1. tétel - ISO/OSI

Bevezető

- Hálózati rétegek
- sok különböző
- egységesítés

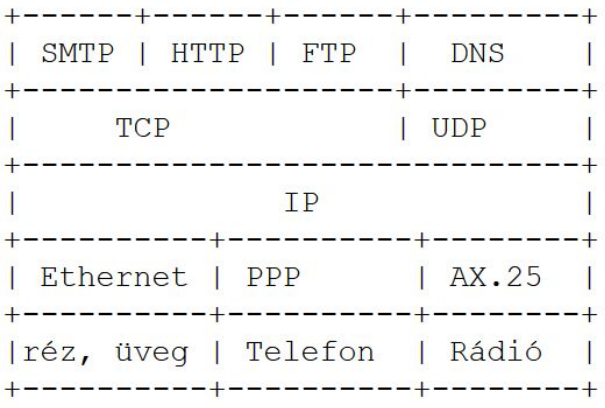
OSI rétegek

- csak ajánlás

- Alkalmazási réteg
  - pl. fájlátvitel, levelezés
- Prezentációs réteg
  - tömörítés/ titkosítás
  - Manapság összevonva az elsővel
- Session
  - kapcsolat felépítés/bontás
  - Manapság összevonva a negyedikkel
- Transzport
  - hibamentesítés, csomagok rendezése
- Hálózati
  - útvonalkeresés, routing
- Adatkapcsolati
  - csomagokat különít (frame), hibamentesítés
  - fizikaival összevonva: ethernet
- Fizikai
  - bitfolyamot közvetít valahogy pl. drót

TCP/IP

- protokoll család
- szabad és nyílt
- nagyon elterjedt



2. tétel - Ethernet

Bevezetés

- Xerox fejlesztés
- 70-es évek
- rengeteg szabvány
- adatkapcs + fizikai

CSMA/CD

- CS = Carrier Sense
    - nem adok ha más ad
- MA = Multiple Acces
    - minden csomaghoz hozzáférek, de csak az enyém érdekel
- CD = Collision Detection
    - Ütközés kezelés (backoff delay, exponential backoff)

Topológia

- busz, és csillag

Fizikai réteg

- Többféle kábel - különböző sebességgel (10base2, 100Base-TX, stb.)
- első a sebesség
- Optikai kábelek
    - MMF - 50-100 mikron, fény pattog, épületen belül
    - SMF - 10 mikron, fény egyenes, nagyobb táv

Media Acces Control

- Ethernet alréteg
- Biztosítja a címzést, vezérli a hozzáféréseket
- MAC-cím
    - IEEE
    - gyártók

Frame

- adatátvitel framekben történik

- Preambulum
    - szinkronizáláshoz (01 felváltva)
- Start Frame Delimiter
    - preambulum végét jelzi
- Cél
    - cél MAC címe
- Forrás
    - forrás MAC címe
- Hossz vagy típus
- Adat
- Padding
    - pótlás, ha nem elég az adat - 46 byte
- CRC

Repeater / HUB

- ethernet szegmensek összekötésére
- broadcast
- hibás frame is átmegy

SWITCH / Bridge

- Hibás nem megy át
- megtanulja a MAC címeket
- unicast
- biztonság

Ethernet Flow Control

- PAUSE frame
- szólok, hogy lassítson
- új PAUSE újraindít

3. tétel - PPoE, VLAN

PPoE

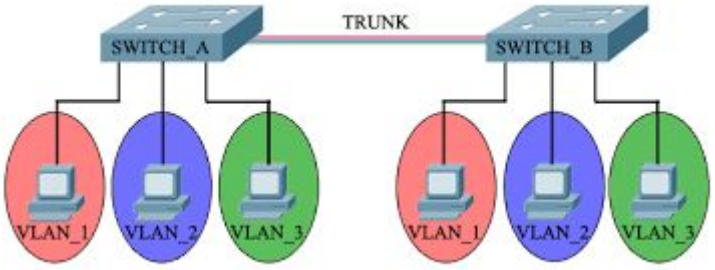
- PPP over Ethernet
- ADSL-re jellemző, telefonvonalak
- 2 fél között
- 2 állapot van
    - Discovery
        - Kapcsolat teremtés a másik féllel
    - Session
        - A kapcsolat léterjötte után

Full duplex Ethernet

- csak PPP kapcsolatnál
- nincs CSMA/CD
- effektívebb

VLAN

- egy ethernetet logikailag több részre osztok
- Előny: biztonság

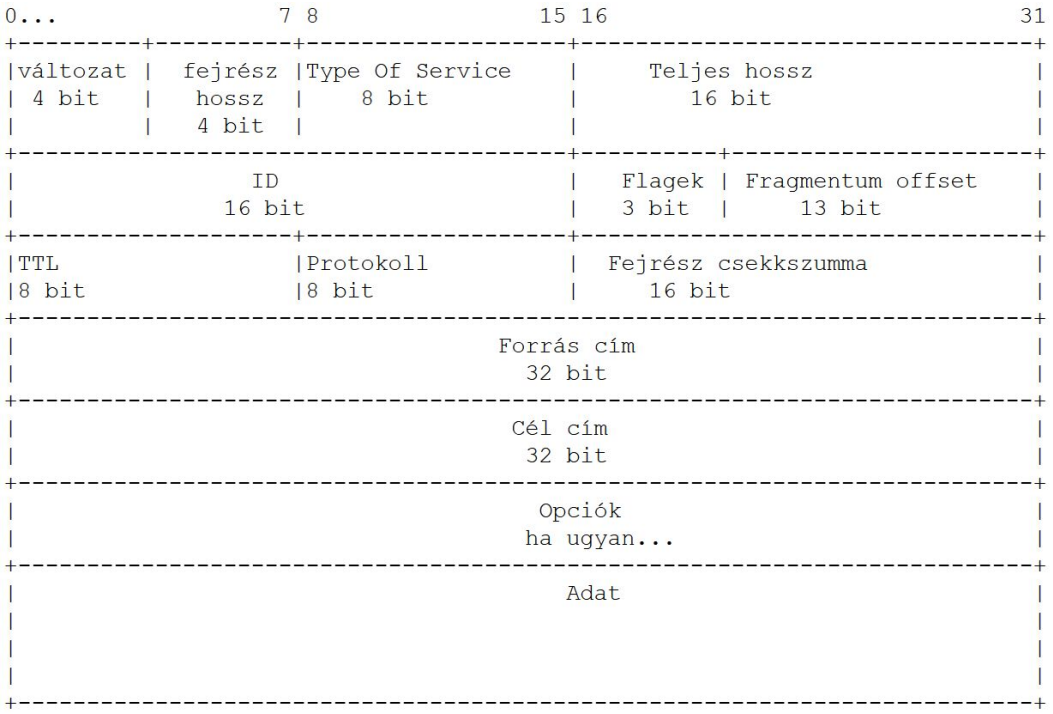


4. tétel - IP

IP

- hálózati réteg
- kapcsolatteremtés
- IP cím - hálózati azonosítás
- Tulajdonságok
    - Nem kapcsolatorientált
        - nem kell létrehozni a kapcsolatot, csak kikúrjuk az éterbe
    - Nem megbízható
        - többszöröződés, sorrend, sérült csomag, célba érés

Header



- TOS
    - csomag hálózaton belüli továbbítása
- Hossz
    - max 65535
- ID
    - egyedi azonosító, véletlen szám
- FLAG
    - csomag darabolásnál van szerepe

