind a 8 ér használatos
  - Mind a 4 pár ad és vesz is 250M-val!
  - Nincs szükség sose cross kábelre!
- 1000Base-TX: 1000Mb/s legalább Cat6 csavart érpáron (se ez se a Cat6 nem terjedt el)

Rézkábelek

- UTP - Unshielded Twisted Pair: a legelterjedtebb
- ScTP - Screened Twisted Pair: árnyékolás a csavart érpáron

Optikai kábelek

- Multimódusú: MMF - Multi-Mode Fiber
  - A fény több úton halad a kábelben
  - LED-del megvilágítható
  - 50-100 micron (!) a kábel átmérője
  - Legalább 3 csomagoló réteg borítja
  - Jellemző méretek (belső méret, első csomagolás):
    - 62,5/125 μ
    - 50/125 μ
  - Viszonylag olcsó
  - Kábelhossz: ¡ 2000m
  - jellemzően épületen belül
- Single módosú: SMF - Single-Mode Fiber
  - A fény nem szóródik a kábelben
  - 10 micron (!) a kábel átmérője
  - Nehezebb betalálni a lyukba (haha lézer)
  - Viszonylag drága
  - Nagyobb távolságokra hasznos

MAC (Media Access Control)

- Az ethernet egy rétege, funkcióhalmaza:
  - I/O
  - Címzés
  - Hiba detektálás (bit hiba (CRC), protokoll hiba, médium hibája pl: szakadás)

Frame

- Preambulum: 65 bit
- Start Frame Delimiter: 10101011
- Cél (destination) cím:
  - Unicast
  - Multicast
  - Broadcast: FFFFFFFFFFFF, minden állomásnak az etherneten
- Hossz vagy típus: ha 1500-nál kisebb, akkor hossz, ha nagyobb, akkor típus (ethernet II.)
  - A hossz (802.3) a következő byte-tól a crc-ig számít
  - Ethernet II-nél a hossz nincs a frame-ben!
- Pad: az adat legalább 46 byte legyen. Ha kisebb lenne, kiegészítjük.
- CRC: A cél címtől az adat vagy a pad végéig számolja a küldő és a fogadó is (hiba esetén eldobja)
- Frame-ek közt: interframe gap. Legalább 96 bit idő

Repeater, HUB

- Ethernet szegmenseket köt össze
- Fizikai szinten funkcionál
- A fizikai jelet megismétli minden interfészén
- Különböző típusú interfészei lehetnek (UTP, koax)
- Protokoll független
- Hibás frame-ket is átvisz

Switch, bridge

- Ethernet szegmenseket köt össze

- Buffereli a frame-eket
- Különböző sebességű interfészei lehetnek (100M, 1G)
- Megtanulja, hogy egyes interfészein milyen mac címek vannak
  - Amíg nem tud egy mac címet, addig az arra küldött csomagokat repeaterként továbbítja (IP-nél ez nem fordul elő a gyakorlatban)
  - Felejt. Azért kell tudni felejteni, különben egy idő után betelne a tárhelye, pl: meki
- Csak arra küldi tovább a frame-et, amerre kell
- Protokoll független
- Előnye a repeaterrel szemben
  - Kíméli a sávszélességet
  - Hibás, frame-et, ütközést nem közvetít
  - Lehallgatás ellen védelmet jelent
- A switch: célhardverként gyártott bridge
- Switch extra funkciói: managelhető, IP címe van, VLAN, stb.

Korlátozni kell a framek hosszát. Típusai: runt, giant.

Ethernet flow control

- 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
  - Egy újabb PAUSE az előzőt érvényteleníti
- A két állomás egymástól függetlenül lehet képes adni/fogadni PAUSE-t
- PAUSE alatt a másik állomás csak MAC control frame-et küldhet (pl. pause-t)

Link halmozás (aggregation): Két vagy több fizikai összekötettést logikailag egynek tekinthetünk

Ethernet autonegotiation

- Pulse-code szekvencia, nem frame (FLP = Fast Link Pulse)
- Point to point összeköttetéseknél (switchek vannak)

Sebesség	**Médium**	**CSMA/CD**	**Topológia**	**Flow control**
10mb	koax, utp	CSMA/CD	busz	nincs szükség flow control-ra
legalább 100 mb	utp, optika	rendszerint full duplex, nincs szükség CSMA/CD-re	csillag, fa	szükség lehet flow control-ra

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

PPPoE

- PPP over Ethernet, RFC2516
- ADSL és FTTH (Fiber To The Home) előfizetőknél ez használatos
- Ethernet type mező: discovery stage, 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 réteg:

HTTP
TCP
IP
PPP
PPPoE
Ethernet
ADSL
Telefonkábel

Fast ethernet

- IEEE 803.3u, 1995
- 100 Mbit/s

Full duplex ethernet

- IEEE 802.3x
- Csak pont-pont összeköttetéseken működik

- Nincs CSMA/CD, bármikor lehet adni
- Az átviteli kapacitás duplájára nő
- Nem kell ütközések miatt késleltetni
- Kábel hossz nőhet, slot time csökkenhet
- (Slot time: minimum frame hossz)

VLAN (Virtual Lan)

- 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-t 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 kapcsolatorientált
  - Az egyes csomagokat a hálózati réteg egymástól függetlenül kezeli
  - Nincs kapcsolat felépítés, kapcsolat bontás
- Best effort, decentralizált, 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
- A felsőbb rétegek gondoskod(hat)nak mebízható, kapcsolatorientált 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

A fejrész hossza 4 bit, mértékegysége 4 byte.

TOS

- RCF791 (precedence, tos, 0, 0)
- RFC 1122 (precedence, tos)
- RFC 1349 (precedence, tos, mbz)
- MBZ: Must Be Zero
- RFC 2474 (ds field, dscp, cu)
- CU: Currently Unused
- RFC3168 (field, dscp, ecn field)
- DSCP: differntiated services codepoint
- ECN: Explicit Congestion Notification
- A csomag hálózat belsejében való továbbítását befolyásoló mező
- Sokszor átdolgozták
- Nincs garancia arra, hogy a hálózati partnerek kezelik

- 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 (min 46 byte)

ID

- A csomagra jellemző egyedi szám
- Az internet kezdetén az implementációkban eggyel több mint a előző
  - Megjósolható → visszaélésre ad módot
  - Manapság véletlen szám
- Tűzfalak tovább randomizálhatják

Flag-ek

- Az IP darabolásánál (fragmentálás) van szerepük
- Don't Fragment (DF), More Fragments (MF)
- Fragmentum offszet
  - Ha fragmentált az IP datagram, azt mutatja, hogy ez a darab 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, végtelen ideig keringhetnének csomagok
- A traceroute program épp ezt használja

Protocol

- Az IP feletti protokoll adja meg

- IANA (Internet Assigned Numbers Authority) osztja

Csekkszumma

- Nem ethernet CRC, sima bit összeadás
  - RFC1071, RFC1624
  - One's complement összeadás
  - 0 ha minden jó
- Csak fejrészre
- A csomag többi részét a felsőbb szint ellenőrizheti
- Hop-ról hop-ra változik

Forrás és célcím

- Az IP réteg szempontjából a legfontosabb: ennek alapján továbbítja (routeolja) a csomagot
- Tűzfalak még ezt is módosíthatják
- Pontozott decimális jelölés
- A négy byte-ot decimális számként írjuk, közéjük ponttal, pl.: 193.239.149.4

Opciók

Ha ugyan...

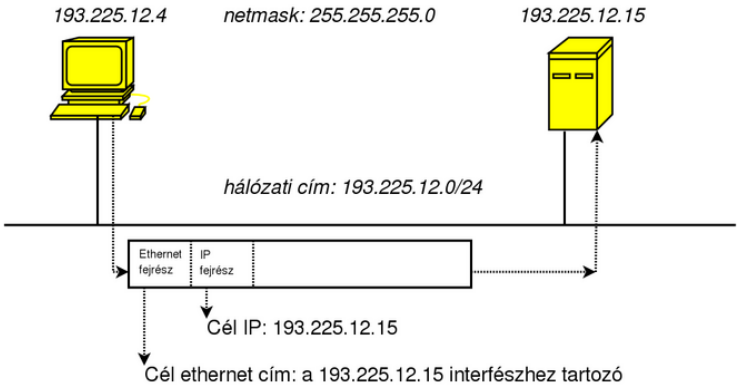
Routing

- Def. szerint a 3. (network) réteg feladata
- Végberendezésekre jellemző konfig.
  - Ha az adatkapcsolati rétegen közvetlenül elérhető a címzett: egyenesen neki
  - Ha nem, akkor default route (alapértelmezett átjáró)

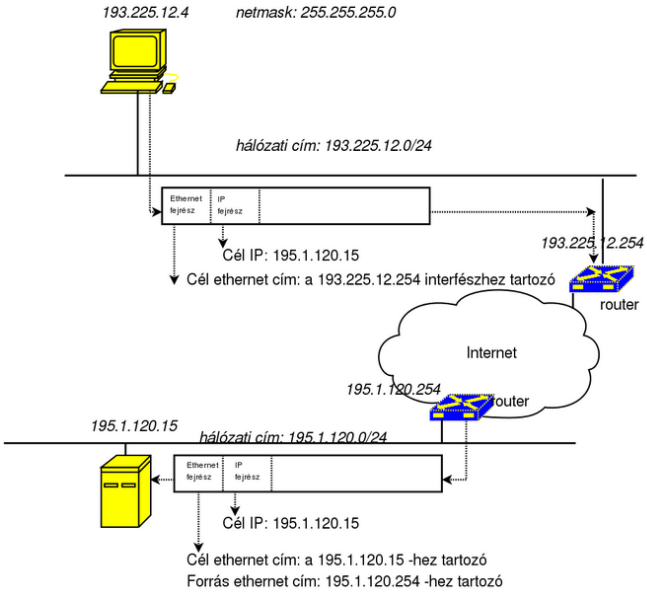
Netmask

- Egy IP címhez képest kijelöl egy tartományt
  - Decimális forma: 255.255.0.0
  - Hexadecimális forma: ffff0000
  - CIDR forma: /16
- A hálózaton kívüli címekre nem közvetlentül küldjük a csomagokat
- Router (gateway) címre
- Ezért a router-nek ebben a hálózatban kell lennie

Közvetlenül etherneten elérhető cím



Routerek közbeiktatása



Routing attribútomok

- Melyik IP címet, címtatartományt (netmask)
- Melyik interfészen
- Milyen forráscímmel
- Milyen link réteg tulajdonságokkal
- %ip route show parancsot nézzük meg...

- %netstat -r parancsot is...

Flagek

- G: gateway
- U: up, él
- H: host route, csak egy IP címre
- D: dinamikus, pl ICMP redirect által keletkezett
- M: modified, pl. ICMP redirect által

TCP-nél használt paraméterek

- MSS: Maximum Segnemt Size, MTU-nál kisebb is lehet
- Window: ennyi byte-ot fogad el nyugtázás nélkül
- irtt: Initial Round Trip Time, ennyi körbefordulási időt feltételez

NAT: Network Address Translation

- A routeren/tűzfalon áthaladó csomag IP címét kicserélik
- Táblázatot kell fenntartani az élő kapcsolatokról
- A TCP/UDP portokat is módosítani kell, hogy ne legyen két „azonos” kapcsolat
- A NAT-olt hálózat bizonyos értelemben rejtve van a világ elől
- Nagyban megnöveli például az egyetlen C osztállyal internetbe kapcsolható gépek számát
- Egyetlen IP cím mögött akár 100 gépünk is lehet
- A NAT mögött használatos címek számára külön tartományokat tartanak fenn: RFC1918

6. PPP, PPPoE

PPP - Point to Point Protocol

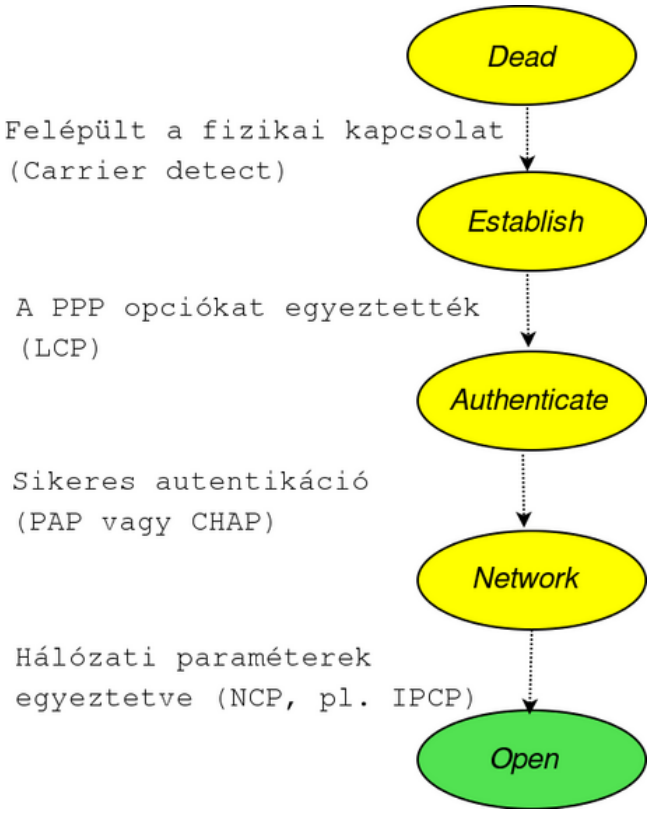
- RFC1661: pont-pont kapcsolatra használatos adatkapcsolati réteg
- IP alatt gyakran használt, másra is használható
  - Az első hasonló protokoll a SLIP volt (Serial Line IP, RFC1055)
- Klasszikus esetben telefon vonalon
- Feladatok — ezeket tűzték ki, amikor tervezték a protokollt (RFC1547)
  - http://tools.ietf.org/html/1547
  - Transzparancia: tetszőleges byte folyamot át lehessen vinni
  - Hatékony sávszélesség kihasználás
  - Hiba felderítés (CRC)
  - Többfajta protokollt támogasson (ne csak IP-t), ezeket multiplexálni lehessen
  - Magasabb protokoll paraméterek egyeztetését oldja meg (pl. IP címek)
  - Többféle fizikai médiumon működhessen

