Nemes Axel Roland 2021. 11. 21.

Kommunikációs Hálózatok jegyzet

1-2. Előadás

Vonalkapcsolás:

- összeköttetés kiépítése és fenntartása A és B között
- állandó átviteli ráta

Csomagkapcsolás:

- adatok továbbítása kis „adagokban”
- egymástól függetlenül
- csomópontból csomópontba („store-and-forward” elv)

Hálózati csomópontok: IMP-ek

Csomagtovábbítás:

- Kezdetben: NCP (Network Control Protocol)
- Később: IP és TCp együttes használata: TCP/IP

IP:

- a hálózat csomópontjainak viselkedését szabályozza
- **best effort** szolgáltatást nyújt a csomagok továbbítását illetően

TCP

- a hálózat végpontjai közötti kommunikáció
- megbízható adatátvitelt nyújt
- (később UDP)

Mi az Internet?

Számítógépek milliárdjai összekapcsolva

- **Végpontok**: Hostok – Hálózati alkalmazásokat futtatnak
- **Kommunikációs linkek** (üvegszál, rézvezeték, stb…) [Adatátviteli ráta = sávszélesség]
- **Routerek**: csomagot továbbítanak

Közelebbről nézve:

Végpontok (network edge)

- alkalmazások és felhasználói terminálok

Hozzáférési hálózat (access network)

- komm. csatornák

Gerinchálózat (network core)

- routerek
- hálózatok hálózata

Network Edge

- **Végfelhasználók** (alkalmazásokat futtatnak)
- **Kliens/szerver modell** (kliensek szolgáltatást vesznek igénybe a szervertől)
- **peer-peer modell** (minimális számú/nincs dedikált szerver)

Hozzáférési hálózat és fizikai csatornák: Komm hál 2.

Gerinchálózat

Routerek hálózata

Hogyan vigyünk át adatot?

- **Áramkapcsolás**: dedikált csatorna minden kapcsolatra
- **Csomagkapcsolás**: adat darabokra szeletelve

Az Internet: Hálózatok hálózata

Középpontban: **Tier-1 ISP**-k

- nemzeti/nemzetközi lefedés

Tier-2 ISP-k: (regionális ISP-k):

- 1/több Tier-1-hez és netalán más Tier-2-esekhez csatlakoznak
- Fizet a Tier-1nek az Internethez való csatlakozásért

Tier-3 ISP-k és Helyi ISP-k

Legközelebb a végponti rendszerekhez

Hálózati összeköthetőség

A és B végpont összeköthető legyen

- linkek és hálózati csomópontok sorozatán keresztül
- megfelelő útvonalválasztó és –irányító módszerek és protokollok

“jó” linkjeink legyenek

- kellő sebességűek
- megfelelő fizikai paraméterekkel (hibaarány, késleltetés)

ha valamely linken v. csomópontban forgalomtorlódás:

- a forgalom elterelése
- a forrás szabályozása
- megelőző módszerek

Csomagok eltérülnek, elvesznek, eldobásra kerülnek a csomópontok minden igyekezete („best effort”) ellenére.

Kell tehát kommunikációs eljárás és protokoll amely a végpontok közötti adatcserét támogatja.

3. Előadás

Web és http

Weblap objektumokat tartalmaz PL.: HTML, JPEG, hangfile, stb…

Minden objektumnak saját URL!

- alkalmazni kívánt protokoll
- gépnév/ip cím
- fájl elérési útja
- port

HTTP – HyperText Transfer Protocol

- Kliens-Szerver modell
- Kliens: kérést küld/fogad, megjeleníti a webes objektumokat
- Szerver: kérésekre webes objektumot küld

FTP – File Transfer Protocol

- Fájl átvitel
- Kliens-Szerver modell
- Kliens: le/feltölt, msáol, stb…
- Szerver: távoli host

Elektronikus Levelezés

Három fő alkotóelem:

- Message User Agent
  - Levelek írása, szerkesztése, olvasása
- Levélszerverek
  - postaláda tartalmazza a felhasználó bejövő leveleit
  - Kimenő levelek várólistája a szerveren
- Levelező protokoll (SMTP)
  - Szerverek közötti levelek továbbítása
  - kliens küld
  - Szerver fogad

P2P fájlcsere

Mindenki lehet szerver és kliens is

Centralizált (Napster)

Teljesen elosztott (Gnutella)

1. Generáció: hibrid, központi szerver
2. Generáció: Teljesen elosztott, nincs központi szerver
   a. robosztusabb
   b. hátrány: elárasztás (sok üzenet miatt)
3. Generáció: párhuzamos kommunikáció
   a. Egyszerre több csomópont
   b. seed, tracker, leech csomópontok

Előnyök:

- robosztus
- jól skálázható

Hátrányok:

- nagy sávszélesség igény
- P2P forgalom blokkolása
- rosszindulatú szeletek injektálása

Eddigi alkalmazások igényei a hálózattal szemben:

- alapvetően „csak” megbízható kapcsolat végpontok között
- valamekkora átbocsátóképesség („sávszélesség”)
- ⇨ IP és TCP elég!