- Fragmentum offset
    - a csomag helye, ha darabolt
- TTL
    - hoppok száma
- Protocol
    - IP feletti protocol
- Checksum
- Forrás és célcím
    - IP

IP osztályok

Osztály	Tartomány	Netmask	Egy hálózatban kioszthető cím
A	**0**.0.0.0 - **127**.255.255.255	/8 azaz 255.0.0.0	2^24 > 16 millió
B	**128**.0.0.0 - **191**.255.255.255	/16 azaz 255.255.0.0	2^16 = 65536
C	**192**.0.0.0 - **223**.255.255.255	/24 azaz 255.255.255.0	2^8 = 256

- Van D meg E osztály is - multicast, jövőbeli megoldások

CIDR

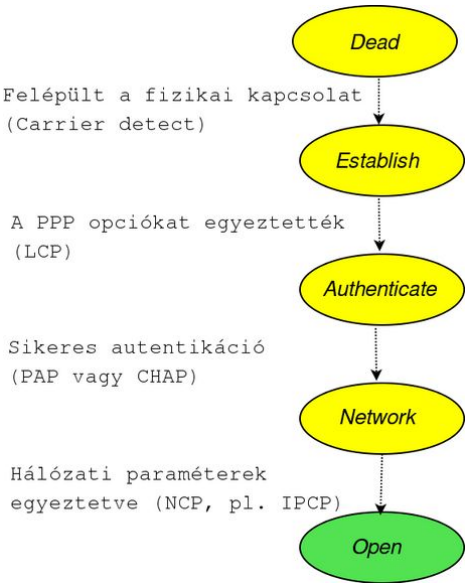
- ennek megszüntetése
- lassabban fogyjanak el az IP-k
- hálózatot a lekisebb címével adom meg

5. tétel - PPP, PPPoE

PPP

- két fél között
- IP mellett gyakran használt
- telefon vonalak
- Feladata
    - Frame elkülönítés
    - Hiba felderítés
    - “multiplatform”
    - hatékonyság

PPP állapotok



LCP

- Link Control Protocol
- PPP része
- Kapcsolat felépítés/bontás

PAP

- Password Autenthication Protocol
- Kliens küld: azonosító + jelsző
- Szerver válaszol

- Nem biztonságos

CHAP

- jobb mint a PAP
- Challange Handshake …
- Három lépés:
    - véletlen számot küld a szerver
    - a kliens a kapott szám alapján képez választ
    - szerver ellenőrzi

6. tétel - ARP, RARP

ARP

- Adress Resolution Protocol
- Hogyan tudja meg a hálózati címét egy állomás?
    - mi a baj a MAC címmel?
    - Ip-t összerendeljük a MAC-kel
- Etherneten elsősorban

ARP kérés nem létező hostra

- ha nincs válasz
- próbálkozunk párszor
- jelezzük a hibát

Működés

- Target Hardware Adresst üresen hagyom
- Broadcast
- Ha valaki felismeri a saját IP-jét, akkor kitölti a mezőt és visszaküldi

ARP cache

- hogy ne kelljen folyton ezt csinálni, ezért megtanuljuk a címeket

Proxy ARP

- ha a másik hálózatban van akit keresünk
- broadcastre nincs válasz
- amúgy sincs értelme ha másik hálózatban van
- a router azt hazudja magáról hogy ő az, és rajta keresztül küldünk
- Default Gateway: ha eleve tudom, hogy más hálózatban van, neki küldöm és rábízom magam.

UNARP

- megtanultam hol van valaki, de átkerül máshova
- ilyenkor küld egy broadcastet, hogy mindenki törölje, már nincs ott

ARP gratiutous

- foglalt e már az IP
- nyomok rá egy ARP-t, ha van válasz foglalt

ARP cache poisoning

- Man in the middle (ettercap)
- IP nem biztonságos
- ARP felülírás
- védekezés : bevasalás

RARP

- Mi a saját IP-m?
- Legegyszerűbb megoldás
- metódus:
    - Broadcast: nem tudom az IP-m
    - RARP szerver: szabad IP-k közül visszaad egyet
- hátrány:
    - csak IP, ez kevés (defaultgateway, netmask…)
    - honnan tudja melyik hálózatban van

7. tétel - BOOTP, DHCP

BOOTP

- RARP helyett - > jobb
- BOOT protocol
- AZ IP-n kívűl ad
    - netmask
    - router cím
    - bootserver cím
    - boot file name stb.

DHCP

- mégjobb
- BOOTP-vel kompatibilis, formátum megegyezik
- dinamikusan tudsz vele IP-t osztani
    - idő után lejáró IP-k
    - Ad egy engedélyt egy IP-re
    - így nem fogynak el
    - kliens kilép: felszabadul az IP

PXE

- még tovább (csak INTEL-en)

8. tétel - ICMP vezérlő

ICMP

- Internet Control Message Protocol
- IP fölött és IP alatt (kiegészítő protokoll)
- hibaüzenetek, szolgálati üzenetek stb.

ICMP üzenetek

- hibaüzenetek / vezérlő üzenetek

ICMP hibaüzenetek

- Sose ad hibaüzenetet
  - IP broadcast, multicast
  - ICMP hibaüzenet
  - IGMP üzenetek
  - IP csomag, aminek rossz a forráscíme
- Hiba üzenet tartalmaz
  - IP header
  - 8 byte IP aadat
  - portok

ICMP hiba példák

```
network unreachable (Section 9.3)
host unreachable (Section 9.3)
protocol unreachable
port unreachable (Section 6.5)
fragmentation needed but don't-fragment bit set (Section 11.6)
source route failed (Section 8.5)
destination network unknown
destination host unknown
source host isolated (obsolete)
destination network administratively prohibited
destination host administratively prohibited
network unreachable for TOS (Section 9.3)
host unreachable for TOS (Section 9.3)
communication administratively prohibited by filtering
host precedence violation
precedence cutoff in effect
```

Destination unreachable

- hibaüzenet
- router küldi, szól, hogy a cél nem elérhető

9. tétel - ICMP/2

ICMP adress mask request/reply

- RARP-pl kapcsolatban használatos
- arra hogy megtudjuk a netmaskot
- nem használjuk, DHCP jobb

ICMP timestamp request/reply

- saját idő, fogadási idő
- hálózat sebessége
- óra szinkronizálás és beállítás

NTP

- NTP szerver - referencia időhöz szinkronizálunk

Router solicitation / advertisement

- ha nem tudjuk ki a gateway, küldünk egy multicastet a routereknek
- ők megmondják (advertisement)

Path MTU discovery

- mi a legkisebb MTU a cél felé
- ez változhat
- küldés optimalizáláshoz

Source Quench

- Torlódás megoldásához
- tetszőlegesen lassíthatom a másikat
- egy idő után másik gyorsít, hátha ideiglenes a probléma

Redirect

- Router tudja, hogy van egy nála jobb
- a host változtat a routing tábláján
- VESZÉLYES csak default gatewaynek higgyük el

Time exceed

- lejárt TTL, miatt eldobtam a csomagot

Echo request/reply

- másik gép elérhetőségének ellenőrzése
- ping
- küldött adatot visszaküldi
- oda vissza út ideje, adatveszteség