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

- Flag: a frame kezdetét és végét jelzi: 0x7E
- Address: PPP-nél mindig 0xFF
- Control: PPP-nél mindig 0x03
- Protocol: ethernet „type”-hoz hasonló
  - LCP - Link Control Protocol
  - PAP, CHAP - Password Authentication Protocol, Challange Handshake Authentication Protocol
  - NCP - Network Control Protocol, pl. IPCP
  - Adat - pl. IP
- Adat
  - Általában legfeljebb 1500 byte. LCP-vel másban is megállapodhatnak
  - Az adatfolyamban a flag byte-ot escape-elni kell: byte stuff

LCP: Ling Control Protocol

- Kapcsolat felépítés / bontása



2. ábra. PPP állapotok

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

PAP: Password Authentication Protocol

- Két lépés
  - Kliens küldi az azonosítót és a jelszót (kódolatlanul)
  - Szerver válaszol, hogy rendben van-e
- Nem biztonságos (lehallgatható, visszajátszható

CHAP: Challange Handshake Authentication Protocol

- RFC1994
- Három lépés
- A szerver egy véletlen számot küld a kliensnek (challange)
- A kliens a kapott challange-ből, és a jelszóból választ képez
- A szerver ellenőrzi
- Nem megy cleartext-ben a jelszó, nem visszajátszható a folyamat

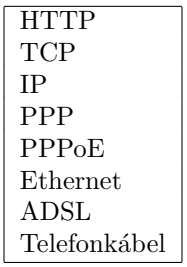
NCP: Network Control Protocol

- A magasabb réteg konfigurálására szolgál, több van (pl. külön Novell)
- Egy időben akár több magasabb szintű kapcsolat lehet egy PPP session fölött
- IP: IPCP, IP Control Protocol, RFC1332
  - IP cím
  - TCP/IP fejrész tömörítés
  - DNS szerverek címe RFC1877

PPPoE

- PPP over Ethernet, RFC2516
- ADSL és FTTH (Fiber To The Home) előfizetőknél ez használatos
- Ethernet type mező: discovery stage, 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

Több mint 7 réteg:



Loopback interfész

- Sokszor a kliens és a szerver ugyanazon az eszközön van
- Akkor is akarunk hálózatot látni, ha nincs fizikai összeköttetés
- IP címe: 127.0.0.1
- Minden tekintetben úgy viselkedik, mint egy valódi interfész
- Az egyes szolgáltatásoknál rendelkezhetünk róla, hogy ezen az interfészen is szolgáltassanak-e
- Nem csak számítógépeknél, hanem más dobozoknál: switch-ek, routerek stb.

MTU - Maximum Transmission Unit

- A fizikai médium rendszerint korlátozza az átvihető frame méretét
- Mértékegység: byte
- Célszerű ennél nem nagyobb csomagokat használni a felső rétegen
  - Nem tudhatjuk, hogy a célállomásig milyen közegen, milyen MTU-val megy át a csomag
  - Elvben lehet a legkülönbözőbb
  - Van TCP/IP implementáció, ami nem tolerálja
- Ethernet: 1500 a szokásos
- 802.3: 1492
- Path MTU: egy adatkapcsolatnál a közbülső legkisebb MTU

7. ARP, RARP

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

- ARP - Adress Resolution Protocol
- RFC826
- Elsősorban Ethernet

ARP csomag formátum

- 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

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
- Timeout: néhány perc (impl. függő)
- Erővel is lehet ARP entryket betenni, kivenni (Unixokon arp parancs)

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

- 4. előadáson van kép is
- Modemeken át jönnek állomások
- Modemek helyett manapság VPN, Virtual Private Network végződések
- 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

- RCF1868
- Ethernet ARP broadcast **reply**
- A source ethernet cím üres!
- A cahce-ekből mindenki kiüríti az IP címéhez tartozó ARP entry-t

Gratuitous ARP

- A saját IP címemre adok ki ARP kérést
- Segít felfedezni, ha tévedésből többször osztották ki ugyanazt az IP címet
- Sokszor az operációs rendszerek ezzel kezdik az IP cím használatát

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

Man in the Middle támadás ARP cache poisoninggal

Volt tavaly bevtechen...

Ettercap

- ARP hamisításon alapuló eszköz
- Egy broadcast domain-on belül működik

- Switch-elt hálózaton hallgatózásra (snooping)
- Diagnosztikai, hibakeresési céllal jól használható
- „Script kiddie” támadásokra is alkalmas
- Man in the Middle támadás végrehajtására
- VLAN-ok -viszonylagos- védelmet jelentenek

Mi van, ha nem tudjuk a saját IP címünket?

- RARP megoldás (RFC903)
- Külön ethernet frame type: 0x8035
- Broadcast kérés
- Válasz unicast
- Válaszban a címzett IP és ethernet címe

RARP hátrányok

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

8. BOOTP, DHCP

BOOTP

- Kliens gépek automatikus konfigurálása, indítása
- RFC951, RFC1542
- IP címen kívül visszaad:
  - Netmaskot
  - Router címet
  - Boot server host címet
  - Boot fájlnevet
  - ...
- UDP csomagok, server port: 67 (bootps), kliens port: 68 (bootpc)
- A kérés ethernet broadcast, a válasz rendszerint unicast (lehet broadcast)
- Hogyan használhat a kliens UDP-t, amikor még IP címe sincs?
  - A 0.0.0.0 forrás címet használja (unspecified address)
  - A 255.255.255.255 cél címet használja (broadcast)

BOOTP csomagformátum

- opcode: 1 - request, 2 reply
- htype, hlen: mint arp-nél
- hop count: a kliens 0-ra állítja, proxy szerverek (= bootp-t közvetítő router-ek) eggyel növelik
- xid: tracnsaction id. Ezzel derül ki, hogy mi mire válasz
- secs: az első bootp kérés óta eltelt idő
  - Ha a kliens nem kap választ exponential backoff szerint újra próbálkozik
- flags: Itt kérheti a legfelső biten a kliens, hogy a BOOTP reply is broadcast legyen
- 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 következő szerver IP címe, akivel a boot folytatható
- giaddr: Gateway IP address. Ha relé (bootp átengedésre konfigurált router) van, annak a címe

- chaddr: a kliens hardware (ethernet) címe - akkor van jelentősége, ha reléken át éri el a szervert a kliens
- sname: server host name
- file: boot fájlnév
  - Az operációs rendszeredet tartalmazó fájl, amit le kell töltened
  - A fájl maga TFTP protokollal tölthető le
- vend: különböző kiegészítések (vendor specific information)
  - Magic cookie: az első 4 byte
  - Tag kód, hossz, érték
  - Netmask
  - Router(ek) IP címe
  - DNS szerver IP címe

DHCP

- Dynamic Host Configuration Protocol, RFC2131
- BOOTP kompatibilis
  - Portok, üzenet formátum megegyezik a BOOT-vel, de bővül a lehetőségek köre
  - 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
  - DHCPACK

  - DHCPNAK
  - DHCPRELEASE (A kliens elengedi a lease-t (IP címet))
  - DHCPINFORM (A kliens csak konfigurációs paramétereket (opciókat) kér)
  - DHCPDECLINE (A kliensnek mégse kell a cím, mert úgy tudja, hogy már használatos)

- 4 lépcsős kezdeti handshake
  - DISCOVER (broadcast) →
  - OFFER ←
  - REQUEST →
  - ACK ←

- Lease-ek (A szerver az összes kiosztott lease-ről nyilvántartást vezet)
- parancs : lease ip cím

PXE - Preboot Execution Environment

- Az Intel szabvány, PC-k hálózati boot-olására
- DHCP-n és TFTP-n alapul
- DHCP kiegészítéseket tartalmaz
- Szabványos user interfészről is szól
- Nem csak Intel hardware-en használható

9. ICMP, ICMP hibaüzenetek

- ICMP - RDC792
- To control = vezérel
- STD 5 (IP-vel együtt)
  - Egy-egy RFC, vagy RFC-k egy halmaza lehet internet standard, (STD)
  - A szabvánnyá válás folyamatáról szól az RFC2026
- 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 altipusa
- Checksum: egyes komplemens összeadás, mint az IP-nél, az egész icmp üzenetre, 16 bites darabokban

ICMP üzenet-típusok

Description	Query (q), Error (e)
Destination unreachable:	egyik sem
network unreachable	e
host unreachable	e
protocol unreachable	e
port unreachable	e
fragmentation	e
source route failed	e
destination network unknow	e
destination host unknow	e
source host isolated	e
destination network administratively prohibited	e
destination host administratively prohibited	e
network unreachable for TOS	e
host unreachable for TOS	e
communication administratively prohibited	e
host precedence violation	e
precedence cutoff in effect	e
source quench	e
redirect	egyik sem
redirect for network	e
redirect for host	e
redirect for type-of-service and network	e
redirect for type-of-service and host	e
time exceeded:	egyik sem
time-to-live equals 0 during transit	e
time-to-live equals 0 during reassembly	e
parameter problem:	egyik sem
IP header bad	e
required option missing	e

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 fejrészet (20-60 byte)

- Az első 8 byte-ját az IP adat résznek
- TCP és UDP esetén ez tartalmazza a portokat

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)

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
  - 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
    - Path MTU discovery

Source quench

- Torlódás kezelés - congestion control
- 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
- Tűzfalak kiszűrhetik
  - hiba
  - furcsa jelenségeket okozhat
  - esetleg sokáig nem vesszük észre

Redirect

- 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
- 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
- Code = 1 : fragment reassembly time exceeded
- A visszaküldött csomagból látszik, hogy melyik kapcsolathoz tartozik
- Túlbuzgó tűzfalak ezt is kiszűrhetik - ez is nehezen kideríthető hibához vezethet

10. ICMP vezérlő (nem hiba) üzenetek

Description	Query (q), Error (e)
Echo reply	q
echo request	q
router advertisement router solicitation	q q
timestamp request timestamp reply	q q
information request information reply	q q
address mask request address mask reply	q q

Router advertisement/router solicitation

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

Router Solication

- 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

Echo request/reply - a ping program eszközei

- Type: request = 8, reply = 0

- ID: egy ping instanciát azonosít
- Sequence Number: egy instancián belül a sorszámot
- Adat: a küldött adatot vissza kell kapjuk
- ping program
  - Klasszikus eszköz egy IP cím elérhetőségének vizsgálatára
  - -c kapcsoló (count): ennyi request-et küld
  - -s kapcsoló (size): ekkora adat részt küld (+8 byte ICMP fejrész)
  - -f kapcsoló (flood): gyorsan küld sokat egymás után
  - Ha távoli gép nem elérhető, érdemes a default route-ot pingetni
  - -R kapcsoló (record route)

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
- 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
- Tűzfalak korlátozhatják
- Alternatívák
  - traceroute -I: ICMP csomagokat küld
  - mtr (my/Matt's traceroute): ICMP csomagok, látványos felület, mozi
  - tcptraceroute: TCP csomagok, a 80-as, vagy bármely más portra

11. UDP

- RFC, J. Postel, 1980
- IP protokoll mező értéke: 17
- 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
  - BOOTP/DHCP
  - RIP
  - NTP
  - DNS
  - NFS
  - Multimédiás 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
- 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
  - 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

UDP lite

- Egyes alkalmazásoknál (pl. hang) lehet, hogy jobb megtartani a sérült csomagot is
- UDP Lite - RFC3828 (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: fragmentál
  - Datagram-okat több packet-re
  - Teheti a küldö oldal
  - Teheti bármelyik közbülső router
- A fragmentálás a felső réteg (UDP, TCP) számára transzparens
- Az egyes fragmentumokban külön-külön IP fejrész van
  - Az egyes fragmentumokban ugyanaz lesz az ID
  - A felsőbb réteg fejrésze nem ismétlődik
- 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

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^{13})
- Egyes alkalmazások még tovább mennek: 512 byte-os csomagoknál nagyobbakat nem engednek meg (TFTP, BOOTP)

UDP szerver kérdések

- UDP-nél is lehet bizonyos értelemben kapcsolatról beszélni
- A kliens az elküldött UDP csomagnak megfelelő táblázatot, állapotgépet kezel - hasonlóan a TCP-hez
- 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

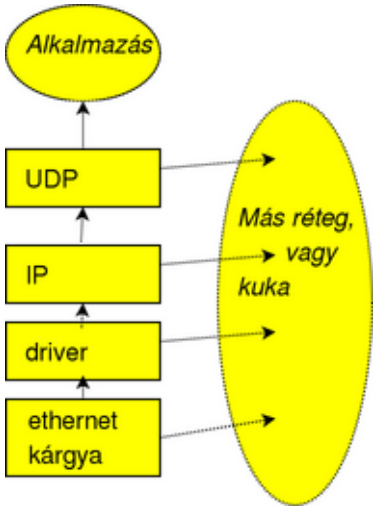
- Egy gépnek több IP címe lehet
- Alapértelmezésben minden IP címen figyelnek a szerver programok
- Az operációs rendszer módot adhat arra, hogy csak egy-egy IP címen figyeljenek
- Módot adhat arra is, hogy csak bizonyos IP címről/címekről fogadjanak el klienst

12. IP multicast ethernet hálózaton

Ethernet és IP broadcast és multicast

- TCP-nél, mivel az kapcsolatorientált, nincs értelme
- Broadcast és multicast rendszerint UDP-vel használatos
- A multicast egy → soknak kommunikáció, de multicast-tal megoldható a több → soknak kommunikáció is (pl. videokonferencia)
- Az ethernet kártya rendszerint csak azokat a csomagokat adja fel, amik:
  - Az ő MAC címére érkeznek
  - Broadcast címre érkeznek
- Minden csomagot felad, ha promiscuous módba tesszük
- Lehet kérni, hogy bizonyos multicast címekre érkező csomagokat feladjon
- Ethernet multicast címek: amiknek első 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

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:
  - 01:00:5e:00:00:00 - 01:00:5e:ff:ff:ff
  - A 00:00:5e-vel kezdődő unicast tartomány is a IANA birtoka
- Ennek első felét 01:00:5e:00:00:00 -től IP multicastra használjuk: ez 23 bit

Az IP címeknél multicast tartomány

Class	Range
Class A	0.0.0.0 - 127.255.255.255
Class B	128.0.0.0 - 191.255.255.255
Class C	192.0.0.0 - 223.255.255.255
MULTICAST	224.0.0.0 - 239.255.255.255
Reserved	240.0.0.0 - 247.255.255.255

- Azt is mondják, hogy a 224.0.0.0-239.255.255.255 címek D osztályúak (224.0.0.0/4)
- Már láttunk alkalmazást: Router advertisement/solicitation ICMP üzenetek
  - 224.0.0.1: all hosts
  - 224.0.0.2: all routers
- Itt 28 bit áll rendelkezésre
- Mappelés: az IP multicast felső 5 bitjét nem vesszük figyelembe

Csatlakozás egy multicast csoporthoz / Csoport elhagyás

- Az alkalmazás utasítja az IP réteget, az IP réteg az ethernet drivert:
  - Kér/lemond egy csoportot
  - Egy bizonyos interfészen!
  - Egy hoston, egy interfészen több alkalmazás is kérheti ugyanazt!
  - Ha egy csoportot kér egy alkalmazás, és abba a csoportba küld is, akkor a kernelnek éppen úgy kell bánnia vele, mintha kintről érkezett volna!