További nem túl igénytelen alkalmazások:

Audió-videó streaming

- tárolt
- élő
- interaktivitás (pause, fast forward, stb…)

Élő beszéd (VoIP)

Elő videókonferencia (audió+videó)

Multimédia Streaming

Streaming: a kliens megkezdi a lejátszást, még **mielőtt a teljes tartalom megérkezne**

időbeli követelmény a még továbbítandó adatokra: időben érkezzen meg a kijátszáshoz (playout)

A beszédminőséget befolyásoló tényezők csomagkapcsolt hálózatokban:

- **Késleltetés**
  - max ~150 ms
- **Késleltetés ingadozás**
  - max néhány 10 ms
- **Csomagvesztés**
  - véletlenszerű eloszlás

A megbeszélt multimédia alkalmazások igényei: összefoglalás

A multimédia alkalmazások igényei:

- **érzékenyek a késleltetésre** (delay sensitivity)
  - a késleltetés nagyságára
  - a késleltetés ingadozására
- de **tűrik az adatvesztést** (loss tolerance)

Szemben az első csoport (http, FTP, stb…) „egyszerű” internetes alkalmazásaival (adatátvitellel), amelyek

- **nem tűrik az adatvesztést**
- de **tolerálják a késleltetést**

Tehát a multimédia alkalmazásoknál nem kell és nem is szabad TCP-t használni!

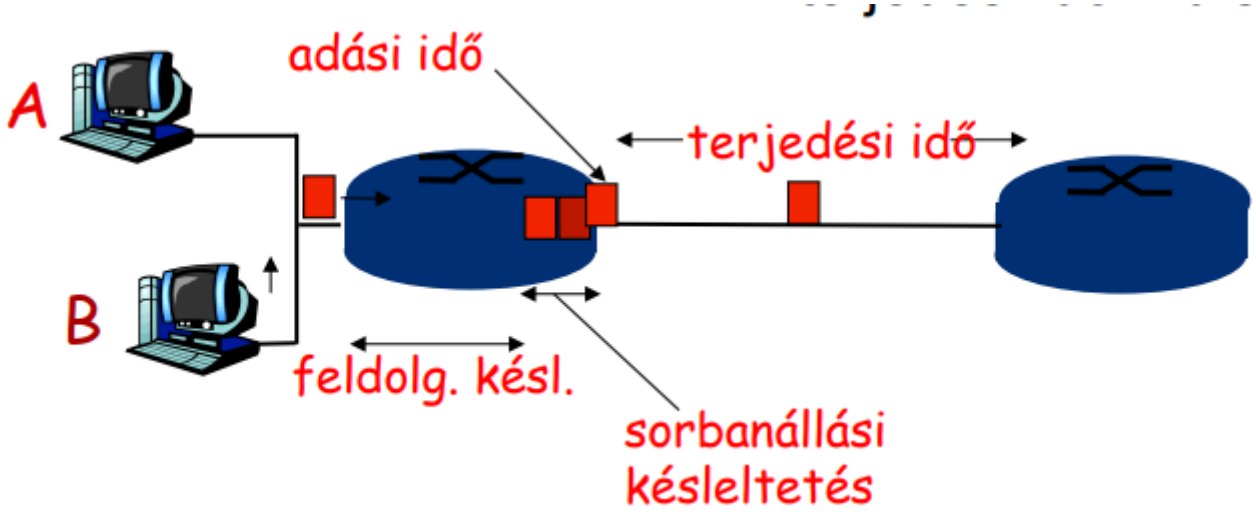
Mik a késleltetés és csomagvesztés okai?

csomagok sorban állnak a csomóponti gépek tárolóiban: késleltetés

csomagok **beérkezési sebessége** > kimenő link **átbocsátóképessége**: csomagvesztés

A csomagok késésének négy fő oka:

1. Feldolgozás a csomópontban
   a. Hibaellenőrzés
   b. kimenő link meghatározása
2. Sorbanállás
   a. várakozás továbbításra az adott kimenő linken
   b. router terheltségétől függ
3. Adási idő
   a. R = link adatátviteli sebessége (bit/s)
   b. L = csomaghossz (bit)
   c. adási idő = L/R
4. Terjedési idő
   a. d = link fizikai hossza
   b. s = terjedési sebesség a közegben
   c. terjedési idő = d/s



1. Csomóponti késleltetés =
   - fedolgozási idő
   - sorbanállási idő
   - adási idő
   - terjedési idő

   Összege

2. Sorbanállási késleltetés

- R = adatátviteli sebesség
- L = csomaghossz
- a = átlagos csomag érkezési ráta
- Forgalom intenzitása: La/R
  - ⇨ La/R ~ 0: nincs sorbanállás
  - ⇨ La/R -> 1: nagy késleltetés
  - ⇨ La/R > 1: több igény mint amit ki lehet szolgálni

Csomagvesztés

A tár/buffer kapacitása véges

Ha tele a buffer és érkezik adat akkor eldobja ➔Csomagvesztés

Csomagvesztés lehet még ha hibás a csomag!