IP Record Route

- Minden gateway beírja az IP-jét a fejlécbe
- tudjuk merre megy a csomag

Trace Route

- TTL kicsi, növeljük
- újra újra küldjük
- várjuk a választ
- hálózat térképezés / diagnosztika
- hackerek használják sok helyen blokkolt

10. tétel - UDP

UDP

- transport réteg
- egyszerre több program tudjon kommunikálni (portok)
- IP-t használ
- rendkívűl egyszerű és gyors
- Nincs kapcsolat építés / bontás - csak kibassza az éterbe
- van hogy balami / sérül elveszik, de nagyrészt úgyis jó
- azonosítás: IP-k, Portok 4ese

UDP lite

- 2004 - multimédia a neten
- jobb a hibás csomag mint eldobni
- live streemek
- csak ha a címzés rossz, akkor van baj
- nem érdekel ha egy két dolog sérül

UDP mérete

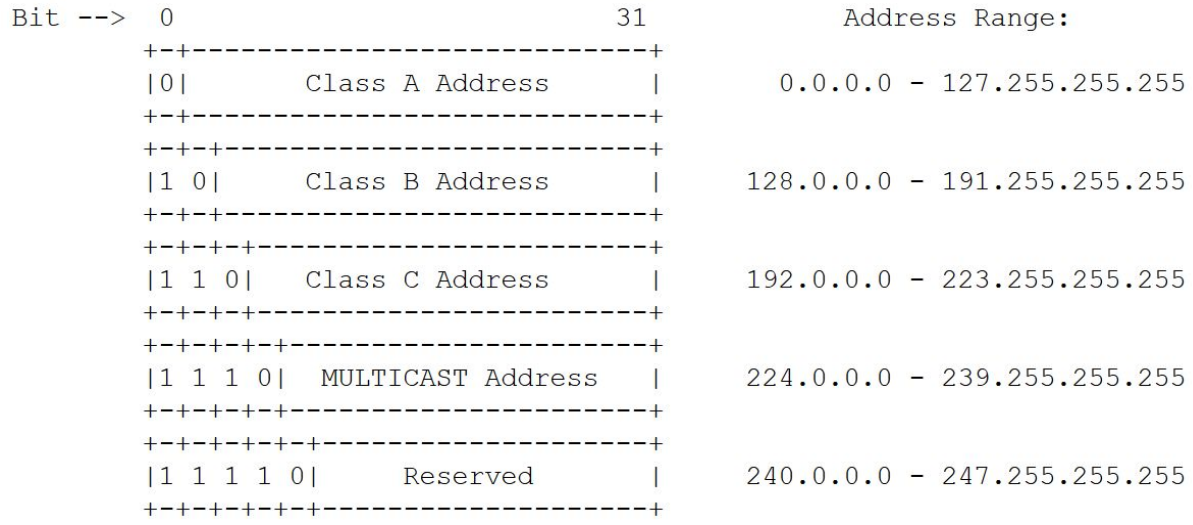
- max(IP méret) - hossz(UDP fej + IP fej) = 65535 -28 = 65507
- általában 8192 byte, de más alkalmazások jobban szűkítik

11. tétel - IP multicast

Multicast és broadcast

- TCP - kapcsolatorientált tehát semmi értelme
- UDP - pl konferencia hívás
- multicast effektív, csökkenti a felslegesen terhelt gépek számát
- nagy fájloknál különsen spórol

Ethernet és IP multicast



- D osztály - multicast címek
- pl. 224. 0.0.1: all host
- 224.0.0.2: all router

IGMP

- szervezés a feladata
- multicast nehezen implementálható, könnyen hálózat foglaláshoz vezethet
- segít megoldani a problémát, csökkenti a terhelést

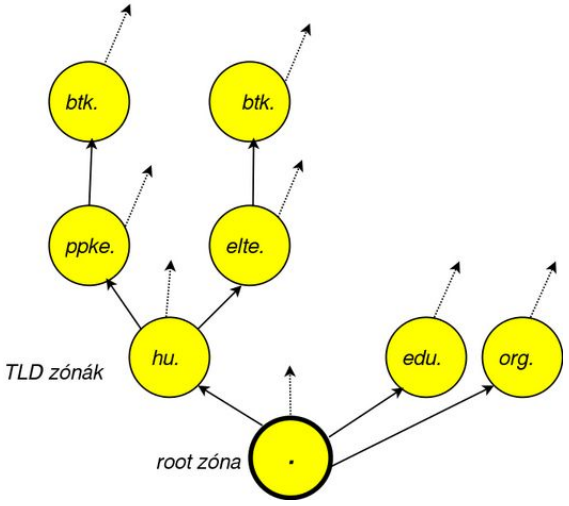
12. tétel - DNS

Bevezetés

- számítógépek címzése
- számunkra fura
- emberibbé tétel
- IP - név megfeleltetés

DNS hierarchia

- Host table
    - eredetileg ez voltt (ma is létezik)
    - gépeden soronként tárolja a név - IP megfeleltetéseket
- hiearchikus felépülés
    - világ egyik legnagyobb hálózata
    - gráf
    - TLD: Top Level Domain
    - ha látok egy címet tudom hol van / hova tartozik



Domain

- .-okkal elválasztottt karaktersorozatok
- alfanumerikus
- ltéezik ékezetes de mi a fasznak
- átkódolás
- FQDN:
    - Fully Qualified Domain Name
    - klasszikus értelemben vett weblap cím, teljes elérési útvonal
- inverz leképezés : mivel pont fordítva mondom mi hol van, ezért fordítva is kérem le

DNS szerver

- UDP 53
- közvetítő szerver
- IP - név
- adatbázist épít (caching only name server)
- többieknek adatot ad (autoratív nameserver)
- kliens: resolver (egy fv ami kikeresi az IP-t)

Cache/TTL

- DNS szerver megjegyzi hogy mi hol van ne kelljen újra lekérnie
- csak egy ideig tárol (TTL), hogy ne poshadjon meg a tudása

DNS cache poisoning

- kamu adatot vigyünk a DNS cachéba
- Man in The Middle

Primary, secondary szerverek

- egyy zónát több szerver is kiszolgálhat
- Van master és slave
- primary adatbázisből vesz infőt
- a slave-k a master alapján frissülnek

DNS rekordok

- zónafájl
    - egy zóna adatait tartalmazza
    - vannak benne rekordok
- SOA rekord
    - infók a DNS zónáról
    - infók a secondaryknek a firssitési ütemrl
- NS rekord
    - adott domain autoratív name serverei
- MX rekord
    - levél célállomás
- A rekord
    - host -IP összekapcsolás

13. tétel - TFTP

Bevezetés

- Trivial File Transfer Protocol
- egyszerű fájlátviteli protokoll
- egyszerűen implementálható
- boot szervernél használják
- oprendszer betöltéséhet
- UDP 69
- STOP AND WAIT
    - minden blokkra nyugtát vár
    - addig nem cisnál mást csak vár, majd újraküld

Bűvész inas szindróma

- ha a k. csomag késik -> nincs válasz
- újra küldöm
- de közben megérkezik a válasz
- ezért elküldöm a k+1-et
- innentől mindent kétszer küldök

Biztonság

- nincs hitelesítés, titkosítás, ezért nem érdemes használni
- manapság elavultnak számít, legtöbször nem támogatják
- egyéb krolátozások alkalmazhatók
    - hozzáférés korlátozása
    - csa bizonyos IP-k férhessenek hozzá