13. IGMP, PIM

IGMP - Internet Group Management Protocol

- Nem a multicast üzenetek küldése, hanem „szervezés”, vezérlés a feladata
- Host membership query/report protokoll
- Akkor van szerepe, ha nem csak egy LAN-on akarunk multicast forgalmat
- A routereknek tudni kell, hogy melyik interfészükre milyen multicast forgalmat kell továbbítani
- Egy-egy multicast csoport egy-egy részgráfot - célszerűen egy fát - jelöl ki a router-ek közt: általában nem feszíti ki az egész hálózatot
- RFC1112 - IGMPv1, STD 5 része
- RFC2236 - IGMPv2
- RFC3376 - IGMPv3

Reverse Path Check Forwarding (RPF)

- Eljárás, ami a csomag továbbítást befolyásolja a **forrás cím alapján**
- A router összeveti a bejövő (pl. multicast) csomag forrás címét a routing táblával: ha máshonnan jön, mint ahonnan várja, - **ahova ő küldené, ha a forrás cím cél cím lenne** -, akkor nem továbbítja
  - Így nem csak a TTL gondoskodik róla, hogy nem lesz kör a topológiában
  - Vegyük észre, hogy multicast kommunikációnál még veszélyesebb a kör: eleve többszöröződnek a routereknél a csomagok!
- Border routerekben RPF-et használhatnak hamisított source IP cím használata (IP Source Address Spoofing) ellen

IGMP üzenetek

- IGMPv1: Version, Type, Unused, Checksum, Group Address

IGMPv2

- Type, Max Resp Time, Checksum, Group Address
- Üzenet típusok:
  - 0x11 = Membership 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, az minden csoportot jelent, general query
- Csekkszumma: a szokásos one's complement összeadás
- Ha egy host csatlakozik egy csoporhoz egy interfészen, akkor küld egy megfelelő riportot
- A routerek időről-időre érdeklődő (general query) üzeneteket küldenek minden interfészükön
  - 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

IGMPv3

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

Switch-ek és multicast

- Lehet, hogy a switch a multicast csomagokat minden interfészére kikürtöli
- Lehet, hogy hallgatózik, és megtanulja, hogy hol milyen csoportok vannak (IGMP snoop)
- 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

PIM - Protocol Independent Multicast

- IP protokoll: 103
- IGMP-vel csak közeli LAN-okon át lehet multicast-ot szervezni
- A routerek közti multicast szervezés újabb protokoll(oka)t igényel

- A routerek és így az állomások valamilyen unicast routing protokoll (ezért PI) + PIM alapján találják meg a multicast partnereket
- **DR** - Designated Router
  - Egy-egy LAN-on (broadcast domain-ben) a PIM routerek választják maguk közül
  - Az egész LAN-t ez fogja képviselni a multicast forgalom szempontjából
- **RP** - Rendezvous Point: egy-egy csoporthoz tartozó találkozó router: a receiverek itt találkoznak a source-(okk)al
- A routerekbe rendszerint kézzel konfigurálják be, hogy hol van(nak) RP-k
- A routerek (S,G) = (Source IP, IP multicast group address) párokhoz építenek feszítő fát
- Join/Prune (csatlakoztató, lemetsző) üzeneteket küldenek a router-ek az RP/source fele
  - A multicast üzenetek az ellenkező irányban haladnak, mind ahogy a Join
  - A join üzeneteket periódikusan ismételni kell
  - Így felépül egy csoporthoz tartozó fa
  - Prune, ha nincs csoport tag a kapcsolódó hálózatokon
  - Prune, ha RPF jobb utat mutat
- A fa lehet:
  - Shared tree: bárki küldhet a csoporba (pl. videokonferencia)
  - Source specific tree: egy forrása van (pl. rádió, tv adás)
- A router a szomszédjairól nem tételezi fel, hogy tagjai az újonnan megjelent csoportnak
- A receiverek az RP-fele küldenek Join-t
- Ha adni akar egy állomás egy csoportba, akkor unicast csomaggal szól az RP-nek (Register)
- Kezdetben a source unicast-ba becsomagolva (encapsulated) küldi az RP-hez az adatfolyamot, az RP kicsomagolja, és multicast-ként továbbítja
- Erről folyamatról részleteket a 8. előadás alján találhatunk
- Később - ha RPF szerint jobbnak látják - kivághatják a fából az RP-t
- PIM SSM RFC3569 - Source Specific Multicast
- Dense (sűrű) Mode (RFC3973):

– A router a szomszédjairól feltételezi, hogy eleve tagjai az újonnan megjelent csoportnak
– Ha jelentkezik egy állomás egy csoportba, akkor azonnal beteszi a csoportba az interfészt
– Minden megfelelő irányba továbbítja a csoportba tartozó üzeneteket
– Nincs szükség RP-kre

14. DNS működés

A feladat

- Az Interneten számok (IP címek) azonosítanak gépeket
- Emberek számára ez nehezen megjegyezhető

Nevek bevezetése

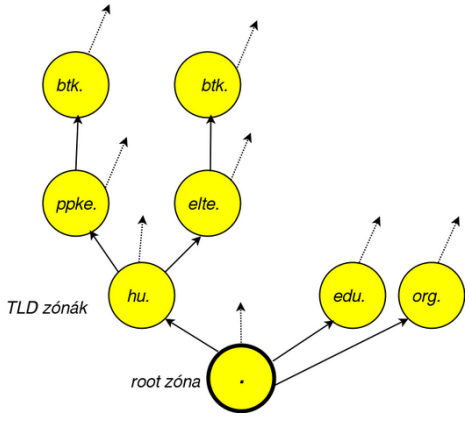
- **Név - Cím** megfeleltetés
- **Cím - Név**

Host táblák

- Központi, szétküldött fájl
- Soronként tartalmaz egy-egy név-IP cím megfeleltetést
- Kezdetben használták
- Unixokon máig is él: /etc/hosts

Hierarchikus, osztott adatbázis

- A nevek hierarchikusan épülnek fel
- A hierarchia egyes szintjein önállóan döntenek
- Minden szinten tovább lehet delegálni
- A feloldás rekurzívan történik a hierarchián
- Nagyon jól skálázható: ma sok millió név, sok százezer névszerver
- Egyedül a .com alatt kb. 115 millió elágazás (delegálás)



A DNS

- RFC1034, RFC1035
- Nevek:
  - Alfanumerikus karakterek és a - (dash)
  - **LDH** (Letter, Digit, Hyphen) karakterek
  - Kis /nagybetű nem számít
  - Hiererchia balról jobbra
- Label - két pont közötti rész

Kliens-szerver elv

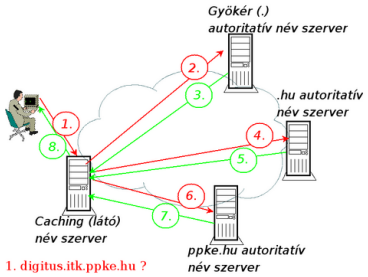
- Kliens: resolver
  - Gyakorlatilag minden IP szoftvernek része
  - Legalább egy szervert látnia kell
  - Konfigurációs paraméter
- Szerver: DNS szerver
  - Sok operációs rendszer alatt
  - Nem feltétlenül autoritás
  - Kapcsolat a többi szerverrel
  - Cache!!

A kommunikáció

- UDP 53-as porton történik
- Néha TCP 53-as port
- A név fán a gyökértől kezdve lépésről lépésre a hiararchia szerint

A DNS szerverek kettős feladata

- Látni
  - Az elosztott adatbázist kérdezni szerte az interneten
  - Ez a rekurzív név szerverek, más néven rekurzor-ok, vagy caching név szerverek feladata
- Mutatni
  - A rájuk tartozó részről a többi szerver számára adatokat szolgáltatni
  - Ez az autoritatív más néven hiteles név szerverek feladata
- A szerverek ennek megfelelően lehetnek



  - Caching szerverek, vagy
  - Autoritatív más néven hiteles szerverek
  - Egy-egy szerver mindkét feladatot elláthatja

DNS üzenet formátum

Header, Question, Answer, Authority, Additional

Header

- 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

Question

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

- QCLASS: majdnem mindig 1, azaz IN, Internet Class

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.

Rezolver konfiguráció

- Minden TCP/IP-t használó gépen szükséges
- IP címmel kell megadni
- Több lehet (primary, secondary)
- Elvben az internet bármely pontján
- Célszerűen: hálózati értelemben közel levő
- A DNS szerverek (>8.x Bind) korlátozhatják

Cache, TTL

- A DNS szerverek a megtudott nevekre emlékeznek
- Haszon
  - Gyorsabb feloldás
  - Hálózat kímélése
- Hogy meddig kell cache-elni, 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
- Minimális védekezés: csak az autoritatív név szerverektől szabad elfogadni adatot
- 9. előadás második résznél levő kép sokat segít a megértésben

A "Kaminsky bug"

- 2008. augusztus, Blackhat konferencia: Dan Kaminsky nyilvánosságra hozta, hogy a cache mérgezés könnyebb mint gondolták

- A gyártók júniusban javították a programjaikat, „upgrade” kampányt kezdtek

Segédprogramok

- Csak a DNS rendszerben való böngészésre
- DNS intelligencia
- Tetszőleges szervertől kérdezhetünk
- NSLOOKUP
- HOST, DIG: unixokon

Domain

- Minden elem domain
- Nem csak IP cím lehet, hanem
  - hardware info
  - tovább delegálás
  - levelezési info
- A név fa egy pontja

Zóna

- A fának egy egyben kezelt ága
- A mutató szerver szempontjából egy egység
- Egy szerver rendelkezhet több zóna felett

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
  - konfigurációs (SOA rekordban eldöntött) paraméterek

Gyökér (root) domain

- Pont olyan mint akármely domain

- A szerverei szét vannak szórva az interneten
  - Minden szerverbe be kell konfigurálni!
  - Fontos, hogy mindig elérhetők legyenek

TLD - Top level domain

- Eredetileg felhasználó típusa szerint
- gTLD-k - generic Top Level Domains
  - edu, gov, org, com
- Az Internet nemzetközivé válásával országkódok
- ccTLD-k - country code Top Level Domains
- Manapság egy domain név birtoklása üzleti érték
- Új top level domainek jelentek meg
  - biz, info, pro

Domain nevek

- FQDN = Fully Qualified Domain Name
- használható, egyértelmű
- Hierarchikus felépítés

Inverz leképzés

- IP címből név
- Visszavezetik név leképezésre
- Speciális domain:IN-ADDR.ARPA
- Az IP címet fordított sorrendben kell írni
  - Pl: 67.84.225.193.in-addr.arpa
  - A segédprogramok automatikusan megfordítják
- Sokszor elfelejtkeznek róla!

15. DNS rekordok

Name szerver programok

- Klasszikus: BIND (Berkeley, Internet Name Daemon Software)
- Újabbak: djbdns, NSD
- Zónák szempontjából
  - Primary
  - Secondary
  - Cache only
- Firewall mögött levő: slave
- Cache-t incializálhatjuk a root szerverek címeivel

Caching only szerver, forwarder

- Érdemes minden lokális hálózaton legalább egy DNS szervert futtatni
- Ha nem autoritás, akkor caching only
- A cache-ét kiegészítheti, ha forwardereket használ
- Forwarder
  - Mielőtt az interneten érdeklődne, először ezt kérdezi
  - Nagyobb, hatékonyabb lesz a cache

Zóna fájl

- Az adatbázis adatait tartalmazza
- Elemei: Resource Rekordok (RR)
- Legfontosabb
- Start Of Authority (SOA)
- Name Server (NS)
- Address (A)
- pointer (PTR)
- Canonical Name (CNAME)
- Mail eXchanger (MX)

SOA rekord

- Globális zóna adatok

- A legtöbb adat a secondaryk-nak szól
- Szokás a sorszámban a dátumot kódolni
- ÉÉÉÉHHNNVV forma (az 5. évezredig jó)
- A sorszámot FONTOS növelni
- Ha a refresh, retry nagy
  - kiméli a secondary-kat, de:
  - változások lassabban terjednek
  - Expiration ideig lehet, hogy nem vesszük észre, ha elrontottunk valamit!

Idő értékek a SOA rekordban

- Másodperc az egység, de Bind-nál W (hét), D (nap), H (óra) is megadható
- Refresh, Retry, Expiration, TTL

Address rekord

- Név - IP cím hozzárendelés
- Nem szabad az inverz bejegyzésről elfeledkezni

NS rekord

- Egy aldomain autoritását tovább delegálja
- Az apuka és a gyerek zónában is megjelenik
- Ajánlatos több autoritatív név szerver azaz több NS rekordot használni
- Az argumentumban szereplő gép gazdája a felelős!!!