Elveszett csomagokat vagy újra küldik vagy sem

Hogyan egyeztessük az alkalmazások igényeit és a hálózat problémáit?

szolgáltatásminőséget (QoS – quality of service) biztosító módszerek és protokollok

Egy ötlet, foglaljunk erőforrásokat:

- csomóponti képességeket
- link-kapacitásokat

Másik ötlet:

jelöljük meg a csomagokat a különböző igények szerint és a csomópontok ennek megfelelően kezeljék azokat (pl.: **Prioritás**)

4. Előadás

PROTOKOLLARCHITEKTÚRÁK

A kommunikációs hálózatoknál is szükség van **viselkedési szabályok**, azaz **protokollok**

- lerögzítésére
- betartására

Protokoll Def:

> *A protokoll két vagy több kommunikáló egység közötti üzenetcsere formátumait és az üzenetek sorrendjét határozza meg, valamint az üzenetek vételéhez kapcsolódó és egyéb események által kiváltott tevékenységeket.*

Szintaxis: adategységek formai leírása

Szemantika: az adatelemek értelmezése

Viselkedés: mit kell tenni valamely üzenet vételekor, küldésekor

Miért nem elég 1 protokoll?

- Részfeladatokara tördelés
  - ⇨ egyszerűbb a tennivalók elvégzése
  - ⇨ részenként lehet tovább fejleszteni

Részek – Funkcionalitások – Rétegek

Rétegek: minden réteg egy szolgáltatást valósít meg

- a saját **rétegen belüli tevékenységével**
- támaszkodva az **alatta lévő réteg** által nyújtott **szolgáltatásra**

A rétegezett protokollarchitektúrák előnyei és hátrányai

Előnyök:

- Összetett feladat **kezelhető részekre bontható**
- **könnyen módosítható** (részek függetlenek egymástól)
- Feljebb lévő feladatoknak **közös kiszolgálást** nyújthat egy lejjebb lévő rész

Hátrányok:

- Adott réteg működéséhez szükséges információk egy másik rétegben
- **Feladatok duplikálása** elkerülhetetlen ➔hatékonyság romlás

SAP: Service Access Point

Nem minden rétegre van szükség minden csomópontban!

Protokoll-adategységek (PDU – protocol data unit)

Általános felépítése, tartalma

Fej (header)	Adat (payload)	[Farok (trailer)]

A fejrészben és farokrészben található kiegészítő információk:

- Címek
- sorszámok
- hibavédelem
- egyéb opcionális paraméterek

Borítékolás – Beágyazás

ISO - International Organization for Standardization

OSI - Open Systems Interconnection

7	Üzenet
6	Üzenet
5	Üzenet
4	Szegmens
3	Csomag
2	Keret
1	

- **Alkalmazási réteg** (*Application layer*)
- **Megjelenítési réteg** (*Presentation layer*)
- **Viszonyréteg** (*Session layer*)
- **Szállítási réteg** (*Transport layer*)
- **Hálózati réteg** (*Network layer*)
- **Adatkapcsolati réteg** (*Data link layer*)
- **Fizikai réteg** (*Physical layer*)

Fizikai Réteg (komm hál 2)

Bitek, bitcsoportok továbbítása a fizikai csatornán

- **vonali kódolás**: szimbólumreprezentáció
- **moduláció**: vivő viszi át az információt

Adatkapcsolati réteg (komm hál 2)

Csomagok összeállítása

- Fizikai címek kezelése (MAC címek)
- Közeghozzáférés vezérlése (Media Access Control)

Hálózati réteg

Egyedi linkek logikai összefűzése **végpontok közötti csatornává**

- **Logikai címzés**: hálózati eszközök közötti kapcsolásokhoz
- **Útvonalkeresés** a hálózaton belül

Tipikusan itt működnek: **útvonalválasztók** (router-ek)

Adatkapcsolati-Hálózati összehasonlítás

- **Adatkapcsolati**: fizikai címzés
- **Hálózati**: logikai címzés

Adatkapcsolati réteg egy hálózaton belül teremt kapcsolatot, **hálózati réteg több hálózaton át**!

Szállítási réteg

Végpont – végpont kommunikáció

- ***Hibamentes*** *összeköttetés létrehozása*
- Hibás csomagok ismétlése
- Duplikált csomagok eldobása
- Csomagsorrend helyreállítása
- *Forgalomszabályozás*

Viszonyréteg

- Kapcsolat irányának kezelése
  - Duplexitás kezelése
- Összeköttetés kezelése
  - **Kapcsolat felépítése, lebontása**

Megjelenítési réteg

Előző rétegek metaadatokat kezelnek

Ez a **felhasználói adatokat** kezeli

- **Adatábrázolás**
  - az átvitt felhasználói információ szintaktikai ellenőrzése
  - szükség szerinti konvertálása (OP rendszer különbségek)
- **Adattömörítés**
- **Adattitkosítás**