Felhasználás

- Routerek, switchek oprendszer frissítése, telepítése

14. tétel - TCP

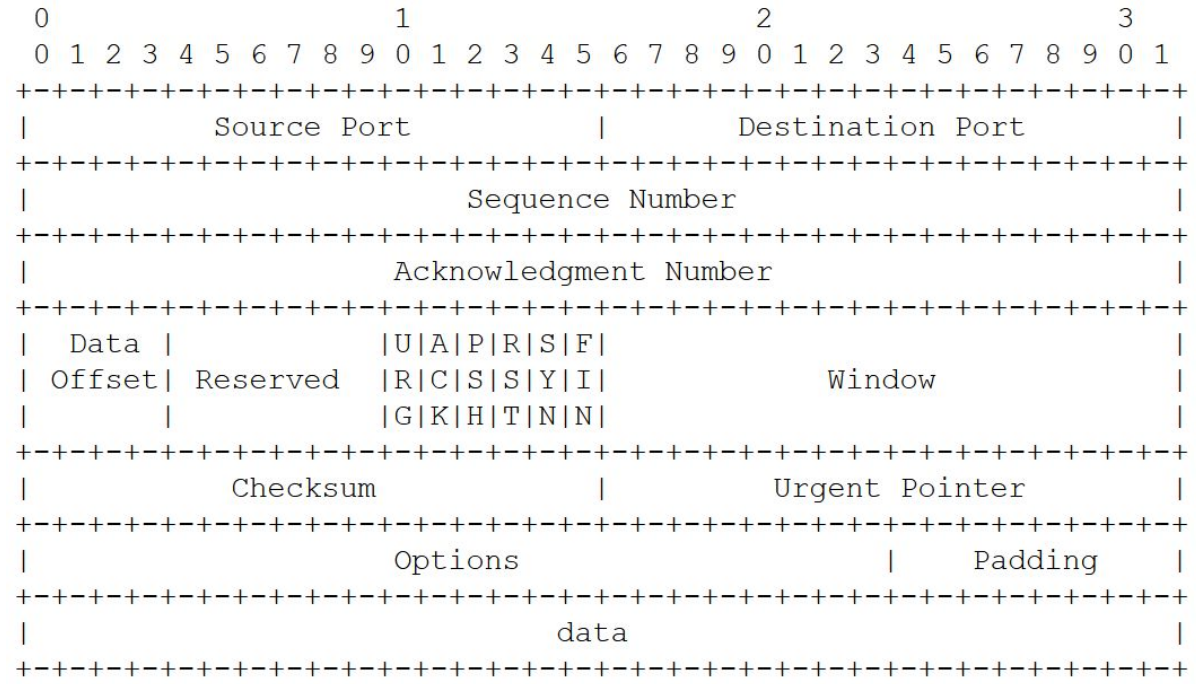
Bevezetés

- Transmission Control Protocol
- Kpacsolat orientált ~ telefon
- Az IP feletti megbízhatóság
- alternatíva : UDP
- alkalmazások egymástól függetlenül : Portok

Szolgáltatások

- csomagot darabol
- nyugtákat vár
- checksumokat alkalmaz ellenőrzésre
- IP csomagok hibáit (sorrend, hiba, duplikáció) szűri
- sebesség szabályozás
- fullduplex

Header

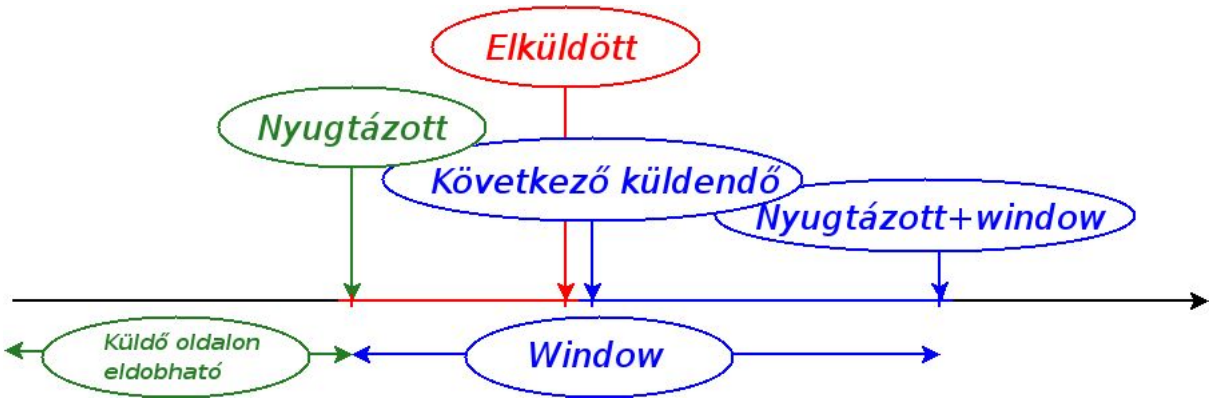


- Source és Destination port értelemszerűen
- Sequence Number
    - az első adatbyte sorszáma
- Ack Num

    - ha ACK = 1, akkor a byte sorszáma, ameddig nyugtázunk
- Data Offset
    - fejrész hossza
- Reserved
    - csupa 0 későbbi használatra
- Flagek: sokféle van RST, URG, ACK stb.
- URG pointer: ha van URG, akkor a sürgés adatra mutat
- padding: kipótolja az options méretét

15. tétel - TCP flow control

Sliding window



- adatáramlás szabályozása
- window ~ buffer mérete
- a csomagküldés telejsen hibamentes és tökéletes: állandó window
- nyílik: jobbra megy
- zárás: 0 lesz a mérete
- Shrink: jobb széle balra megy

Kapcsolatépítés

- TCPkapcsolat építés 3 lépése:
    - kezdeményező SYN-t küld
    - Jön egy ACK és egy SYN
    - Visszamegy az ACK
- Kezdeményező: aktív open
- elszenvedett: passzív open

Kapcsolat zárás

- kezdeményező küld egy FIN-t B-nek
- kap egy ACK-ot
- ha már B sem akar küldeni visszaküld egy FIN-t
- és a kezdeményező küld egy ACK-ot
- ezután megszánik a kapcsolat.
- Kezdeményező: aktív close
- Elszenvedett: passzív close

TCP syn flood támadás

- egy gép folyton SYN flages csomagokat küld random portokról/IP-kről
- a támadó gép küldené az ACKokat, de nem bír velük -> DOS

TIME_WAIT

- Az aktív closet végrehajtó utolsó állapotánál
- már megkapta a FIN-t, és elküldte az ACK-ot
- de mi van ha az ACK elveszik, ezért vár 2MSL ideig utána zár
- MSL: ennyi ideig lehet egy szegmens a hálózaon

TCP reset

- A és B kapcsolatban van
- E bejuttatt egy Resetet és így felbontja a kapcsolatot
- IP-ket portokat, sequence numbert tudni kell hozzá

TCP retransmission timer (RTO)

- Ha RTO ideig nem jön ACK egy csomagra, újra kell küldeni, mert elveszett
- egy időérték - Round Time Tripből (idő amíg odaér + ACK ideje) számoljuk

Delayed ack

- Nem azonnal küldünk ACK-ot
- várunk még, hátha jön még adat
- vagy hátha mi küldünk, amivel tudunk egyben nyugtázni is (piggy back)