Glue rekord

- Az apuka zónába kényszerülő A rekord
- pl: ppke.hu delegálja az btk.ppke.hu-t

Lame delegálás

- Az apuka zóna szerint autoritás
- Mégsem mutatja a zónát
- Okok
  - Legtöbbször ember-ember kommunikáció hiányossága
  - Rossz konfiguráció

CNAME rekord

- Kanonikus név (alias)
- Leggyakoribb felhasználás:
  - Szolgáltatás jelölése
  - Ha valami CNAME, akkor nem lehet

MX rekord

- Mail eXchanger
- Levél célállomást jelöl
- Az első paraméter prioritás, nem súly!
- Intézmény címzés egyszerűsítés

SRV rekord

- Service meghatározására
- Az MX rekord általánosítása

HINFO és TXT rekord

- Emberek tájékoztatására szolgál
- HINFO = Hardware információ
- Hardware és szoftver neve
- TXT = Text
- Tetszőleges szöveget tartalmazhat
- Szellemes TXT használat kurrens adatbázis feltüntetése

Delegálás az in-addr.arpa zónákban

- Az IP címtartományokkal gazdálkodók delegálják
- Rendszerint szolgáltatók
- A, B, C osztály
- Az IP címet fordítva írjuk

PTR rekord

- Az in-addr.arpa - olyan, mint más domain
- Címhez domain nevet rendel

- Az egyenes és inverz delegálás nem jár feltétlen együtt

Ékezetes domain nevek

- Köznyelvi szavakat akarunk domain névként használni
- A domain névként csak LDH karaktereket: [a-z0-9-] használhatunk
- Bár tetszőleges bináris információ lehetne a DNS rekorokban
- IDN = Internationalized Domain Names

IDNA - International Domain Names in Applications

- 2001 decemberében IETF döntés
- A DNS szerkezet nem változik, az alkalmazások konvertálnak IDN/LDH közt
- Előnyök
  - Nem kell a DNS protokollt változtatni
  - Nem kell a DNS infrastruktúrát változtatni
  - ACE - Ascii Compatible Encoding
    - Eszköz, mely lehetővé teszi, hogy a DNS szoftver komponensek semmit ne változzanak az ékezetes domain-re való áttéréskor

Punycode

- Adam Costello (Berkeley Egyetem) munkája, RFC3492
- Az ascii karkakterek változatlanok maradnak
- Nem a karaktereket, hanem a karakter-pozíció párokat kódolja
- Nem is ezeket a párokat, hanem ezek különbségeit
- Nagyon tömör - sokszor tömörebb, mint az utf8!

Implementációk

- Böngészők
- libidin
  - Működő stringprep, punycode és IDNA implementáció
  - Library + utilityk
  - Szabad (GPL) szoftver
  - Az interneten látható IDN eszközök többsége ezen alapul

IDN a .hu alatt

- Csak magyar nyelvű domain nevek
- Megengedett karakterek: LDH, ékezetes kis betűk
- Nem vezetünk be kötegeket

16. TFTP

TFTP - Trivial File Transfer Protocol

- Egyszerűen implementálható fájl küldés/fogadás
- Boot szervereknél használatos: DHCP paraméter a host és a fájl neve
- RFC1350
- UDP-t használ, 69-es porton szólitja meg a kliens a szervert
- A tényleges adatforgalom már nem a 69-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 elsp két byte opcode, read=1, write=2, data=3, ack=4, error=5
- Írásnál és olvasásnál 0-val terminált fájlnév
- Mode: netascii, vagy octet
  - netascii-nál a sorok CR/LF közt
- 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 csekszuma

A stop and wait elv sávszélesség

- A tényleges átviteli sebesség nem csak a fizikai vonalak sebességétől, hanem a késleltetéstől is függ
- A fogadónak lehetne ilyesfajta mondása: „van három csomagnyi bufferem, küldhetsz egymás után annyit”
- Ez a **window mechanizmus**

Bűvészinas szindróma (Sorcerer's apprentice syndrome)

- A k-adik nyugta késik, de nem vész el
- A k-adik adatot újra küldi a küldő
- Megérkezik a késett k-adik nyugta

- A küldő küldi a k+1-edig adatot
- Megérkezik az újraküldött k-adikra küldött nyugta
- A küldő küldi a k+1-edig adatot
- Ettől kezdve minden csomagot kétszer fog küldeni

Biztonság

- A TFTP protokollban nincs azonosító/jelszó se semmi más biztosíték
- A szerver implementációk korlátozásokat tesznek lehetővé, mint pl.:
  - Csak bizonyos fájlokhoz
  - Csak bizonyos IP címekről
  - Csak olvasási joggal

Hol használnak TFTP-t?

- Router, switch operációs rendszerek frissítésekor
- Router, switch konfiguációs fájlok tárolására, mentésére
- Vékony kliensek, PC-k bútolásakor (PXE)

17. TCP

TCP - Transmission Control Protocol

- Az alapja RFC793 - Postel
- A leggyakrabban használt 4. szintű protokoll
- Kapcsolat-orientált: mindig pontosan két partner közti kommunikáció

TCP szolgáltatások

- A TCP darabokra bontja az információt. Egy IP rétegnek átadott darab: szegmens
- Minden szegmensre nyugtát vár
- Ha nem jött nyugta, újraküld
- Minden szegmenst csekkszumma véd. Ha ez hibát jelez, egyszerűen eldobja
- Az IP csomagok nem feltétlenül sorrendben érkeznek meg. A TCP helyreállítja a sorrendet
- IP csomagok duplázódhatnak. A TCP kiszűri a duplikátumokat
- A TCP folyamvezérlést - flow controlt - alkalmaz: az adatáramlás sebességét fékezni és gyorsítani tudja a rendelkezésre álló erőforrásoktól függően
- A TCP byte (oktet) folyamot visz át: nincs rekord vagy sorhatár fogalom. Ilyesmiről a felsőbb szintnek kell gondoskodni, ha szükséges
- A TCP full duplex kapcsolatot tesz lehetővé: mindkét irányban és egymástól függetlenül áramlik a byte folyam

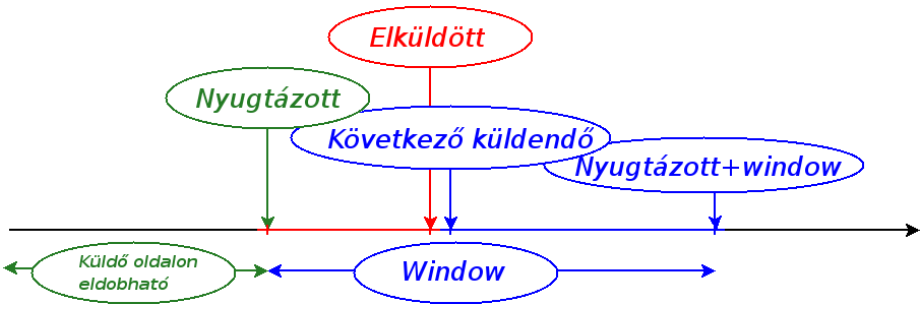
Header

- A kapcsolatot ez a négyes határozza meg: source és destination port, source és destination IP cím (telefon hívás analógia)
- Sequence number: a byte folyamban az ebben a szegmensben küldött első byte sorszáma
  - Hagyományosan nem is véletlen, megjósolható
  - Támadásra ad módot, manapság véletlen szám
  - SYN, synchronize flag: a kapcsolat felépítésekor használatos
- Acknowledgment number: a vett byte-folyamot eddig a sorszámig nyugtázom. A következő venni kívánt sorszámot tartalmazza

  - Az acknowledgement number akkor érvényes, ha az ACK bit áll
  - Az ACK bit rendszerint áll: ha nem jött új adat (vagy jött, de nem akarom nyugtázni), megismétlem az előző acknowledgement numbert
  - A TCP sliding-window (csúszó ablakos) nyugtázást alkalmaz, szelektív és negatív nyugták nélkül
    * Ha kimaradt egy csomag, de a következő már megjött, nincs mód arra, hogy a megérkezettet nyugtázzam
    * Ha megjött egy csomag, mondjuk 2001-től 3000-ig de hibás, nincs erre más mondásom, mint az, hogy újra nyugtázom 2000-ig a folyamot

- Data offset: a fejrész hosszát adja meg 4 byte-os egységekben
- Flagek
  - URG: az urgent (sürgős adat) pointer érvényes
  - ACK: az acknoledgement number érvényes
  - PSH: azonnal add fel a felső rétegnek a szegmenst (push)
  - RST: lecsaptam a kagylót, durva bontás (reset)
  - SYN: szinkronizáljuk a sequence numbereinket, kapcsolat indítása
  - FIN: befejeztem az adatküldést, leteszem a kagylót (finish)
- Window: ablak. Az acknowledgement numbertől ennyi byte fogadására készen állok
- Csekkszumma: a szokásos one's complement 16 bites darabokra, a teljes szegmensre
  - Nem csak az TCP csomagot, hanem egy „pszeudó fejrészt" is figyelembe vesz
- Az IP fejrészből ismétel meg elemeket (A rossz helyre küldött TCP csomagokat lehet így kiszűrni)
- Urgent pointer: ha áll az URG bit, akkor az adatban eddig a pontig sürgős adat van
- Opciók
  - Típus: 1 byte-os opció
  - Típus, hossz, opció adatok
  - Gyakran használt opció: MSS, Maximum Segment Size
- Adat: nincs feltétlenül

Sliding window



- A nyugtázott+window sorszámig folyamatosan küldhető adat
- A nyugtázott+window sorszámnál nagyobb sorszámú oktet nem küldhető
- A window folyamatosan csúszik jobbra az ábrán
- A window-t a fogadó csökkentheti/növelheti
- A window mérettel a fogadó szabályozhatja a küldés ütemét
  - Flow control: az alkalmazás igényeihez igazodhat
  - 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
- Újraküldés (RTO lejár és nem kapott ACK-t, vagy 3 ACK)

TCP kapcsolatfelépítés

- Three-way handshake
  1. A kezdeményező (kliens) küld egy SYN flages csomagot. Eldől a kapcsolatra
     - A hívó fél ISN-je
     - portok. A hívó port többnyire véletlenszerűen választott

2. A hívott (szerver) küld egy ACK-t, és SYN-t tartalmazó csomagot. Nyugtázza a kapott csomagot, a SYN elfogyaszt egy sequence numbert. Eldől: A másik ISN
3. A hívó küld egy ACK-t, nyugtázza a másik csomagját. Az a SYN is elfogyaszt egy sequence numbert.

- A kezdeményezőre azt mondjuk: active open-t hajt végre
- A hívott: passive open-t
- Ha a kapcsolat felépítés nem sikerült, a kezdeményező timeout után újra próbálkozik
- Ha többedszerre sem sikerül, értesíti az alkalmazást a kudarcról

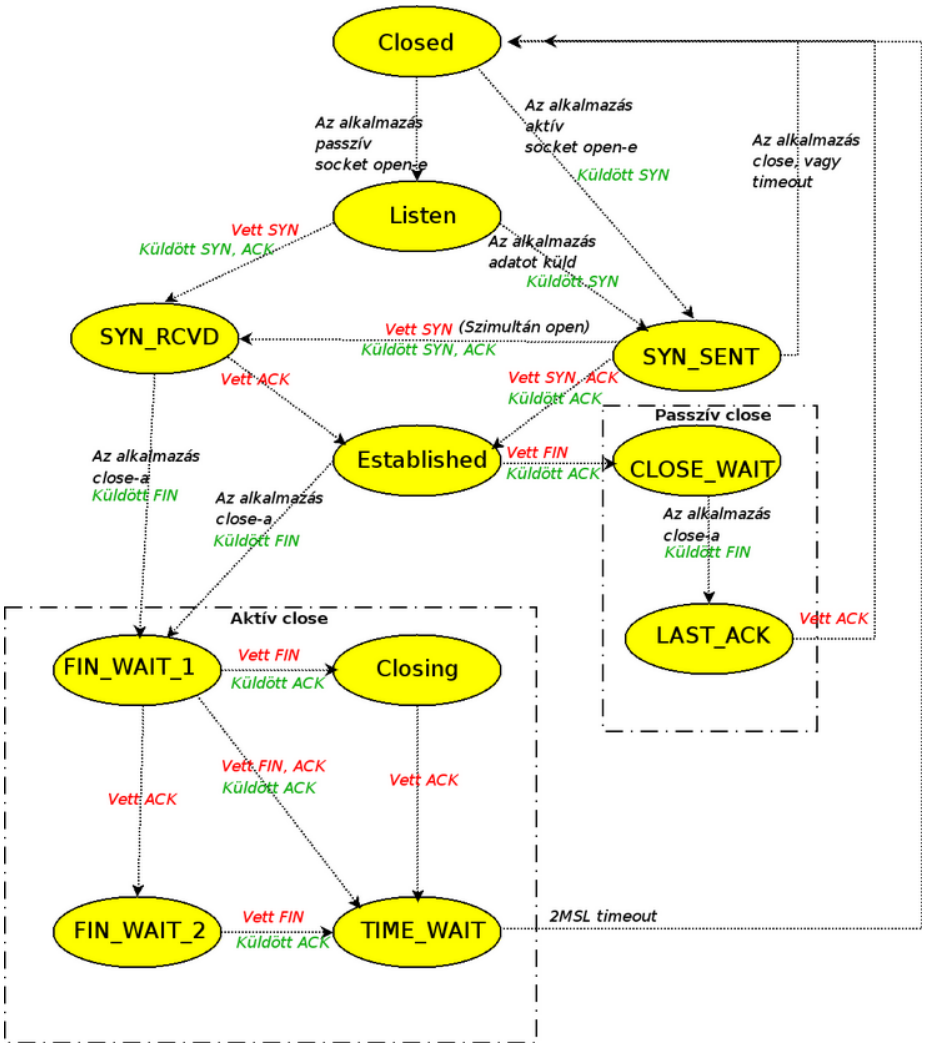
TCP SYN flood támadás

- A támadott IP címre SYN csomagok tömegét küldik
- Védekezés: Egy hálózatból kifele csak az oda tartozó source címekkel mehessenek ki csomagok