Alkalmazási réteg

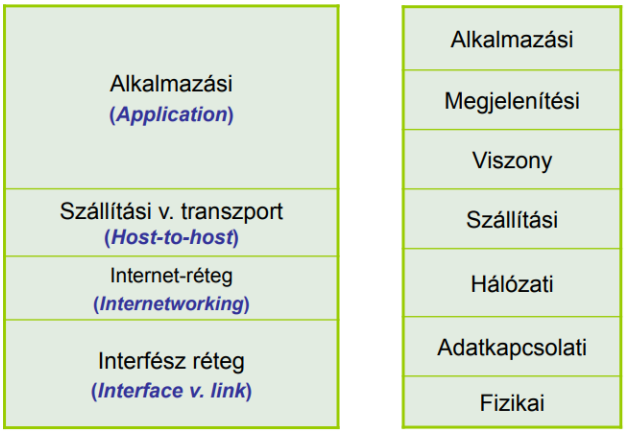
A végpontokon futó **alkalmazási programok**

Az OSI referenciamodell szerepe, jelentősége

- Architektúraként nem valósult meg teljesen soha
- Referenciamodellként továbbra is hasznos

TCP/IP protokoll architektúra

- Rétegek és viszonyítás az OSI modellhez



- **Interfész réteg**
  - kommunikáció egy hálózati szegmensen belül
    - Ethernet
- **Internet réteg**
  - független hálózatokat köt össze
    - Címzés és routing
    - IP
- **Transzport réteg**
  - processz-processz kommunikáció
    - Hibaellenőrzés, forgalomszabályozás, portkezelés
    - TCP, UDP
- **Alkalmazási réteg**
  - interfész a felhasználóhoz
    - FTP, SMTP, http

5-6. előadás

ROUTING – ÚTVONALVÁLASZTÁS

Hálózati réteg

Szegmens átvitele a feladótól a címzettig

Küldő oldal:

- Szegmens -> Csomag (IP fejlécet kap)

Címzettnél:

- Szállítási rétegnek küldi a szegmenst

Router megvizsgálja a fejlécet ➔ Ez alapján dönt

Hálózati réteg feladatai

Routing protokollok <-> ÚTVONAL TÁBLA

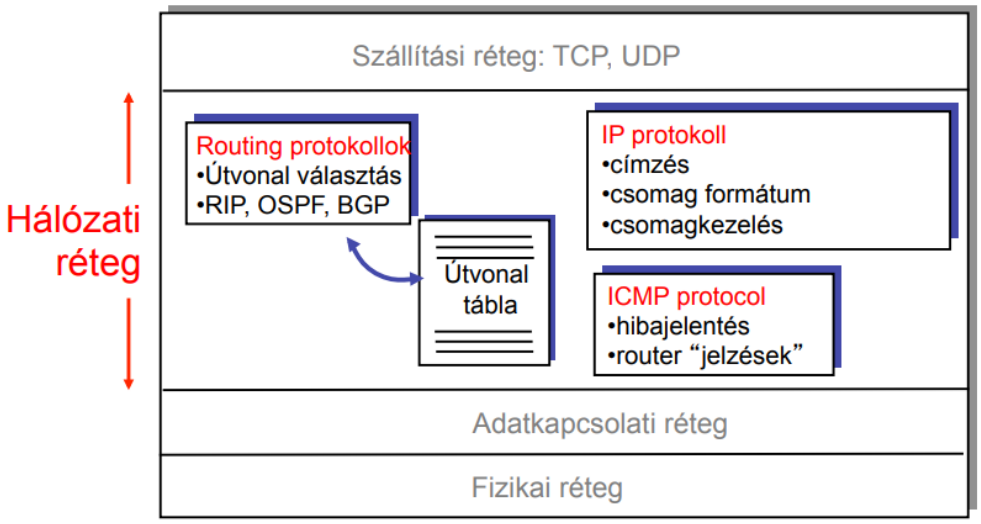
- Útvonal választás
- RIP, OSPF, BGP

IP protokoll

- címzés
- csomag formátum
- csomagkezelés

ICMP protokoll

- hibajelentés
- router “jelzések”



Switching és Routing közötti különbség

Switching:

- **Adatkapcsolati rétegben** MAC címek alapján
- Elárasztást használ: csak helyi hálózatokban

Routing

- **Hálózati rétegben**, IP címek alapján
- Skálázhatóbb megoldás

Routing protokollok: RIP, OSPF, EIGRP, BGP

Routing:

> *Az a mechanizmus, mely segítségével a szállítandó információ a megfelelő úton kerül továbbításra a végpontok között.*

Szűkebb értelemben: csak **útvonal választás**

Tágabb értelemben: csomagok **továbbítása is** (forwarding)

Kihívások:

(Általában) **Nem állandó hálózat méret**: adaptív módon

(Gyakran) **„Nagy" a hálózat mérete**: nincs pontos, aktuális infó a hálózat állapotáról

Routing-hoz tartozik:

- Útvonal választó *módszerek*
- *Algoritmusok*
- Ezeket megvalósító *protokollok*

Útvonaltáblák (forwarding table)

Útvonalválasztás:

Cél-végpont	Következő csomópont
…	…

Táblák kitöltése:

- Manuálisan
- Automatikusan
  - centralizáltan
  - elosztottan

Centralizált (anyuci megcsinál mindent nekünk):

1. egy ponton összegyűjtjük a hálózatról megszerezhető valamennyi ismeretets
2. meghatározzuk az egyes csomópontok útvonaltábláinak bejegyzéseit
3. Ezeket továbbítjuk a csomópontoknak