Slow start

- nem csak a fogadó, hanem a küldő is flow controlt alkalmaz
- kiegyensúlyozza a hálózat sebességét
- lassan növeljük a csomagokat amiket küdlünk, a hálózat max kapacitásáig
- congestion window - reciver window - egyensúly

Passzív ftp

- Az adatkapcsolatot nem a szerver nyitja hanem a kliens
- PASV paranccsal
- kliens ephemeral portja és a kapott port között nyílik kapcsoal

16. tétel - Distance Vector Routing Protocol, RIP

Bevezetés

- Routing
    - útvonalválasztás
    - IP szinten
    - minimális utat keresem
    - multi path routing
        - több úttal is foglalkozunk
- Routing tábla
    - minden hálózati eszközbe van
    - mi az útvonal egy célponthoz

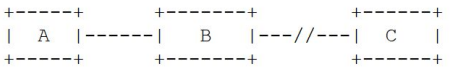
Distance Vector Routing Protocol

- Minden szomszédnak elküldjük a routing táblát (hirdetés)
- Ezt összevetve a sajátunkkal optimalizálunk
- egy idő után konvergencia

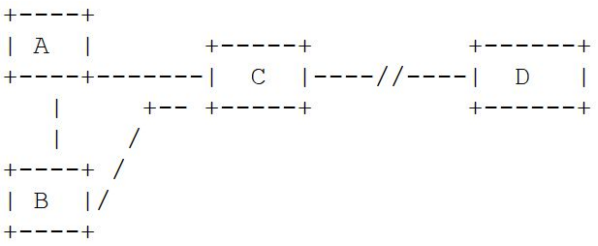
Link State Protocol

- nem a routing táblát, hanem a kapcsolataikat küldik (nem csak szomsédoknak)
- a kapott infók alapján mindenki felépíti a saját routing tábláját

Split horizon



- B és C között szakadás
- de A attól még azt hiszi, hogy tőle elérhető 2 távra a C
- Megoldás:
    - Split Horizon
        - ha A csak B-től tud C-ről akkor B irányába nem sugároz infót
    - Poisoned Reverse
        - végtelen költséggel hirdeti vissza
- Problémás lehet még
    - ekkor A B hirdetnek egymásnak hiába baszódott a C-D



RIP

- Routing Information Protocol
- Distance vektor alap
- Split Horizon, Poisoned Reverse technikákat ismeri
- 15 ös hoppmax (azt már infinitynek veszi ami nagyobb)
- RIP1 - IP broadcast
- RIP2 - IP multicast
- UDP 520

Miért RIP?

- PRO
    - egyszerűen konfigurálható és gyors
- Konrtra
    - nagy hálózatban off a 15ös szabály miatt

17. tétel - IGP/EGP és BGP

AS

- Autonóm Rendszer
- Routing szempontjából egy önálló entitás
- azonosító: AS number, RIPE osztja
- S fajták:
    - Stub: egy másik AS fele van kapcsolata
    - Multihomed: több AS fele van kapcsolata
    - Transit: nem csak a saját forgalmát bonyolítja

IGP/EGP

- Interior Gateway Protocol / Exterior Gateway Protocol
- AS-en belül: IGP
- AS-en kívűl: EGP

BGP

- Border Gateway Protocol
- az interneten ezen alapul a routing
- TCP alapú
- AS-en belül is használható: iBGP
- Distance vector: a célhoz vezető AS-eket tartja számon: AS Path
- nem feltétlenül a legolcsóbb a legjobb
- Route-olás:
    - rövidebb AS path
    - AS-en belüli út preferált
    - A rövide ilyen út preferált
    - kisebb IP
- BGP dampening
    - gyakran változó hirdetéseket nem veszik figyelembe

BGP üzenetek

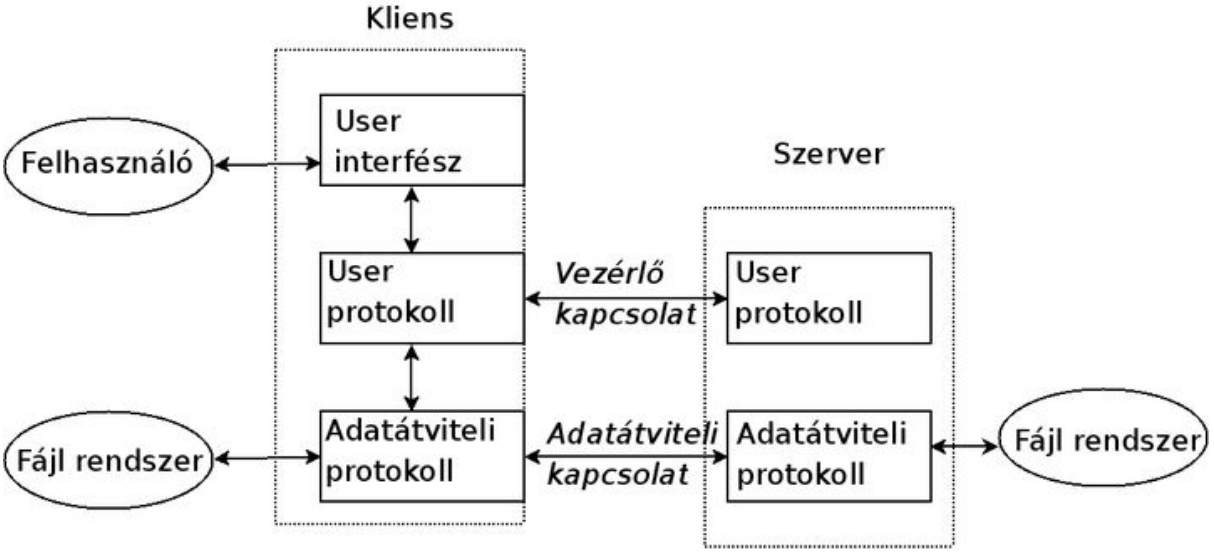
- open
    - kivagyok, holdtime(ennyi időnként szólj)
- update
    - route küldése
- keepalive
    - nincs új infó de ne dobj el

- notification
    - hibaüzenet
- refreshroute
    - küldd el a routing táblád

18. tétel - FTP

Bevezetés

- File Transfer protocol
- tartalomátvitel
- ügyfél és kiszolgáló alapú



- nem kifejezetten biztonságos -> SFTP

Irányítás

- a szerverrel egy soros parancsokkal lehet kommunikálni
- mindig be kell lépni a kapcsolathoz, ehhez azonosítás szükséges
    - USER
    - PASS
- Egyéb fontos parancsok
    - LIST - fájl listázás
    - ABOR - átvitel megszüntetése
    - RETR - fájl átmásolása
    - STOR - fájl feltöltése
    - QUIT - kapcsolat bontása

Szerver válasz

- xyz valami szöveg
- 1-5 ös “skála” sikertelenségi mérték
- szöveg embernek, szám programnak szól
- 20-as portot használja