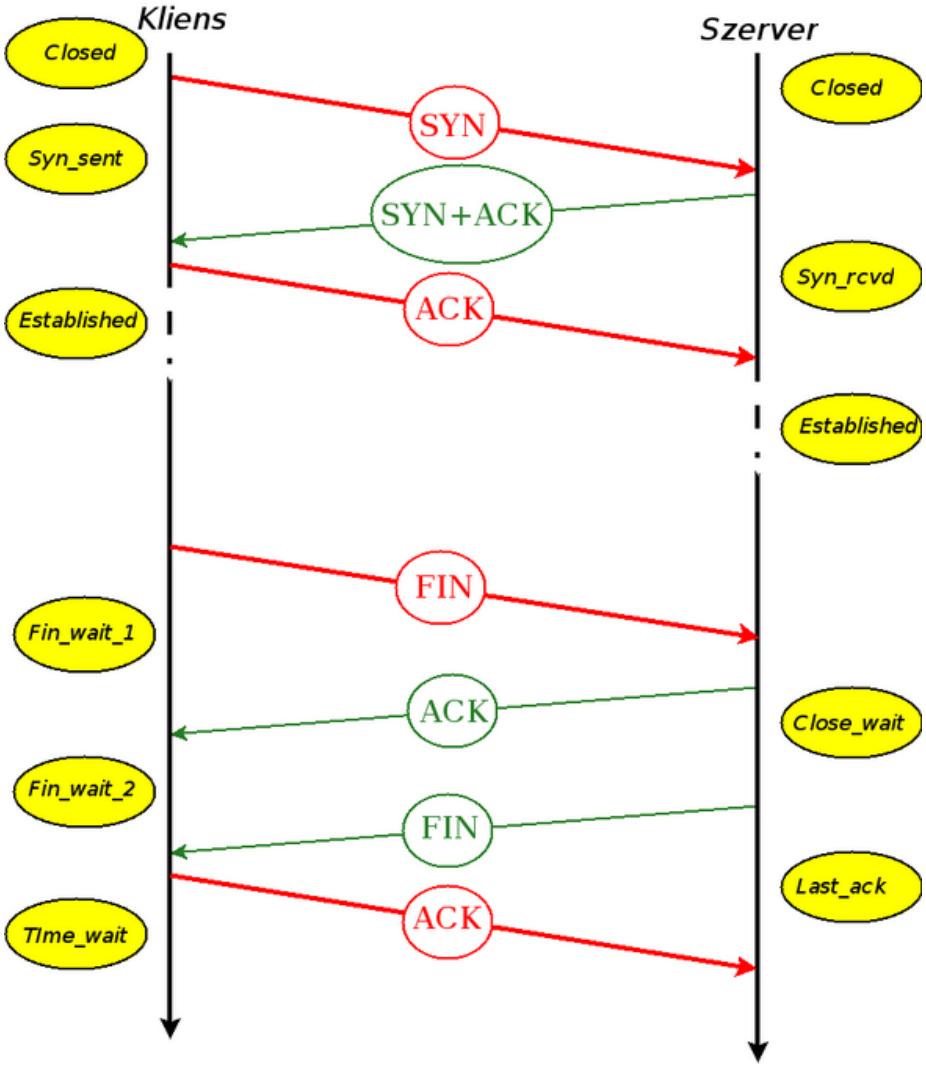
Kapcsolat zárás

- A TCP kapcsolat full duplex, ezért a FIN csak azt jelenti: én nem küldök több adatot
- TCP half close: csak az egyik irányban zárt kapcsolat (ritka)
- Négy csomag utazhat
  1. Az egyik (A) fél küld egy FIN-t
  2. B nyugtázza ACK-val
  3. B is küld FIN-t, ha befejezte az adatküldést
  4. Erre is megjön a nyugta
- A FIN-is elfogyaszt egy sequence numbert
- A active close-t, B passive close-t hajt végre

TCP állapotábra



A szokásos folyamat



TIME-WAIT állapot

- Az aktív close-t végző utolsó állapota
- Lehet, hogy elvész az utolsó küldött ACK
- Ilyenkor a passzív oldal újra küldi a FIN-t
- MSL: Maximum Segment Lifetime. Becsült érték: ennél tovább nem lehet a hálózaton egy TCP szegmens

- TIME-WAIT állapot = 2MSL állapot: 2xMSL ideig van ebben egy kapcsolat, utána "closed"
- Lehet, hogy amiatt nem indul el egy szerver, mert nem tudja megnyitni "listen"-re a socket-et: TIME-WAIT-ben van
- Nem csak a TIME-WAIT, hanem a FIN-WAIT-2 is olyan, hogy benneragadhatunk: timeout után fel kell szabadítani az erőforrásokat, alapállapotba kell menni

Quiet time elv

- Ha egy gép újraindul, akor a TIME-WAIT-ben levő kapcsolatairól nem tudhat
- Ezért 2xMSL ideig újrainduláskor ne létesítsen TCP kapcsolatot
- A gyakorlatban a boot idő nagyobb

Reset

- Akkor küldik, ha olyan csomag érkezik, amit nem találunk szabályosnak
- Például ha olyan portra érkezik TCP kérés, amin nem figyel alkalmazás
- Kapcsolat erőszakos bontására is használható (abort)

Half open kapcsolatok

- Ha TCP kapcsolatban álló állomások közül az egyik (vagy rajta az alkalmazás) újraindul, a másik fél élőnek hiheti a kapcsolatot
- Küldhet adatot, ez meglepetés lesz az újra indult oldalon
- Ilyenkor RESET a válasz

Szimultán open

- Keresztbe küldhet két alkalmazás SYN-t ugyanarra a socket párra
- Ilyenkor mindkettő ACK-val válaszol
- A kapcsolat felépül és csak egy kapcsolat épül fel!

Szimultán close

- A küldött FIN-re nem ACK, hanem FIN jön
- ACK-val meg kell válaszolni, és várni az ACK-ra: CLOSING állapot
- lyenkor mindkét oldal TIME-WAIT állapotba kerül

TCP reset támadás (TCP Reset attack)

- A és B közt élő TCP kapcsolatba E RESET csomagot csempészhet

- 2004 májusában nagy port vert fel
- Felfedezték, hogy nem szükséges a kurrens sequence numbert tudni: elég a window-ban levő bármi!
  - Nagy window méretnél (ami egyre gyakoribb) könnyebb a támadás
  - Brute force támadás is indítható

TCP szerverek

- Az egyes ismert szolgáltatásokhoz a IANA portokat rendel: well-known ports
- Az implementáció, sőt a konfiguráció mást is választhat
- Unixokban az /etc/services tartalmazza az ismert portok nevét és számát
- A szerverek induláskor figyelnek a választott porton (Listen állapot)
- Ha kérés érkezik, a szerver egy gyerek processzt forkol
- A netstat parancs mutatja a pillanatnyi kapcsolatokat

inet daemon

- Unixokon klasszikus, egyszerű szolgáltatások indításának eszköze
- Belső szolgáltatások (pl. echo) és tetszőleges program indítása

TCP daemon, tcpd, alias tcp-wrapper

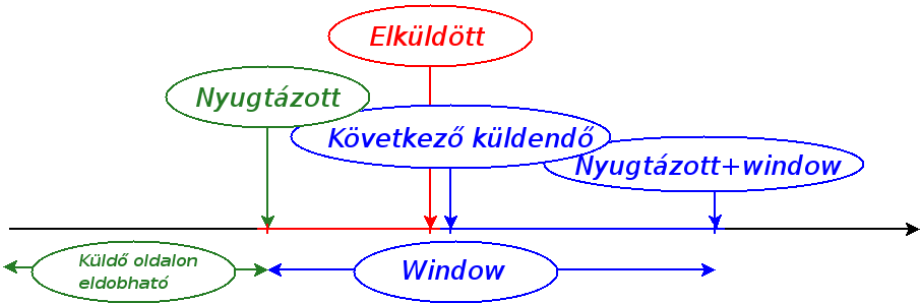
- inetd és egyes programok közé ékelődhet
- Árnyaltan lehet akciót kezdeményezni
  - Elutasítani/elfogadni a kapcsolatot
  - Nalózni
  - Figyelmeztető levelet küldeni, tetszőleges parancsokat kiadni
- Két fő konfigurációs fájl: /etc/hosts.allow, /etc/hosts.deny
- Nem csak inetd mögött, hanem libwrap-pal egybelinkelt bármilyen programmal használható

18. TCP flow control

Hogyan adjunk, ha ki akarjuk használni a sávszélességet?

- Figyelembe kell venni a késleltetést is (TFTP példa)
- Akkor küldhetünk folyamatosan, ha az első csomagunk nyugtájának megérkezéséig folyamatosan adhatunk
  - A csomagfordulási idő, RTT – Round Trip Time korlátoz
- Sávszélesség * RTT elküldött bit legyen a window
- Pl. 100 Mb/sec vonal, 0,01 sec RTT: 1Mbit = 125 KByte
- Pl. 100 Mb/sec vonal, 2 sec RTT: 200Mbit = 25 MByte !

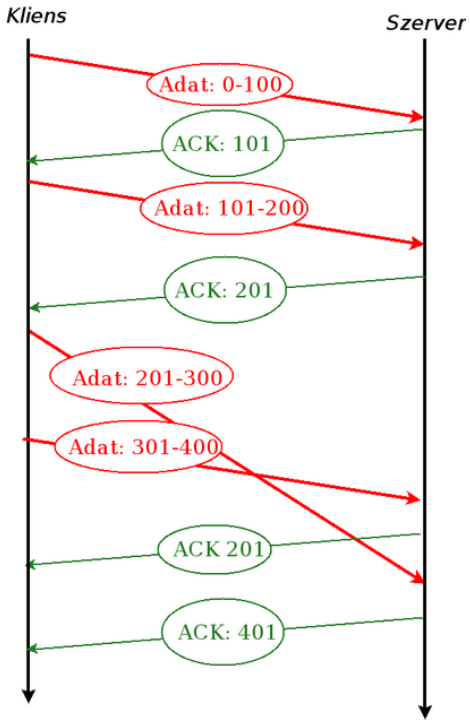
Sliding window



- A nyugtázott+window sorszámig folyamatosan küldhető adat
- A nyugtázott+window sorszámnál nagyobb sorszámú oktet nem küldhető
- A window folyamatosan csúszik jobbra az ábrán
- A window-t a fogadó csökkentheti/növelheti
- A window mérettel a fogadó szabályozhatja a küldés ütemét
  - Flow control: az alkalmazás igényeihez igazodhat
  - 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
- Újraküldés (RTO lejár és nem kapott ACK-t, vagy 3 ACK)

TCP nyugták



Window scale opció

- RFC1323
- Fejrész:
  - 2-nek a shift count -adik hatványa lesz a window paraméter mértékegysége
  - 0 azt jelenti, hogy a window byte-okban számolandó (nincs scale)
  - 3 azt jelenti, hogy 8 byte-os darabokban
  - Maximum 14 a megengedett = 2^{30} = 1G
  - A SYN flag-gel együtt küldhető a three-way handshake során
  - Akkor él, ha a SYN/ACK-ban is megismétli a partner
  - Lehet különböző a két irányban

Delayed ACK

- A TCP réteg nem feltétlenül küld nyugtát, ha tud
- Egy timer lejártát megvárja, hátha addig lesz:
  - elküldendő adat, amivel „mellékesen” nyugtáz: piggyback nyugta
  - újabb bejövő adat, amit nyugtázhat.

ACK generálás

Esemény	Akció
A nyugtázotthoz 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 (time ketyeg)	Azonnal küldj ACK-t csúsztasd a windowst
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

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
- Exponential Backoff: ha újraküldünk, az RTO-t duplázzuk, egy határig (jellemző érték: 9 perc)

Interaktív adatforgalom

- Kis csomagok

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

Nagle algoritmus

- Probléma: a lassú vonal végén ülő felhasználó gépelésével mindig újabb csomagokat generál
- RFC896
- Nem küldünk újabb csomagot, bufferelünk, míg van nyugtázatlan kintlevő csomag
- Timeout után mindenképpen ürítünk
- Önszabályozó: ha gyors a hálózat, nincs is hatása
- Egyes alkalmazásoknál hátrányt jelent

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

Slow start

- Congestion avoidance: mit csináljunk, hogy elkerüljük a torlódást
- 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 (klasszikusan) 1 MSS méretű
  - Lehet 10 MSS, sőt van javaslat még nagyobbra
- Minden ack-zott szegmens 1 MSS-sel növeli a cwnd-t
- Slow start alatt az átvitel exponenciálisan n
- 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
- Additive increase, multiplicative decraese
- Bevezetünk egy új változót: slow start threshold, ssthresh
- Slow startot alkalmazunk, ha cwnd < ssthresh, congestion avoidance-t különben
- Ha congestion-t észlelünk (RTO, vagy tripla ack), akkor
  1. Újraküldjük a szegmenst (fast retransmit)
  2. A nyugtázatlan-byte-ok/2-re, de legfeljebb 2 MSS-re csökkentjük ssthresh-t (multiplicative decrease)
  3. Ha RTO történt, akkor slow start-tal indítunk: cwnd = 1
  4. Ha tripla ack, akkor cwnd = ssthresh + 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 (cwnd=sshthresh+3MSS): ez a fast recovery

RED - Random Early Deection

- A TCP congestion control eljárások egészségesebbé tették az internetet
- Active queue management a routerekben (hálózat belsejében)
- 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
- RED működése
  - 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
  - Változók: minimum (m) és maximum (M) küszöb, $0 < \text{maxp} < 1$
  - 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 hosszal arányos 0 és maxp közti valószínüséggel dobjuk el a csomagot
- WRED - Weighted RED: vannak „egyenlőbb" csomagok: valamilyen szempont szerint bizonyos csomagokat kevésbé valószínűen dobunk el