Elosztott (talpraesett, magának dolgozik):

1. csomópont egyénileg gyűjt infókat a hálózatról
2. saját útvonal táblát készít

Előnyök-Hátrányok:

Centralizált:

- Előny:
  - ✓ Egyetlen kép a hálózatról
  - ✓ Minden útvonal-tábla konzisztens
- Hátrány:
  - × Nagy hálózatnál változhat az állapota
  - × Sérülékenység (m.o. lehet: tartalékolás)

Elosztott:

- Előny:
  - ✓ Kevésbé sérülékeny
- Hátrány:
  - × Nincs egységes kép a hálózatról
  - × Eltérő nézőpontok a csomópontokban

Gyűjthető információ:

Csomópontok közötti összeköttetések, linkek

- Linkek jellemzői
  - Átviteli sebesség (Mbit/s)
  - Forgalom (sorhossz)
  - Terhelés (%)
  - Kiszolgálási díj

A routing módszerrel szembeni követelmények

1. **Minél kisebb méretű útvonaltábla (skálázhatóság)**
   - kisebb tár -> olcsóbb csomópont
   - gyorsabb keresés a táblában
   - kisebb routing forgm

2. **Robusztusság**
   - Hibás táblák esélyének minimalizálása
   - Hibás tábla okozhat:
     - „fekete lyukat”
     - hurkot
     - oszcillációt

3. **Optimális útvonalak kijelölése**
   - optimalizálás igénytől függ
     - legrövidebb (mérték?)
     - legmegbízhatóbb
     - legolcsóbb

Routing a mai kommunikációs hálózatokban

Routing az összekötöttség (**connectivity**) maximalizálására

Módszer: **elosztott**

Két alapvető módszer:

- **Távolságvektor** – Bellman-Ford alg.
- **Linkállapot** – Dijkstra alg.

Módszerek különböznek:

- milyen infót gyűjtenek
- hogyan gyűjtik az infót
- hogyan terítik az infókat

Routing algoritmusok csoportosítása

Globális / Decentralizált információ

1. **Globális:**
   - Minden router tudja a teljes topológiát, linkköltség infót
   - Mindenki maga gyűjti és számolja ki
   - „linkállapot” algoritmusok
2. **Decentralizált:**
   - Router csak a fizikai szomszédok linkjeit, költségeit ismerik
   - Iteratív számolás: szomszédtól hallott infóra alapoz
   - „távolságvektor” algoritmusok

Statikus / Dinamikus

1. **Statikus:**
   - Lassan változnak az útvonalak
2. **Dinamikus:**
   - Gyorsan változnak
   - Periodikus frissítés (topológiai/linkköltség vált. esetén)

A módszerek lényege

Információgyűjtés, és -terítés történik

„Kinek mit mondunk?”

Távolság vektor:

- hálózatról alkotott elképzeléseiket -> szomszédaiknak

Linkállapot

- szomszédaikról nyert tapasztalataikat -> mindenkinek

Távolságvektor módszer

Gyűjtött infó és annak módja:

- hálózatról alkotott elképzeléseiket -> szomszédaiknak

„Elképzelések”:

- melyik csomópont milyen távol van
- lista: csomópont azonosító – távolság párok

Távolság vektorát közli mindegyik csomópont **valamennyi szomszédjával**

Távolság mérése

Egyszerű esetben: **lépések/hop-ok száma**

Súlyozás:

- a link átviteli sebességével
- a sorbanállási hosszal
- a költséggel

távolság = költség

Így a távolságvektor elemei az adott csomópont által elképzelt költségek a hálózat többi (ismert) csomópontjához

Távolságvektor algoritmus

Iteratív, aszinkron:

- Helyi iterációt kiválthat:
  - Helyi link költségének változása
  - Új távolságvektor szomszédtól

Elosztott:

Minden csomópont csak akkor értesíti a szomszédait, ha változik a távolságvektora

Minden Csomópont:

1. **Vár**
   a. helyi link változásra
   b. üzenetre szomszédtól
2. **Újraszámolja elképzeléseit** (Bellman-Ford)
3. **Értesíti szomszédjait**
   a. Ha valamelyik célállomáshoz változott a távolságvektora

A távolságvektor módszer alapproblémája

Végtelenig számolás

- meghibásodások léphetnek fel
- a csomópontok tévesen korrigálhatják távolságvektorukat
- egyre növelik egy adott csomóponthoz való távolságukat

Következmény:

- egymásnak irányítják az el nem érhető csomóponthoz tartó forgalmat, ezzel túlterhelnek linkeket

A távolságvektor módszer javítási lehetőségei

1. **Split horizon**
   a. Megtiltja a csomópont számára az útvonal hirdetését azon interfész felé, ahonnan tanulta
2. **Split horizon with poison reverse**
   a. Ha ő A-tól tanulta meg hogyan éri el B-t, akkor ő A-nak azt hirdeti, hogy B végtelen távra van
   b. (Amennyiben a csomópont egy másik csomóponton keresztül ér el egy harmadikat, akkor ennek a másiknak azt hirdeti, hogy tőle végtelen távolságra van a harmadik.)
3. **Holddown timer**
   a. Router elindít egy időzítőt, amikor olyan értesítést kap, hogy egy csomópont elérhetetlen. Amíg le nem jár, eldob minden értesítést az ebbe a csp-ba vezető útvonalakról.
4. Növekszik a konvergencia idő, további komplexitás