Passzív és aktív FTP

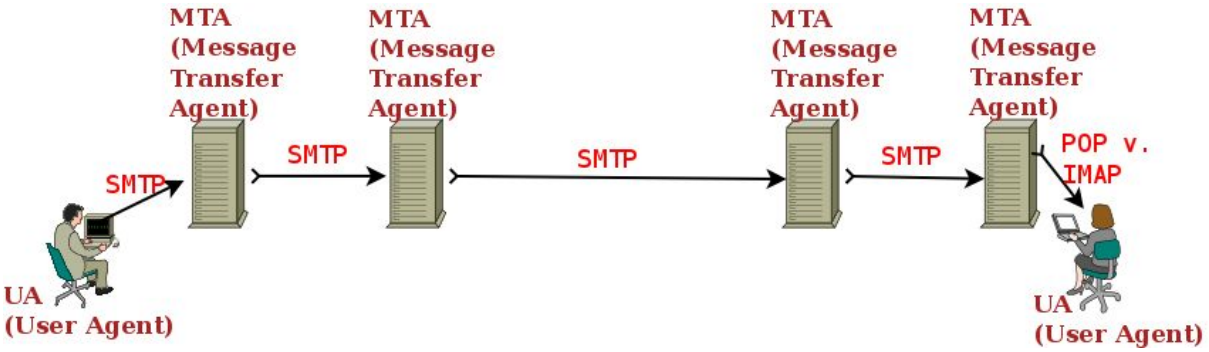
- Aktív FTP
    - biztonsági kockázat
    - közvetlen kommunikáció
    - kétirányú
    - PORT parancs
- Passzív FTP
    - külön csatornán a kommunikáció és a fájlátvitel
    - PASV

Anonymous FTP

- USER anonymous
- PASS email
- nyilvánosan olvasható fileok listája

19. tétel - SMTP

Bevezetés



- SMTP = Simple Mail Transfer Protocol
- egyszerű szövegalapú levelezéshez használt protocol
- UA: felhasználó által kezelt mail program
- MTA: leveleket továbbító szerverprogram
- több MTA-n megy keresztül
- mindenütt SMTP, végén POP vagy IMAP
- Post Office Protocol, Internet Mail Acces Protocol

Vezérlés

- hasonló utasítások mint az FTP-n
- válaszok: xyz valami
    - same

SMTP folyamat

- 25-ös port
- kliens egy ephemral portot használ

SMTP parancsok

- HELO fqdn.domain.nev
    - kliens köszönése
    - kapcsolat kezdeményezése
- MAIL FROM: <kaki@puki>
    - megadja a feladót
    - lehet öresen is hagyni
    - puki FQDN
- RCPT TO: <kaki@puki>

    - másolat
    - max 100 cím
- DATA
    - üzenet szövegének kezedete
    - üzenet vége : egy üres sor egy .-tal
    - 7 bites ASCII
- QUIT
    - beszélgetés vége
- RSET
    - minden információt amit kitöltöttünk töröl
- NOOP
    - ok válasz
    - csak hogy lássuk legit e a kapcsolat
- VRFY
    - ellenőrzi hogy egy adott cím létezik e

ESMTP

- SMTP bővebb változata
- tőbb parancs
- EHLO nyitás HELO helyett
- max email méret, plussz biztonság, 8 bites kódolás (ékezetes karakterek)

RBL

- Real time Black hole List - olyan feladók gyűjteménye akik valszeg spammelnek

20. tétel - levél

Valami ronda RFC

- levélformátum
- sorok max 998 karakter
- levél: fejrész & body
- fejrész az első üres sorig tart
- DATE
    - feladás dátuma
- TO
    - cmzett
- CC
    - másolat
- BCC
    - titkos másolat
- FROM
    - feladó
- SENDER
    - tényleges feladó
- REPLY-To
    - erre a címre válaszolj
- MASSAGE-ID
    - levélazonosító
- SUBJECT
    - tárgy
- References
    - ID-k amikre hivatkozik a levél (levlistánál hasznos)
- Recevied
    - MTA-k írhatnak bele - kiderül merre jár a levél

Kitől fogadok?

- régen nem korlátozták, manapság SPAM-ek miatt nem midnegy

MIME

- Multipurpuse Internet Mail Extensions
- ha nem csk a 7 bites kis karaktereinket akarjuk
- hanem úgy bármi mást
- fa felépítés
- meghatározza, hogyan kell majd kódolni az üzenetet, stb.

21. tétel - WWW

Bevezetés

- world wide web
- világháló
- az interneten működő hiperlinkeken keresztül összekötött dokumentumok
- hálószerű
- CERN - egy random tag találja ki, eredetileg info megosztásra
- mivel nem volt szabadalmaztatva nagyon gyorsan fejlődött
- később a CERN mindenki számára elérhetővé tettek

Architektúra

- kliens - browser
- szerver - httpd (deamon program)
- HTML - a web nyelve
- URL - webcímek
- HTTP
- statikus dinamikus/tartalmak

URI

- Unified Resource Identifier
- egy erőforrás azonosítása egy karaktersorozat alapján ~ UNI
- reserved karakterek
    - ezek le vannak foglalva, különféle azonosítási céljuk van
    - pl / ? !
- unreserved karakterek
    - ezeket szabadon használhatjuk
    - pl. - _ ~ .
- a nem megengedett karaktereket kódoljuk
- Kétfajta: url, urn

URN

- URN: Unified Resource Name
- absztrakt, konkrét dolog azonosítására szolgál

URL

- valamilyen erőforrás helyét azonosítja

- kis és nagy betű host névben nem számít
- relatív url pl images/kep.jpg

22. tétel - HTTP

Bevezetés

- tcp feletti protokoll
    - adatveszteni nem akarunk
- HyperText Transfer Protocol
- alapértelmezésben egyetlens kérés / válasz TCP-b
- stateless
    - nem tart fenn állapotot a szerver az egyes kliensekről
    - nem erre készült, de ez terjedt el
    - egyszerű, de nem tud sokat -> cookies
    - sütik nélkül minden weboldal lekérés egymástól független lenne

Cookie

- a cookiek segítségével a szerver bekukucskál a gépünkbe
- információcsomag
- legtöbbször azonosítás, bevásárló kosár megtartására, nyomonkövetésre használják
- gyakran reklámokkal jön -> célzott reklámokat akarnak
- ha rossz kezekbe kerül szopás : XSS

HTTP kérések és válaszok

- kérés és válasz szerkezete megegyezik
- a kezdő sor megmondja milyen kréés vagy milyen válasz

```
GET http://www.elte.hu/index.html HTTP/1.1
```

```
HTTP/1.1 404 Page Not Found
```

- a leggyakorib http kérés : GET
- HEAD
    - Ugyanazt adja vissza, mint a GET, csak magát az üzenettestet hagyja ki a válaszból.
- GET
    - A megadott erőforrás letöltését kezdeményezi. Ez messze a leggyakrabban használt metódus.
- POST
    - Feldolgozandó adatot küld fel a szerverre. Például HTML űrlap tartalmát. Az adatot az üzenettest tartalmazza.
- PUT
    - Feltölti a megadott erőforrást.

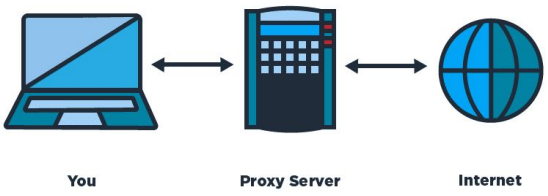
- DELETE
    - Törli a megadott erőforrást.
- TRACE
    - Visszaküldi a kapott kérést. Ez akkor hasznos, ha a kliens oldal arra kíváncsi, hogy a köztes gépek változtatnak-e illetve mit változtatnak a kérésen.
- OPTIONS
    - Visszaadja a szerver által támogatott HTTP metódusok listáját.
- CONNECT
    - Átalakítja a kérést transzparens TCP/IP tunnellé. Ezt a metódust jellemzően SSL kommunikáció megvalósításához használják.