ECN - Explicit Congestion Notification

- RED-del együtt használt eljárás
- Kerüljük a csomagok eldobását, helyette szinezünk
- TCP és IP, végberendezés és router aktív együttműködése
- Az IP és TCP fejrészben új flag-eket használ (ECT, CE, ECE ECN Echo, CWR
- TCP kapcsolat felvétel ECN-nel
- ECN a routerben (middlebox-ban)
- ECN a fogadó oldalán
- ECN az adó oldalán
- Kompatibilitási probléma

Persist timer

- Ha a fogadó bufferei elfogytak, bezárja a window-t
- Ha újra tud fogadni, nyitja: ACK megismételt acknowledgement 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 windown kívül!) (window jobb széle után)

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.í.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 kér

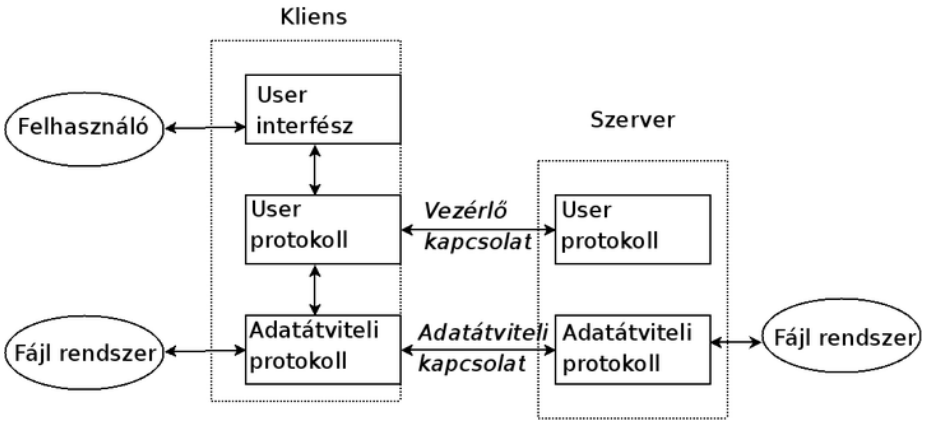
Keep-alive timer

- TCP kapcsolatok eredendően nem bomlanak forgalom hiányában sem
- Az eredeti szándék szerint az alkalmazások használhatnak „AYT" (Are You There) mechanizmust
- Mégis divatba jött a keep-alive mechanizmus
  1. Feleslegesen lebonthat később még használható kapcsolatokat
  2. Feleslegesen hálózati erőforrásokat használ
  3. Kidobott pénz, ha forgalom után fizetünk
- opcionálisan használni lehet
- Keep-alive time: jellemzően 2 óra, általában rendszerparaméter
- Ha egy kapcsolat keep-alive ideig néma, akkor a mechanizmust alkalmazó oldal küld egy csomagot róla
- A másik oldal erre ACK-t küld

19. FTP

FTP - File Transfer Protocol

- RFC959
- Klasszikus szolgáltatás
- Két TCP kapcsolaton épül
  - Vezérlő (control)
    - A kliens egy ephemeral portjáról a szerver 21-es portjára
  - Adat (data)
    - Klasszikusan a szerver építi fel a saját 20-as portjáról
    - A kliens is felépítheti, ez a passzív ftp



- Web böngészők (pl. Firefox) és fálj menedzserek (pl. Windows Commander) is támogatják
- Alternatívák
  - SCP, SFTP, HTTP...
- Különböző architektúrák, fájl rendszerek, fájl formátumok, karakter készletek
- Fájl típusok
  - ASCII
  - EBCDIC - főleg IBM által használt klasszikus karakterkódolás
  - Image - bitfolyam, változatlan formában tárolandó
  - Local - különböző byte hosszakat használó gépek közti átvitel

- Format control
  - ASCII és EBCDIC fájlok tulajdonsága
  - Nyomtatási vezérlő információ
  - Lehetséges értékei
    - Nonprint
    - Telnet - CR, LV, VT van a fájlban
    - Fortran - a rekordok első byte-ja vezérlő byte
- Fájl szerkezet
  - Byte stream
  - Rekordok – szöveges típusú fájloknál használatos
  - Indexelt lapok (page) – tetszőleges sorrendben küldhetők
- Átvitel módja
  - Stream (bájt folyam). Ez az alapértelmezés
  - Block mode
    - Blokkokban történik az átvitel
    - Minden blokk fejrészből és adatból áll A fejrész hosszból és leírásból.
  - Tömörített mode
    - Az egymás után következő egyforma bájtokat rövidítve kódolja
    - Nemigen használatos: jobb tömöríteni a fájlokat pl. gzip-pel

FTP control kapcsolat

- A kliens egysoros, CR/LF-fel záródó parancsokat ad
- Ezek a parancsok nem feltétlenül a kliens előtt ülő felhasználó közvetlen parancsai4
  - USER kicsoda
  - PASS jel
  - TYPE típus - fájl típust határoz meg a következő átvitelekre
  - LIST file-ok - listát ad egy könyvtárról
  - RETR file - hoz egy fájlt a kliensre
  - STOR file - visz egy fájlt a kliensről
  - ABOR - a folyamatban levő átvitelt elveti
  - PORT n1,n2,n3,n4,n5,n6 - adatkapcsolat nyitás kezdeményezése (aktív mód)
  - PASV - adatkapcsolat nyitás kezdeményezése (passzív mód)
  - QUIT - bontja a kapcsolatot

- FTP szerver válaszok
  - Szerkezet: xyz valami szöveg
    - xyz három számjegy, a program csak azt nézi
    - A szöveg nekünk, embereknek szól
    - x jelenti, hogy siker vagy kudarc
      - 1 = részleges, előzetes sikeres válasz
      - 2 = siker
      - 3 = részleges, közbülső állapotot jelző sikeres válasz
      - 4 = átmeneti sikertelenség
      - 5 = végleges sikertelenség
  - Ha többsoros a válasz
    - A folytatósort - (dash karakter) jelzi a szám mögött
    - Az utolsó sorban megismétlődik a numerikus kód, - nélkül
  - Példák a 10. előadásban
- Ez az ember által is fogyasztható parancs/válasz modell az alapja más protokolloknak is
  - SMTP - levelezés
  - HTTP - web lapok (Példa: HTTP 404-es hiba)
- SIP - multimedia kapcsolatok

FTP adat kapcsolat

- A parancsok csatornájától független TCP kapcsolat épül fel
- Rendszerint minden fájl átvitelhez külön
- A könyvtár listázás és más hasonló parancsok is ezen a módon: a könyvtár listázása is egy fájl átvitel!

Aktív FTP

- A kliens PORT port nr. paranccsal közöl egy portot - és a saját IP címét
- A szerver a saját 20-as portjáról erre kezdeményez egy kapcsolatot
- Ha nincs megadva port nr, akkor arra a portra kapcsolódik, ahonnan a control kapcsolathoz tartozó TCP kapcsolat felépült
- Aktív FTP gondot okoz csomagszűrő tűzfalaknál
  - A kliens gépekre bemenő TCP kapcsolatot nem engedélyezünk, mert biztonsági kockázatot jelent
  - Gyakori jelenség, hogy tűzfal mögül nem lehet ftp-zni

  - 1. megoldás: passzív ftp
  - 2. megoldás: a tűzfalnak „tudni” kell, hogy milyen ftp „PORT” parancs ment ki - ez az u.n. „Connection tracking”

Passzív FTP

- A kliens PASV parancsot ad
- A szerver válaszol IP címmel és ephemeral porttal. Az üzenet hasonlít a PORT parancshoz:
- A kliens építi fel az adatkapcsolatot egy saját ephemeral portjáról erre a portra
- Fájl vége: stream módnál (ez az általános) egyszerűen lezárja a küldő a TCP kapcsolatot
- A parancs csatornán küldött ABOR parancs hatására a küldő
  - Lezárja az adat TCP kapcsolatot,
  - A parancs csatornán jelzi, hogy végrehajtotta az abortot.
  - Még ez után is jöhetnek az adat TCP kapcsolaton csomagok!

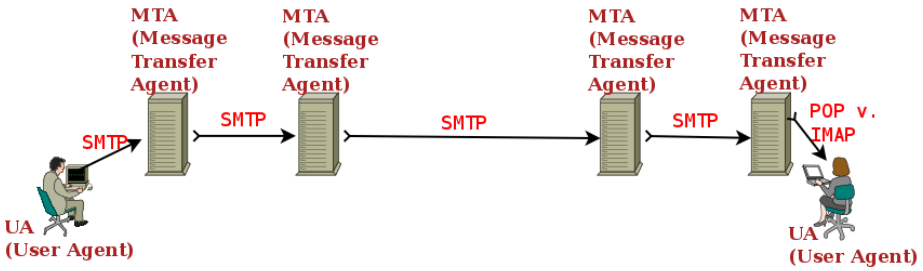
Anonymous FTP

- A USER parancsra közös választ ad a kliens: „anonymous”
- Nyilvánosan olvasható, de nem nyilvánosan írható fájlok közzétételének klasszikus módja
- A szerver jelszóként rendszerint a felhasználó e-mail címét várja
- Manapság messze a legjellemzőbb FTP használat

20. SMTP, ESMTP

SMTP - Simple Mail Transfer Protocol

- Szintén klasszikus protokoll
- RFC821, újabban RFC2821 majd RFC5321 (2008. október) finomította
- **UA - User Agent:** 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
  - 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 ugyanaz a TCP kapcsolat közvetíti, mint a parancsokat
- A felépülő full duplex TCP 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

SMPT parancsok

- HELLO fqdn.domain.nev
  - A kliens köszönése
  - Paraméter: a gép domain neve
  - Visszautasíthatják, ha hazudik (PTR rekord lookup)
  - A válasz 250 ha elfogadja a köszönést
- MAIL FROM: <mailbox@valahol>
  - Paraméter: a feladó címe <> közé zárva
  - Lehet üres
  - A „valahol" résznek érvényes FQDN névnek kell lennie
  - A „mailbox" rész nem feltétlenül egy személy
  - A paraméter értéke az **envelope from**
  - Ez látszódhat Return path:-ként a levelet olvasó számára
  - A kliens választ vár
- RCPT TO: <mailbox@valahol>
  - Ez is az „envelope" része: **envelope to**
  - A megadott címmel, mint címzettel bővül a boríték
  - Több címzett is lehet a borítékon, de legfeljebb 100
  - A <postmaster@domain> kötelezően érvényes cím kell legyen minden domain-ra
  - Más szolgálati címek pl: webmaster@domain
- A szerver 250-nel válaszol, ha elfogadja a címet
- Lehet hogy néhány RCPT TO:-t elfogad, másokat nem
- Jellemező hibakódok:
  - 550 Mailbox not found
  - 452 Too many recipients
  - 551 User has moved to
  - 553 User ambiguous
- Data
  - Ezzel jelzi a kliens, hogy levél következik, ami a „borítékban" van

  - Nincs paraméter
  - A szerver siker esetén 354-es kóddal válaszol (átmeneti állapot)
  - Ez után ASCII sorok következhetnek
    * Csak 7 bites ascii karakterek
    * Sorokra tördelve, a sorokat CR/LF választja el (0d0a)
    * A sorok legfeljebb 1000 (CR/LF nélkül 998) hosszúak lehetnek
  - Az utolsó sor egyetlen pontból áll
  - A ponttal kezdődő sorokat escape-elni kell
    * A küldő a ponttal kezdődő sor elejére még egy pontot illeszt
    * Ha a fogadó a sor elején pontot talál, eldobja
  - Az üzenet végén a szerver válaszol ilyenformán:
    * 250 valami-id message received and queued
    * Példák hibaüzenetre:
      · 554 too many hops, this message is looping
      · 452 Requested action not taken: insufficient system storage
  - Ha sikeres az átvitel, a felelősség a szerveré
  - A 250-es üzenetben a túlsó azonosítót szokás küldeni, ami szerint a naplóban utána lehet nézni az üzenetnek

- QUIT
  - A beszélgetés végét jelzi
  - A szerver 221-es kóddal válaszol, és bontja a kapcsolatot

- VRFY és EXPN
  - Egy cím ellenőrzésére szolgáló parancsok: létezik-e itt ilyen?
  - Biztonsági kockázat miatt kiment a divatból
  - Helyettesíthető egy RCPT TO-val, ami után bontjuk a kapcsolatot

- HELP
  - Humán használatra szolgáló parancs
  - Nem minden szerver ad választ

- ETRN
  - Extended TURN - fordítsunk: most Te küldj nekem, amit csak tudsz
  - Paraméter egy domain név
  - Ennek hatására a szerver a kicsi-domain.hu fele sorban álló leveleket el kezdi kiküldeni
  - Kis, betárcsázással csatlakozó intézmények domain-jénél használatos

- NOOP
  - Kivált egy 250-es választ
- RESET
  - A borítékot (envelope) kiradírozza: minden címzettre és feladóra vonatkozó információ üres lesz

Levél formátum

- A levél formátum szintakszisát írja le
- A levél ASCII sorokból áll, amiket CR/LF zár
- A levél sorai legfeljebb 998 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
- A fejrész mezőket (header fields) tartalmaz
  - A mező szerkezete: sor elején a név, kettőspont, whitespace, tartalom
  - Egy mező általában egy sorból áll
  - Folytatósort a sor eleji whitespace jelez
- Date:
  - A feladás idejét jelzi
  - Példa: Tue, 7 Dec 2004 15:07:54 +0100 (CET)
  - Fontos, hogy az időzóna is része
- To:
  - Azok akiknek/amiknek elsősorban szánjuk a levelet
  - valaki@valahol alakú cím
  - Több is lehet, vesszővel elválasztva
  - Kiegészíthető így: "Kiss Pista"
  - Az UA-n kitöltött To-ból envelope címzett lesz
- CC:
  - Carbon Copy, indigós másolat
  - Épp olyan mint a To: envelope címzett lesz
- BCC:

  - Blind Carbon Copy, titkos, rejtett másolat
  - Épp olyan mint a To: envelope címzett lesz
  - A DATA-val átküldött levélből kimarad
- From:
  - Azt a küldőt jelenti, akinek a nevében megy a levél
- Sender (A személyt, vagy más entitást, aki ténylegesen küldi)
- Reply-to: Kérem erre a címre válaszolj
- Message-id:
  - Egy véletlen szám, ami azonosítja a levelet
  - Általában @ után a rendszert azonosítja, ahol a levél keletkezett
- References:
  - Message-id-k sorozatát tartalmazza, amikre hivatkozik a levél
  - News csoportoknál vezették be
  - Levelezési listáknál is hasznos: **thread**-ek keletkeznek
- Subject:
  - Nagy illetlenség kitöltetlenül hagyni
  - Rövid, és lényegre törő legyen
- Return-Path (Az envelope sendert mutatja)
- Received:
  - A közvetítő MTA-k által betett mező
  - Segítségével nyomon lehet követni, hol és mikor járt a levél
  - Ki lehet szűrni a körbe keringő leveleket
  - Az egyes MTA-k hop count korlátozást használnak
  - Az időzóna fontos része a received soroknak

ESMTP - Extended SMTP

- 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

Kitől fogad el egy SMTP szerver levelet?