Linkállapot módszer

- szomszédaikról nyert tapasztalataikat -> mindenkinek

Tapasztalatok:

- a szomszédokhoz vezető linkek aktuális állapota (pontosan ismerhető)
- a link költsége valamilyen mérték szerint

Az információ elküldése mindenkinek „elárasztással”

- elküldik a szomszédokhoz, amelyek továbbadják
- kivéve azon a linken, amelyen érkezett

A linkállapot-információk alapján: **legrövidebb utak kiválasztása**

Minden csomópontban **globális link-állapot információk** az elárasztásnak köszönhetően

Linkállapot routing algoritmus

Dijkstra alg.

Link állapot broadcast-nek köszönhetően

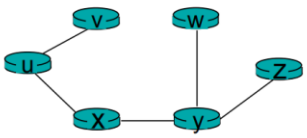
- sorszámozott hirdetések + öregítés
- minden csomópontnak ugyanaz a hálózati képe van

Kiszámolja a legrövidebb útvonalat a **forrástól minden másik csomópontig**

- Kitölti a forrás útvonaltábláját

Dijkstra alg.:

Eredménye: legrövidebb útvonalakból alkotott fa (*u*-ból nézve)



Eredménye: útvonaltábla *u*-ban:

célállomás	kimenő link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Problémák: Oszcillációk lehetségesek

Távolságvektor vs. Linkállapot

- **Üzenet komplexitása:**
  - Linkállapot
    - n csomópont, E link
  - Távolság vektor
  - üzenet csak szomszédoknak
- **Konvergencia idő:**
  - Linkállapot
    - lehetnek oszcillációk
  - Távolságvektor (**konv. idő hullámzik**)
    - lehetnek hurkok
    - végtelenig számolás
- **Robosztusság (Ha router rosszul működik?)**
  - Linkállapot
    - csomópont hírdethet téves **link** költségeket
    - minden csomópont saját tábláját számolja
  - Távolságvektor
    - csomópont hírdethet téves **útvonal** költségeket

Távolságvektor alapú protokollok:

- Rosszabbul skálázódnak
- Lassabb konvergencia
- Kisebb hálózatokban használják őket

(Manapság a linkállapot alapú módszerek dominálnak **tartományon belüli routing** esetén)

Routing tábla metrikák

Routing algoritmustól függ!

1. Linkkapacitás
2. Késleltetés
3. Hopszám
4. Megbízhatóság
5. Csomagvesztési ráta

Hierarchikus routing

Eddig minden router egyforma ➔ **Flat hálózat**

Gyak-ban nem igaz

Skálázhatóság(Több száz csomóponttal):

- Túl nagy útvonaltáblák
- Ezek cserélése leterhelné a kommunikációt
- TV alapú algoritmusok nem konvergálnának

Adminisztratív autonómia

Minden admin maga akarja kontrollálni a routing beállításait

Az autonóm rendszerek

Megoldás: Hálózat szegmentálása

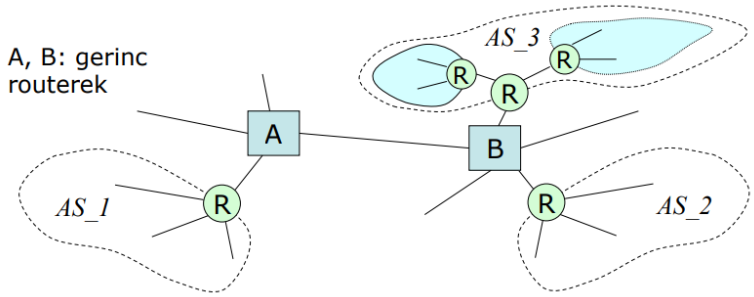
Autonóm rendszerek kialakítása

- **önállóan** oldanak meg **routingfeladatokat**
- a külvilág felé **egyetlen elérési pont**ként szerepelnek

1. Belül **egységes routing policy** érvényesül
2. Egyetlen hálózat, vagy hálózatok csoportja
3. Egyetlen szervezeti egység megbízásából
4. Globálisan egyedi azonosító: ASN (Autonomous System Number)

Az ún. határ-routereken keresztül érhető el

Az egész csoport egyetlen bejegyzés lesz az útvonaltáblákban



AS (Autonomous System) típusok

Multihomed

- Több mint egy AS-el van összekötve
- Meghibásodás esetén sem lesz elvágva
- Viszont nem enged átmenő forgalmat más AS-ek között

Stub (csonk)

- Csak egy másik AS-el van összekötve
- (Gyakorlatban többel is van peering, csak nem nyilvános)

Tranzit

- Csak átmenő forgalmat szolgálnak ki
- (Összes ISP ilyen)