HTTP/1.1 sajátságok

- Virtuális szerverek
    - egy IP-n több szervert lehet hostolni
    - Ezért létezik a Host fejléc elem
- Chunked Transfer Encoding
    - a fejrész elemek között jelzi a szerver a kontent hoszát
    - darabokban küldi az információt
- Prezisztens kapcsolatok
    - A http/1.1-ben egy TCP kapcsolaton több kérés/válasz pár is végbemehet

Proxy

- szerverként és kliensként is működik
- köztes elem
- pl gyorsabb weboldal betöltés - cachel információt
- anonimitás



SSL/TLS

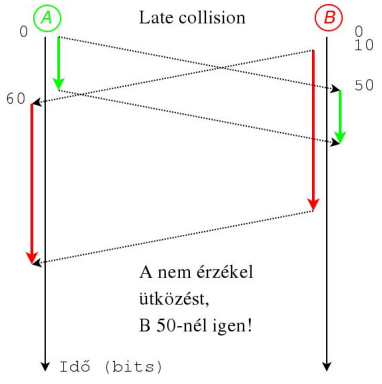
- Netscape találmány
- Secure Socket LAyer
- nyilvános kulcsú titkosítás
- védelem a böngészés felett

- lehallgatás megakadályozása

TLS handshake protokoll

- Kriptográfiai algoritmusokban és paraméterek egyeztetése
- Véletlen számok szerint 'master secret-et' generálnak
- A TLS record réteg számára paramétereket adnak át
    - Rendszeresen cserélik a titkosításra használt kulcsot
- A kliens 'Client hello' üzenettel köszön
    - Közli, hogy milyen protokollt fog használni (pl. SSL vagy TLS)
    - Közli, hogy milyen algoritmusokat támogat
- A szerver 'Server hello' üzenettel válaszol
    - A ,,legnagyobb közös osztó'' lesz a használt protokoll…
- sok verzió volt, botrány man inthe middle

Villámkérdések

- Ephemeral port
  - A szerver vagy kliens által véletlenszerűan választott port. Általában 1024<, de különböző tartományok léteznek.
- CIDR
  - Classless Interface Domain
  - megszünteti az IP cím osztályokat, ezzel vezetve a lassabb IP cím elavuláshoz
- RFC
  - Request For Comment
  - Internetes szabványok gyűjteménye
- IANA
  - Internet Assigned Numbers Authory
  - különböző internetes paraméterek (pl IP) kiosztásáért felelős
- IAB
  - Internet Architecture Board
  - Internet fejlődését felügyeli szervezet (szabványok) - RFC
- RIPE NCC
  - Francia neve van lol
  - Az európai IP címek osztásáért felelős
- Unicast/multicast/broadcast/anycast
  - egycímzett/több címzett/mindenki címzett/bárki lehet címzett (mentők)
- Simplex/duplex/half duplex
  - egyirányú/kétirányú/felváltott kétirányú
- CSMA/CD
  - CS = Carrier Sense
    - figyelem hogy ne beszéljek ha más ad
  - MA = Multiple Acces
    - mindenkit figyelek, de csak a saját csomagjaim érdekelnek
  - CD = Collision Detection
    - ütközésnél 32 zajbit, backoff delay, exponential backoff
- Late collision
  - nem lehet észrevenni az ütközést az egyik oldalon
  - korlátozni kell a kábelhosszt, meg a repeaterek számát
  - 

- Exponential backoff
- Little endian/Big endian
    - legkisebb helyiértékű az első nagy az utolsó
    - legnagyobb az első legkisebb az utolsó
- Spanning tree
    - Kör van a topológiában: csomagok körbe-körbe járnak egyéb gondok
    - Spanning Tree - feszítő fát keresek, aszerint járok el, többet is használhatok
- VLAN
    - Virtual LAN
    - egy ethernetet logikailag több részre osztunk
    - előnye: biztonság
- Loopback interfész
    - ha egy eszközön van a kliens és a szerver ~ localhost
    - 127.0.0.1
- MTU
    - Maximum Transition Unit
    - max amit áttudunk küldeni két hálózat között
    - általában 1500 byte
- TOS
    - Type Of Service
    - Milyen szolgáltatásokat tud a csomagunk
    - a hálózaton belüli továbbítást befolyásoló mező
- TTL – IP csomagnál, DNS rekordnál
    - Time To Live
    - IP-nél minden csomag csökkenti eggyel, ha 0 eldobjuk
    - nem kering végtelen ideig
- Netmask
    - Egy IP-hez képest egy tartomány.
    - Egy IP rajta van e a hálózaton?
- NAT
    - Network Adress Translation
    - Hálózaton belüli címet átalakítjuk, hogy kívül is alkalmazható legyen
    - Belső gépek közti kommunikáció miatt
- ARP cache poisoning
    - ARP cachébe hamis MAC cím kerül -> Man in The Middle
- Path MTU discovery
    - ICMP vezérlő üzenet
    - mi a legkisebb MTU a cél felé
- ICMP redirect
    - ha a router tudja, hogy van nála egy jobb szól
    - csak Gatewaytől!
- AS
    - Autonóm Rendszer
    - Routing szempontjából egy önálló entitás
    - azonosító: AS number, RIPE osztja
    - S fajták:

        - Stub: egy másik AS fele van kapcsolata
        - Multihomed: több AS fele van kapcsolata
        - Transit: nem csak a saját forgalmát bonyolítja
- UDP lite
    - Multimédiás kommunikáció esetén nem érdekel ha elvesztünk pár csomagot
    - a sérült is jobb mint a semmi
- SOA, NS, A, PTR, MX rekord
    - DNS Resource Record tagjai
    - Start of Authority
        - DNS zóna adatok
        - sorozatszám, időzítések
    - Nameserver
        - meghatározza, hogy melyik nameserverhez kell intézni adott lekéréseket
    - Adress
        - az állomás névhez tartozó IP
    - Pointer
        - HIÁNY
    - Mail Exchanger
        - levél célállomás
- FQDN
    - Fully Qualified Domain Name
    - klasszikus értelemben vett weblap cím, teljes elérési útvonal
- Primary (master), secondary (slave) DNS szerver
    - Egy zónát több szerver is kiszolgálhat
    - Van master és slave
    - primary adatbázisből vesz infőt
    - a slave-k a master alapján frissülnek
- Caching only DNS szerver, autoritatív név szerver
    - aut névszeró:egy zóna felelőse, egy ágé
    - ha nem autoritás, akkor caching only - ekkor nincs különösebb dolga, csak folyton lekéréseket hajt végre, amiből saját cachéjét firssíti
- DNS cache poisining
    - DNS cachét manipulálják
    - pornhub helyet ducktv oldala jön be és ellopják az adataid
- TCP syn flood támadás
    - egy gép folyton SYN flages csomagokat küld random portokról/IP-kről
    - a támadó gép küldené az ACKokat, de nem bír velük -> DOS
- TCP aktív/passzív open
    - TCPkapcsolat építés 3 lépése:
        - kezdeményező SYN-t küld
        - Jön egy ACK és egy SYN
        - Visszamegy az ACK
    - Kezdeményező: aktív open
    - elszenvedett: passzív open
- TCP aktív/passzív close