- Régen nem volt szokás korlátozni
  - Az ilyen konfigurációt nevezzük nyílt relay-nek
  - Az ilyet gyorsan felfedezik, spam-ek küldésére használják
- Olyan **IP címekről**, akik számára ő a „sarki postaláda” (kiinduló MTA)
  - Egy subneten levő IP címek
  - Konfigurációs paraméterben megadott címek
- Olyan **címzettek** számára, akiknek ő a „kapunál levő postaláda” (cél MTA)
  - Akiknek ezen a gépen van a postafiókjuk
  - Akik számára MX
  - Konfigurációs paraméterben megadott domain-ok

21. Levél, internet message formátum: RFC2822/822, MIME

MIME - Multipurpose Internet Mail Extensions

- 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

A multipart üzenetek valók a mellékletek (attachment) hordozásra

- Pl.: multipart/mixed, multipart/alternative
  - Paraméter: boundary
- 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

  - gy 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 → =e4
  - Az egyenlőségjelet is kódolni kell: = 3d
  - A recode unixos utility nem csak karakterkészletek, hanem kódolások közt is tud konvertálni (surface-nek nevezi)
- 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

VERP - Variable Envelope Return Path

- **1. probléma**
  - Listáról menő levelek címzettjei sokszor továbbítják máshova a levelet pl. .forward fájl vagy alias segítségével
  - Ez többszörös mélységben is előfordulhat
  - Ha valahol hiba van, nem lehet tudni, hogy ki is volt az eredeti címzett
  - E-mail címek megszűnnek, az ide átirányított más címek maradnak
  - A levelezési listákon is megmarad a cím
  - A visszapattanó levélből nem látható, hogy ki is volt a listatag
- **2. probléma**
  - A felhasználók sok címet használnak, ezeket egymásra irányítják
  - Nem is tudják, hogy az egyes levelezési listákra melyik címmel iratkoztak fel
- **Megoldás: VERP - Variable Envelope Return Path**
  - A címzett címének egy részét betesszük a return address-be = az envelope mail from: címbe
  - Például: owner-listname@listaserver.hu helyett a „MAIL FROM” ban owner-listname+nemecsek=palutca.hu@listaserver.hu lesz a kimenő levél envelope-jában

  - Gondoskodni kell róla, hogy az owner-listname+akarmi@listaserver.hu címekre menő leveleket (a visszapattanó levelet) mind ugyanaz a program kapja, és megfelelően cselekedjen

Hátrány: ha egy smtp szervernek küld a lista szerver több levelet, akkor kénytelen külön-külön küldeni

RBL-ek

- RBL = Realtime Blackhole List
- IP cím gyűjtő listák – ezekről az IP címekről a lista gazdája szerint nem tanácsos levelet elfogadni
  - Bizonyítottan spamelő IP címek
  - Dialup címek
  - Botnet címek
- DNS-en alapul: a 12.34.56.78 IP cím akkor van a listán, ha a feloldható a 78.56.34.12.<list-specific-suffix> név
- Sok RBL van. Lista a pillanatnyi állapotról: http://dnsbl.inps.de/analyse.cgi?lang=en
- A fogadó MTA a konfigurációja szerint eldöntheti, hogy mit csinál, ha a küldő MTA szerepel egy (vagy több) RBL-en
  - Visszautasíthatja az SMTP kapcsolatot
  - Megjelölheti a levelet mint spam gyanúsat
  - Figyelembe veheti úgy, hogy spam-gyanú-faktort növelik
- Gyakori, hogy egy fogadó MTA több RBL-t is használ

Internetes levelezés biztonság

- Az internetes levelezés olyan, mintha mindig nyílt levelezőlapon küldenénk minden levelet
  - A közbülső MTA-k, sőt a közbülső routerek, tűzfalak gazdái gond nélkül tudják olvasni sőt módosítani a leveleinket
  - Van ellenszer évtizedek óta: PGP
  - Mindenkit bíztatok, hogy használja

22. Distance vector routing protokollok, RIP

Routing

- Útvonalválasztás
- Csomópontokban merre irányítsuk a forgalmat?
- Csomag/üzenet/levél/fájl/... irányítás

Hol is kell?

- **IP szinten**
- Adatkapcsolati szinten: (STP) Spanning Tree Protocol
- Alkalmazás szinten: például mail routing

Matematikai modell

- Irányított, súlyozott gráfban minimális költségű utat keresni
- Két pont közt keressük az optimumot - lehetne általánosabban is !

Distance vector protokollok

- Minden csomópont az összes szomszédjának elküldi a teljes routing tábláját: **hirdetéseket.**
- A routerek összevetik a jelenlegi routing táblájukat a kapott hirdetésekkel: amit egy szomszéd router k távolságra lát, azt én k+1 távolságra látom és **felé** route-olom.
- Ha stabil a hálózat állapota, akkor egy idő után a routing táblák is stabilizálódnak minden csomópontban: **konvergencia.**

Ip routing protokollok

- Minden internetben forgalmazó eszközön jelen van
- Melyik CIDR blokkot merre rout-olunk
- 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

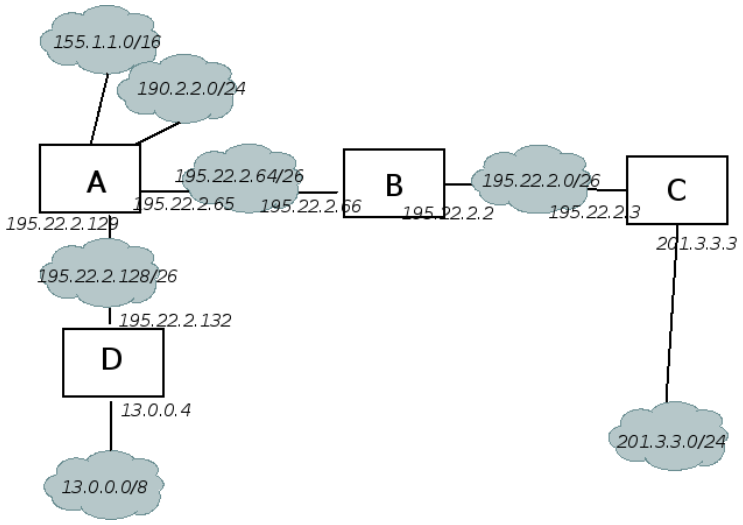
Distance vector protokollok

- 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
- 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

Counting to infinity - Split horizon

- Ha B és C közt megszakad a kapcsolat, B A-tól még mindig hallja 2-es count-tal hirdetni C-t
- **Split horizon:** ha A csak B-től hallja C-t, akkor B fele nem hirdeti
- **Poisoned reverse**
  - Ha A csak B-től hallja C-t, akkor visszafele **végtelen** költséggel hirdeti
- Nem szűnteti meg teljesen a végtelenig számlálást
- A és B egymásnak fogja hirdetni D elérhetőségét, akkor is ha már C és D közt nincs kapcsolat

RIP - Routing Information Protocol



A		
Dst	Hop	Next
155.1.1.0/16	1	-
190.2.2.0/24	1	-
195.22.2.64/26	1	-
195.22.2.128/26	1	-
195.22.2.0/26	2	195.22.2.66
201.3.3.0/24	3	195.22.2.66
13.0.0.0/8	2	195.22.2.132

B		
Dst	Hop	Next
155.1.1.0/16	2	195.22.2.65
190.2.2.0/24	2	195.22.2.65
195.22.2.64/26	1	-
195.22.2.128/26	2	195.22.2.65
195.22.2.0/26	1	-
201.3.3.0/24	2	195.22.2.3
13.0.0.0/8	3	195.22.2.65

C		
Dst	Hop	Next
155.1.1.0/16	3	195.22.2.2
190.2.2.0/24	3	195.22.2.2
195.22.2.64/26	2	195.22.2.2
195.22.2.128/26	3	195.22.2.2
195.22.2.0/26	2	-
201.3.3.0/24	1	-
13.0.0.0/8	4	195.22.2.2

D		
Dst	Hop	Next
155.1.1.0/16	2	195.22.2.129
190.2.2.0/24	2	195.22.2.129
195.22.2.64/26	2	195.22.2.129
195.22.2.128/26	1	-
195.22.2.0/26	3	195.22.2.129
201.3.3.0/24	4	195.22.2.129
13.0.0.0/8	1	-

- Distance-vector alapú
  - Destination: egy cél hálózat
  - Cost (él súly): a küldő távolsága a céltól (hop count)
  - Source: a küldő router ID-je
- Split horizon, poisoned reverse
- Triggered update
- UDP 520 port
- Végtelen = 16
- RIP1 - IP broadcast
- RIP2 - IP multicast: 244.0.0.9

RIP header

- 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:
  - Address family: IP-nél 2
  - IPv4 cím: hálózat, vagy host cím
  - metric: távolság
- Egy RIP2 entry:
  - Address Family = 2, ha IP v4
  - Route Tag: az AS (Autonomous System) ID, ha ilyet is tud a küldő
  - Subnet mask: a hirdetett címhez/tartományhoz tartozó maszk – RIP1-nél nincs!
  - Next hop: én erre az IP címre route-olom ezt

RIP2 autentikáció

- Sepciális RIP entry: address family = 0xFFFF
- Egy jelszó megy 16 byte-on a maradék RIP entry részben

RIP2 autentikáció továbbfejlesztés

- Az eredeti RIP2 autentikáció gyenge
- MD5 vagy SHA1 hash-t képez
- 16 byte-os shared secret-ből és az üzenetből

Időzítések

- **Update:** 30 sec. Ennyi időnként szól a szomszédoknak
- **Timeout:** 180 sec. Ha ennyi ideig nem kap valahonnan update-et, végtelenre (16) állítja az arra menő utakat
- **Garbage collection:** 120 sec. A törlésre szánt (végtelen költségű) utak ennyi idő múlva valóban törlődnek

Miért a RIP?

- PRO: Egyszerűen konfigurálható, viszonylag gyors konvergencia, rövidek a count to infinity loop-ok
- CONTRA: Nem használható nagy hálózatban: legfeljebb 15 hop

23. Autonóm rendszerek, BGP

Autonóm rendszer - Autonoumous System (AS)

- Routing szempontjából önálló entitás
- Pl. egy-egy szolgáltató által felügyelt hálózat
- Egy azonosítót rendel hozzá: AS number
  - Az AS number eredetileg 16 bites, 2006 óta 32 bites lehet
  - Fontos, hogy világállandó legyen
  - Európában a RIPE osztja
  - A világon máshol: ARIN, APNIC, LACNIC, AFRINIC
- CIDR hálózatok, **prefixek** egy halmaza
  - Az AS ezeket tartalmazza
  - Az AS ezeket és a tanultakat hirdeti
  - Route aggregation: össze lehet vonni CIDR blokkokat
    - Pl. a 193.224.0.0/15-at egyben tudja hírdetni a HBONE AS
    - Ebbe tartozik az ITK blokkja, a 193.225.109.0/24
- NIIF AS NR: 1955
- AS fajták
  - Stub: egy másik AS fele van kapcsolata
  - Multihomed: több AS fele van kapcsolata
  - Transit: nem csak a saját forgalmát bonyolítja

Interior Gateway Protocol (IGP), Exterior Gateway Protocol (EGP)

- Egy AS-en belüli routing: **IGP**
- AS-ek közti routing: **EGP**
  - BGP

BGP - Border Gateway Protocol

- 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
- 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 tartja 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
- Route reflector
  - Nem kell full-mesh: mindenki csak a közvetítővel beszél
- A BGP hirdetéseket nem ismétlik periodikusan (RIP, OSPF igen)
- Egy BGP üzenet 1 destination fele csak 1 route-ot hirdet
- 2014-ben már több mint 500 ezer hálózatot hirdetnek a backbone routerek!