AS-k közötti útvonalválasztás

Multihomed és tranzit AS esetén több útvonal közül választhat más AS-ek felé

Hot-potato:

- minél rövidebb legyen az út a saját AS-én belül

Cold-potato:

- minél hosszabb ideig a saját hálózatán belül akarja tartani a forgalmat, szinte a felhasználóig
- szolgáltatás minőségi garanciákat nyújthat

AS-en belül: AS-en **belüli routing algoritmus**
AS-en kívül: **AS-en belüli és** AS-ek **közötti** routing **algoritmus**

Mobil végpontok kezelése

Eddig: egy végpont a csomóponthoz való kapcsolata definiálja

A végpontok átmozoghatnak más **csomópont környezetébe**, **más autonóm** rendszerbe

- mozgás: vándorló/**mobil végpont**

A többi végpont csak a mozgó csomópont eredeti (pl. Ba) címét ismeri

- Jó lenne tudnia hol van aktuálisan hol van Ba

Hálózat kezeli Ba mozgását ➔**Mobil routing** feladata

Kiegészítések a csomóponton:

Mobil ügynök:

- **Home agent** = hazai ügynök
- **Foreign agent** = idegen ügynök

Multicast routing

Érdekes feladat: multicast routing (egy időben több végponthoz/címre kell csomagot eljuttatni)

Elnevezések:

- **Broadcast** – Mindenkinek
- **Multicast** – egy csoportnak
- **Anycast** – egy valakinek a csoportból
- **Unicast** – egy végpontnak

Megoldandó feladatok:

- Címzés
- Csoportkezelés
- Útválasztás segítése

A többescímzés, csoportazonosítás

Csomópontot azonosítják:

- Fizikai címeik
- Logikai címeik

A csoportot **egy**, az egyedi logikai címhez hasonló **azonosító** különbözteti meg: **csoport-cím**

Csatlakozás/Kiválás csoportból: **IGMP** (Internet Group Management Protocol), multicast végpont és routere között.

Multicast routing, módszerek

A csoport résztvevői a hálózatban:

Sűrű elhelyezkedés

- Elárasztás + lemondás
  - Reverse path forwarding ➔ Legrövidebb útvonal fa alkalmazása
  - Csoportlemondás
  - Forráslemondás

Ritka elhelyezkedés:

- Explicit csatlakozás
  - Gerincalapú fa alkalmazása

Detektálni kell melyik eset: **PIM** (Protocol-Independent Multicast)

Protokollok multicasthez

- Multicast végpont és routere között: **IGMP**
- Multicast routerek között: **multicast protokollok** (pl. PIM)
- **IGMP + PIM**: egy routing zónán belül
- Routing zónák között is kell

7-8-9. előadás

Az IP feladata

Adattovábbítás a hálózatok végpontjai között

Két fő funkció

- Címzés (addressing) és útvonalválasztás (routing)
- Tördelés/fragmentálás (fragmentation)

Az IP jellemzői

- Csomagkapcsolt
- Összeköttetés-mentes (connectionless)
- „Best effort” – nincs garancia
    - A csomagokra igaz:
        - Elveszhetnek
        - Duplikálódhatnak
        - Sorrendjük megváltozhat
        - Meghibásodhatnak (nincs hibajavítás, hibaészlelés csak fejlécre)
- Egyéb **NEM** nyújtott szolgáltatások:
    - Torlódáskezelés
    - Ütemezés

IP – Címzés

4 bájtos cím (32 bit)

Hálózat azonosító | Hálózaton belüli azonosító

Alhálózatok

IP cím: Alhálózati rész | Host rész

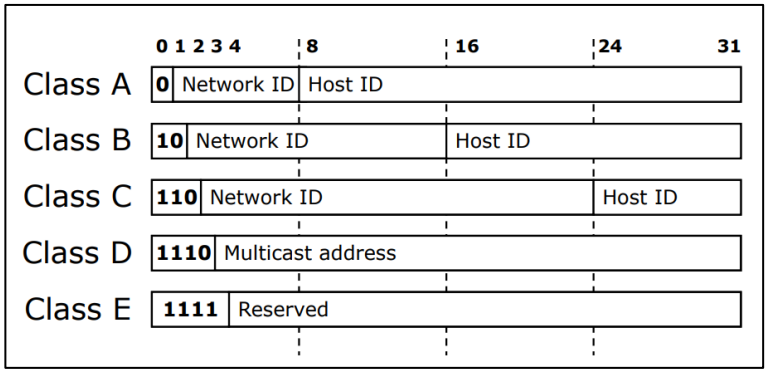
Mi az alhálózat?