    - kezdeményező küld egy FIN-t B-nek
    - kap egy ACK-ot
    - ha már B sem akar küldeni visszaküld egy FIN-t
    - és a kezdeményező küld egy ACK-ot
    - ezután megszánik a kapcsolat.
    - Kezdeményező: aktív close
    - Elszenvedett: passzív close
- TIME_WAIT (2MSL wait) állapot
    - Az aktív closet végrehajtó utolsó állapotánál
    - már megkapta a FIN-t, és elküldte az ACK-ot
    - de mi van ha az ACK elveszik, ezért vár 2MSL ideig utána zár
    - MSL: ennyi ideig lehet egy szegmens a hálózaon
- TCP reset támadás
    - A és B kapcsolatban van
    - E bejuttatt egy Resetet és így felbontja a kapcsolatot
    - IP-ket portokat, sequence numbert tudni kell hozzá
- TCP retransmission timer (RTO)
    - Ha RTO ideig nem jön ACK egy csomagra, újra kell küldeni, mert elveszett
    - egy időérték - Round Time Tripből (idő amíg odaér + ACK ideje) számoljuk
- Delayed ack
    - Nem azonnal küldünk ACK-ot
    - várunk még, hátha jön még adat
    - vagy hátha mi küldünk, amivel tudunk egyben nyugtázni is (piggy back)
- Slow start
    - nem csak a fogadó, hanem a küldő is flow controlt alkalmaz
    - kiegyensúlyozza a hálózat sebességét
    - lassan növeljük a csomagokat amiket küdlünk, a hálózat max kapacitásáig
    - congestion window - reciver window - egyensúly
- Persist timer
    - HIÁNY
- Passzív ftp
    - Az adatkapcsolatot nem a szerver nyitja hanem a kliens
    - PASV paranccsal
    - szerver: PORT NR
    - kliens ephemeral portja és a kapott port között nyílik kapcsoalt
- UA, MTA
    - User Agent: a felhasználó mail programja
    - Mail Transfer Agent: a leveleket továbbító mail program
- DHCP lease
    - DHCP dinaimikus IP osztása, időre adom
    - IP + kliens + intervallum = DHCP lease
- PXE (Preboot Execution Environment)
    - turbós DHCP Inteleken
- Proxy ARP?
    - ha a másik hálózatban van akit keresünk
    - broadcastre nincs válasz

    - amúgy sincs értelme ha másik hálózatban van
    - a router azt hazudja magáról hogy ő az, és rajta keresztül küldünk
    - Default Gateway: ha eleve tudom, hogy más hálózatban van, neki küldöm és rábízom magam.
- Hogyan működik az IP csomag fragmentálás?
    - ha az IP csomag nagyobb mint az MTU, akkor darabolni kell
    - csak akkor tudjuk ha a fragment bit áll
    - a darabokat a fragment offszet érték szerint sorba rendezhetjük
    - ID azonos
    - ha nem áll a fragment bit ICMP error
- Miért felejtenek és mit felejtenek switch-ek?
    - MAC címeket időnként elfelejtik
    - dinamikus a hálózat, lehet valaki nem tér vissza
    - pl Meki
- Mi a különbség hub és switch közt?
    - Hibás adatot a switch nem küld tovább
    - a HUB kb egy repater nem tanul címeket
    - a switch fejlettebb
- Miért felejtik el az eszközök az ARP bejegyzéseiket?
    - dinamikusság miatt, hogy észrevegyék a hálózaton a változásokat
- Hogyan működik a traceroute?
    - egyre növekvő TTL-el küld csomagot
    - majd vár választ
    - Time exceed, vagy megérkezést
    - rendszer diagnosztika, hálózat térképezés
- Hogyan működik a ping?
    - küldünk egy echo üzenetet, majd a céltól változás nélkül visszakapja
    - haszna, hogy le tudjuk ellenőrizni hogy valami elérhető e a hálózaton
    - útidőt számol és adatveszteséget
- Miért tartalmazzák az ICMP hibaüzenetek a hibát kiváltó csomag egy részét is?
    - Milyen IP címről jött a hiba, melyik porton -> magyarul hol a hiba?
- Mire vezetne, ha icmp hiba üzenetre icmp hiba üzenet lehetne a reakció?
    - agyon terhelné a hálózatot, végtelenségig küldözgethetnék egymásnak
- Mit értünk routing protokolloknál konvergencián?
    - routing táblák küldése, utak optimalizálása
    - egy időután beáll egy egyensúlyi állapot
- Mi a kockázata annak, ha bárhonnan elfogadunk RIP üzeneteket?
    - HIÁNY
- Miért fontos, hogy a root név szerverek mindig elérhetőek legyenek?
    - ha nem elérhető a cachéból kiesett címeket nem fogjuk tudni elérni
- Mire szolgál és hogyan működik az inetd daemon?
    - unixokon
    - egyszerű szolgáltatások indítása (pl. echo)
    - standard input és output a TCP lesz
- Hogyan zajlik egy TCP kapcsolat felépítés?
    - TCPkapcsolat építés 3 lépése:

        - kezdeményező SYN-t küld
        - Jön egy ACK és egy SYN
        - Visszamegy az ACK
- Hogyan zajlik egy TCP kapcsolat bontás?
    - kezdeményező küld egy FIN-t B-nek
    - kap egy ACK-ot
    - ha már B sem akar küldeni visszaküld egy FIN-t
    - és a kezdeményező küld egy ACK-ot
    - ezután megszánik a kapcsolat.
- Mitől függ a TCP retransmission timeout?
    - RTT-től (csomag elküldésétől az ACK megérkezéséig tartó idő)
    - bRTT = a * bRTT + (1-a) * mRTT
    - RTO = b * bRTT
    - bRTT: becsült RTt
    - mRTT: mért RTT
- FTP-nél/SMTP-nél hogyan csoportosítjuk a szerver hibakódokat?
    - xyz valami szöveg
    - HIÁNY
- Miért okoz gondot a tűzfalakon az aktív ftp?
    - ne engedjünk be olyan dolgokat, amiket más kezdeményez / egyből megvágják
- Hogyan zajlik egy SMTP kapcsolatfelépítés?
    - HELO fasz.szerver.fing
    - MAIL FROM: <jopuki@maki.suni>
    - RCPT TO: <todor@kolonia.tacsi>
    - DATA
    - szoveg
    - .
    - QUIT
- Mi az a looking-glass?
    - Az interneten elszórva http felületen lekérdezhető routerek
    - Diagnosztikai eszköz
- Mire való a whois szolgáltatás?
    - megadja egy domainhez tartozó IP-t, egyéb hasznos infók amik egy embert érdekelhetnek
- Egy TCP kapcsolatnál a receiver window értéke 1000, a congestion window (cwnd) 1500. Mekkora csomagot küldhetünk?
    - 1000-es különben nem tudná fogadni gondolom én
- Milyen üzenetekben láthatunk HTTP cookie-kat? Mire szolgálnak?
    - pl sok banner tartalmazza
    - a cookiek segítségével a szerver bekukucskál a gépünkbe
    - információcsomag
    - legtöbbször azonosítás, bevásárló kosár megtartására, nyomonkövetésre használják
    - gyakran reklámokkal jön -> célzott reklámokat akarnak
    - ha rossz kezekbe kerül szopás : XSS

- Mire való az SSL/TLS?
    - Netscape találmány
    - Secure Socket LAyer
    - nyilvános kulcsú titkosítás
    - védelem a böngészés felett
    - lehallgatás megakadályozása