BGP üzenetek

- open
  - Ki vagyok? (AS number, Router ID)
  - Hold time (Ennyi időnként kell halljak tőled üzenetet, különben lebontom a kapcsolatot)
- update (Route-ok küldése)
- keepalive (Nem tartalmaz routing információt (különbözik RIP, OSPF-nél!))
- notification (hibaüzenet, Ez után bomlik a BGP kapcsolat)

- route-refresh (Kérlek küldd el a teljes routing tábládat)
- Erre gyakorlati példák a 12. előadás végén találhatóak

Whois

- Egyszerű, emberi fogyasztársra közvetlenül alkalmas információk hálózati alanyokról
- Egy ilyen alany: **objektum**
  - Hálózat
  - AS
  - Domain
- Egy tulajdonsága: **attribútum**
  - Tulajdonos neve
  - Felelős neve (Adminisztratív, technikai)
  - E-mail (?) cím
- Példa: whois -T aut-num AS5377 -h whois.ripe.net
- Egyetlen kérdés/válasz
- TCP 43-as port

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

24. Link state routing protokollok, OSPF

Link state protokollok

- 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
- Fontos, hogy minden router-hez eljusson az információ, elárasztják a hirdetések a router-eket: **flood**
- Mindenki újraszámol

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

- Nem UDP, nem TCP: saját IP protokol: 89
- Saját multicast csoportok: 244.0.0.5 - all SPF routeres, 244.0.0.6 - all DR routers

OSPF folyamatok

1. Van ott valaki?
    - Neighbor discovery - HELLO üzenetek
2. Ki mit lát?
    - Database description és link state request/advertisment üzenetek
    - Minden router-hez el kell juttatni: flood
    - Mindent nyugtázni kell - ACK üzenetek
3. Később csak link state update - ack

4. A router felépíti az egész hálózat fáját, és alkalmazza a Dijkstra algoritmust
5. A router rútol!

OSPF üzenetformátum

- 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: 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
  - Stub area: amelyik nem fogad külső routing információt
    - Nincs átmenő forgalom, csak végállomás
- Area ID - erre az area-ra vonatkozik a csomag
- 32 bit mint az IP cím, ezt is dotted decimal formában szokás megadni
- Checksum - az IP -nél szokásos összeadás
- Authtype: különböző area-kban különböző lehet
- 0: nincs autentikácaó
- 1: jelszó
- 2: Kriptográfiai autentikáció
  - Az egész csomagból, egy sorszámból és a jelszóból számolva
- Authentication a jogosultsági információ, jelszó vagy MD5/SHA1/SHA256 szumma

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

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

Költség - cost: egy router egy link-jének a jellemzője

- Konfigurálható paraméter
- Alapértelmezés: az interfész sebességének reciproka$\cdot 10^8$

LSA - link state advertisement

- Egy linkről szóló információ
- Nyugtázandó
- Továbbítják minden szomszédnak
- Update: csak a különbség az előzőhöz képest (inkrementális update)
- Periódikus update: 30 percenként

- 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
- 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

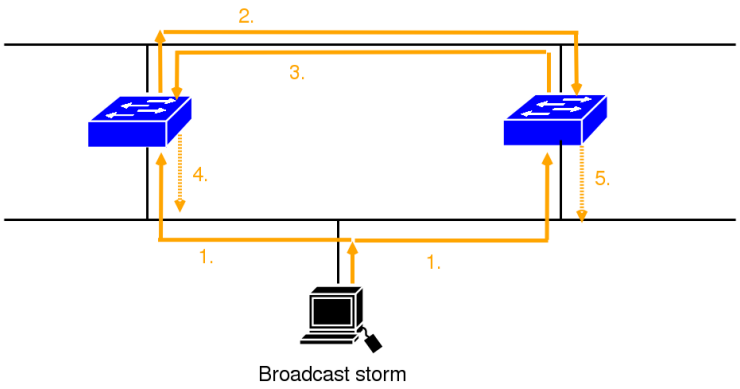
- OSPF gyorsabban konvergál
- OSPF árnyaltabb: figyelembe vesz TOS-t, sávszélességet stb.
- OSPF nagyobb hálózatban is használható
- OSPF kevesebb hálózati forgalmat generál
- OSPF nem „flat” hierachia van benne (area-k)
- RIP egyszerűbb: könnyebben adminisztrálható
- RIP kisebb erőforrás igényű a router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

25. Spanning tree protokoll

STP - Spanning Tree Protocol

- DEC találmány, IEEE átvette: 802.1d
- Collision domain
  - Olyan elemek a hálózaton, akik egy ethernet „folyosón” vannak
  - Az egy collision domain-ban levő eszközök nem adhatnak egyszerre
  - Egyetlen pont-pont, full duplex ethernet összeköttetés önmagában két collision domain!
  - Ha egy eszköznek (pl. switch) több portja van ugyanabban a collision domain-ban, akkor hallja a saját adását is a másik portján
- Broadcast domain
  - Switch-ekkel (bridge-ekkel) összekötött collision domain-ek
  - A broadcast ethernet címre (ff:ff:ff:ff:ff:ff) menő csomagokat mind hallják
  - VLAN-ok definiálásával switch-elt hálózaton több broadcast domain-t is kialakíthatu

Broadcast storm



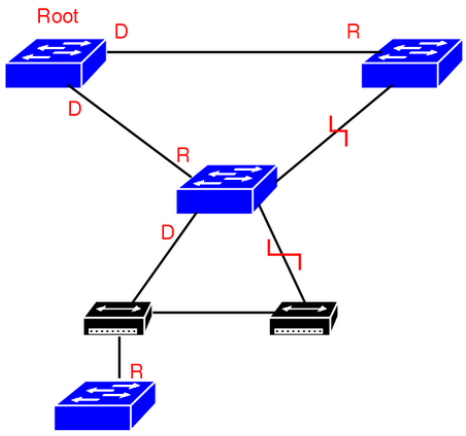
STP ethernet típus: BPDU (Bridge Protocol Data Unit)

- Kissé anakronisztikus a bridge szó...
- Nem ethernet II, hanem 802.3
- A BPDU-kat periódikusan küldik a switch-ek
- 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
- Ha több port is szóba jön egy switch-en, akkor a kisebb id-jű nyer
  - Az ehhez tartozó port: root port
  - A root switch kivételével minden switchnek egy pillanatban pontosan 1 van

Designated bridge, designated port

- Egy collision domain-t a root-tal összekötő bridge/port
- Minden collision domain-ban egy adott pillanatban pontosan 1 van



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 (vicces!)
- Gyári érték, de konfigurálható
- Célszerű konfigurálni, hogy ne egy periférikus switch legyen a root!

BPDU szerkezet

- Az általam root-nak tudott BID
- Az root-hoz tartozó teljes út költsége
- Az én BID-em
  - Jelentősége van, ha egy collision domain-ban több bridge van
- Port ID, amin küldöm ezt a BPDU-t
  - Jelentősége van, ha egy switchnek egy collision domain-ban több lába van
- Maximum age: ennyi idő után felejtsd el ezt az információt
  - Jellemző érték: 20 sec
  - Legalább ennyi időnként jönnek a BPDU-k
- Hello Time: a root ennyi időnként küld BPDU-t
- Forward Delay: ennyi ideig van listening és learning állapotban

Port állapotok – a működést határozzák meg

- Blocking
  - Bekapcsolás után
  - Tartalék üzemmódra kapcsolt port (nem nyert az alagoritmusban)
- Listening – blocking state után, BPDU-kat figyel
- Learning – listening után, fogad és küld BPDU-kat, felépíti a táblázatait, MAC címeket tanul a portjain
- Forwarding – konfigurálás után úgy látja, hogy ez részt vesz a spanning tree-ben
- Disabled – az algoritmus által kikapcsolt, hibás

Port szerepek – az STP algoritmusban betöltött funkciók neve

- Root port
- Designated port
- Alternate port

STP parancsok

- #sho spanning-tree root - merre van a root?
- #sho spanning-tree active - milyen portokon látszik a root?
- #sho spanning-tree blockedports - Milyen blocked portok vannak?

Rapid Spanning Tree - RSTP

- IEEE 802.1w
- Az STP lassan (perces nagyságrend) konvergál
- A fő cél, ezt az időt lerövidíteni
- Az STP továbbfejlesztése
- Ugyanaz az üzenet (BPDU) formátum
- A version mező értéke (2) jelzi, hogy RSTP
  - Együtt tudnak működnik STP-t és RSTP-t beszélő switch-ek
- Új fogalmak
  - Edge port – ahol nem lehet bridge, nem küldünk BPDU-t
  - Alternate port – blocking állapotban levő, a root-hoz más utat jelentő port
  - Backup port - blocking állapotban levő, a fa levelei felé vivő port
- Minden bridge HELO időnként (default 2 sec) küld BPDU-kat
  - STP-nél csak a root kezdeményez, és ezt relézik a többiek
- Ha egy bridge 3· HELLO ideig nem kap a szomszádjától BPDU-t, halottnak tekinti (egyfajta keep alive)

Villámkérdések (minta)

- Ephemeral port
  - 19. tétel FTP, a 70. oldal
- CIDR
  - Classless Inter-Domain Routing
  - Túl sok hálózati cím, 9 havonta duplázódott
  - A módszer lényege, hogy a router-ekben egy bejegyzés nem csupán egy, hanem több hálózat felé való továbbítási irányt írjon le
  - Azaz a célpontok ne egyes hálózatok, hanem hálózatok csoportjai legyenek
  - Így az azonos irányba esô hálózatok egy bejegyzésben megjeleníthetôek és nem kell mindegyikhez külön-külön letárolni a továbbítási irányt
- RFC
  - 2. tétel RFC-k, Internet szervezetek, az 5. oldal
- IANA
  - 5. tétel IP, a 15. oldal
- IAB
  - Internet Architecture Board (IAB)
  - Hálózati protokoll szempontok
  - Felügyeli a létrehozott internet szabványokat
  - Felelős az RFC dokumentum sorozatért
- Unicast/multicast/broadcast/anycast/simplex/duplex/half duplex
  - **Unicast:** 1 küldő 1 címzettnek, például telefon
  - **Multicast:** 1 küldő többeknek, például rádió
  - **Broadcast:** 1 küldő mindenkinek, például sziréna
  - **Anycast:** 1 küldő egy bizonyos értelemben közellevő bármelyiknek, például mentők telefonon
  - **Simplex:** két partner közti egyirányú kommunikáció, például rádió
  - **Half duplex:** két partner közti kétirányú kommunikáció, de egy időben csak az egyik irány működhet. Például telex.
  - **Full duplex kommunikáció:** két partner közti kétirányú kommunikáció, egy időpillanatban mindkét irányban működhet.

  - **CRC = Cyclic Redundancy Check,** ethernet csomagoknál FCS-nek, Frame Control Sequence-nek is szokás nevezni. Hibaellenőrző eljárás, illetve ennek az eljárásnak az eredménye. Az adatfolyamot polinomként fogjuk fel, és egy meghatározott számmal osztjuk. Az eredményt (a maradékot) a küldő az adattal küldi, a vevő ellenőrzi. Bithibák (elvesztés, beszúrás, átbillenés) felismerhetők ezáltal.
  - **Flow control – folyamvezérlés:** a fogadó fél eljárása, amivel a küldő adatfolyamának ütemét befolyásolhatja: például terminál klaviatúrán CTRL/S (stop) CTRL/Q (continue)
  - **Little endian – big endian.** Az átvitel során egy byte bitjei sorban jelennek meg a médiumon. Általában a kisebb helyiértékű bit az első, a legnagyobb helyiértékű az utolsó. Ez a little endian sorrend. Az ethernet CRC bitjei big endian sorrendűek. Big endian sorrendűek az IP csomagok byte-jai is.
  - **PDU – Protocol Data Unit:** egy adategység, amit a egy kommunikációs protokoll értelmez
  - **SDU - Service Data Unit:** az adategység, amit a felette levő réteg számára közvetít

- CSMA/CD
  - 3. tétel Klasszikus ethernet, 6. oldal
- Late collision
  - Ha elindul egy frame, a hálózat legtávolabbi pontján is érzékelni kell mielőtt véget ér
  - Nem lehetne észrevenni az ütközést: **late collision**
- Exponential backoff
  - 3. tétel Klasszikus ethernet, 6. oldal
- Little endian/Big endian
  - Villámkérdések, 99.oldal
- Spanning tree
  - 25 tétel Spanning tree protokoll, 94. oldal
- VLAN
  - 4. tétel Point to point ethernet, fast ethernet, full duplex ethernet, VLAN-ok, 12. oldal
- Loopback interfész
  - 6. tétel PPP, PPPoE, 21. oldal

- MTU
  - 6. tétel PPP, PPPoE, 21. oldal
- TOS
  - 5. tétel IP, 12 oldal
- TTL - IP csomagnál, DNS rekordnál
  - IP csomagnál: 5. tétel IP, 14. oldal
  - DNS rekordnál: 14. tétel DNS működése, 46. oldal
- Netmask
  - 5. tétel IP, 15. oldal
- NAT
  - 5. tétel IP, 17. oldal
- ARP cache poisoning
  - 7. tétel ARP, RARP, 23. oldal
- Path MTU discovery
  - 9. tétel ICMP, ICMP hibaüzenetek, 30. oldal
- ICMP redirect
  - 9. tétel ICMP, ICMP hibaüzenetek, 31. oldal
- BGP dampening
  - 23. tétel Autonóm rendszerek, BGP, 88. oldal
- AS
  - Autonom System
  - 23. tétel Autonóm rendszerek, BGP, 87. oldal
- UDP lite
  - 11.tétel UDP, 35. oldal
- IGMP membership query/membership report
  - 13. tétel IGMP, PIM, 39. oldal
- SOA, NS, A, PTR, MX rekord
  - 15. tétel DNS rekordok 49-51. oldal

- FQDN
  - 14. tétel DNS működése, 48. oldal
- Glue rekord
  - 15. DNS rekordok, 50. oldal
- Lame név szerver
  - 15. DNS rekordok, 50. oldal
- Primary (master), secondary (slave) DNS szerver
  - 14. tétel DNS működése, 47. oldal
- Caching only DNS szerver, autoritatív név szerver
  - 15. tétel DNS rekordok, 49. oldal
- DNS cache poisining
  - 14. tétel DNS működése, 46. oldal
- TCP syn flood támadás
  - 17. tétel TCP, 58. oldal
- TCP aktív/passzív open
  - 17. tétel TCP, 57-58. oldal
- TCP aktív/passzív close
  - 17. tétel TCP, 58. oldal
- TIME-WAIT (2MSL wait) állapot
  - 17. tétel TCP, 60. oldal
- TCP reset támadás
  - 17. tétel TCP, 61-62. oldal
- tcp-wrapper
  - 17. tétel TCP, 62. oldal
- TCP retransmission timer (RTO)
  - 18. TCP flow control, 65. oldal
- Delayed ack
  - 18. TCP flow control, 65. oldal

- Nagle algoritmus
  - 18. TCP flow control, 66. oldal
- Slow start
  - TCP flow control, 66. oldal
- Persist timer
  - TCP flow control, 68. oldal
- Passzív ftp
  - 19. tétel FTP, 73. oldal
- UA, MTA
  - 20. tétel SMTP, ESMTP, 74. oldal
- DHCP lease
  - 8. tétel BOOTP, DHCP, 27. oldal
- PXE (Preboot Execution Environment)
  - 8. tétel BOOTP, DHCP, 27. oldal
- Proxy ARP
  - 7. tétel ARP, RARP, 22-23. oldal