- interfészek: megegyezik az IP címük alhálózati része
- fizikailag elérik egymást, akár router nélkül is

Címosztályok

Kezdetben fix

1. **A, B és C osztályú címek**
    a. Egyedi címzésre (unicast)
2. **D osztály**
    a. Többes címzésre (multicast)
3. **E osztály**
    a. Fenntartott osztály és címtartomány



Speciális címek

- Host ID: csupa 0 ➔ Hálózat címe
- Host ID: csupa 1 ➔Broadcast cím
- Loopback interfész (127.0.0.0)

Címosztályok általánosítása

CIDR – Classless Inter-Domain Routing

A címből nem lehet megállapítani, hol lett kettéosztva ➔ **alhálózati maszk**

Privát címtartományok

Ötlet: pl. cégen belül nem kell globálisan egyedi IP cím ha nem kommunikál kifelé

Ha kifelé is szeretne: **Network Address Translation** (NAT)

Privát IP-címtartományok (csak helyi hálózaton érvényes címek)

10.0.0.0 – 10.255.255.255	(/8, 1 db A osztály)
172.16.0.0 - 172.31.255.255	(/12, 16 db B osztály)
192.168.0.0 - 192.168.255.255	(/16, 256 db C osztály)

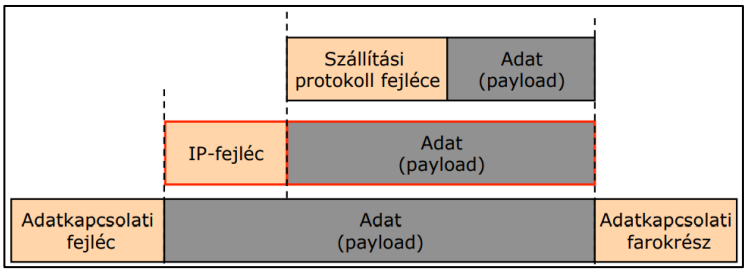
NAT: Network Address Translation

Minden csomagnak, amely elhagyja a helyi hálózatot egyforma NAT IP forráscíme van

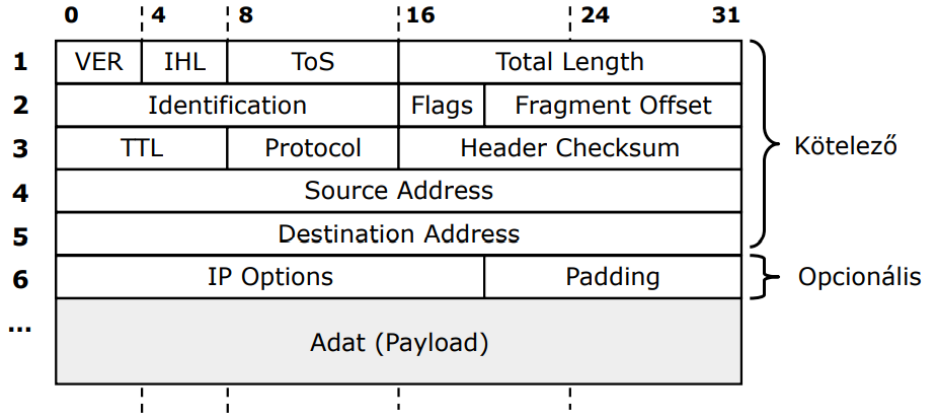
Az IP-csomag szerkezete és „helye”

Az IP-csomag két része:

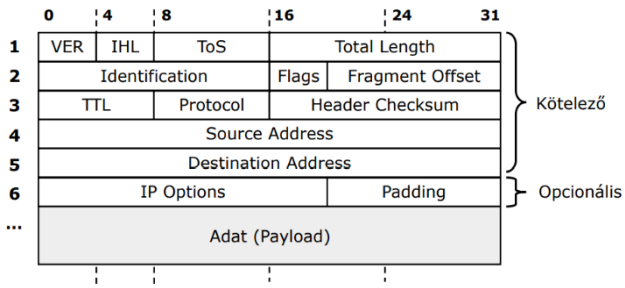
- IP-fejléc / fejrész (IP header)
- Adat (payload)



IP – Fejléc

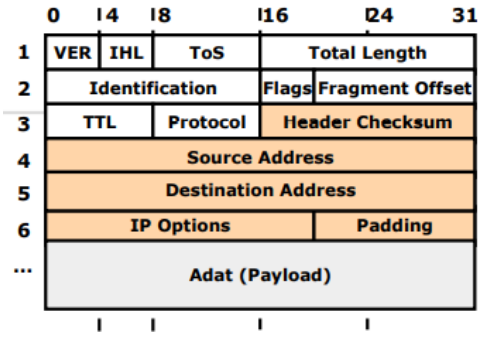


1. **VER** – Version (4 bit)
   a. Az Internet Protocol verziójának száma
   b. IPv4 : 4, IPv6 : 6
2. **IHL** – Internet Header Length (4 bit)
   a. Az IP fejléc mérete 32 bites szavakban
   b. Értéke:
      i. Minimum: **20 byte**
      ii. Maximum: **60 byte**
3. **Total Length** (16 bit)
   a. A teljes IP csomag mérete bájtokban



Nemes Axel Roland

4. **ToS** (Type of Service)
    a. Legtöbb router nem támogatja
    b. Leggyakrabb felhasználás:
        i. **DSCP** – Differentiated Services Code Point
        ii. **ECN** – Explicit Congestion Notification
5. **Identification** (16 bit)
    a. Az IP-töredékek egyedi azonosítása
6. **Flags** (3 bit)
    a. 0: Fenntartott
    b. 1: DF – Don't Fragment
        i. **1**: ha tördelni kellene, el kell dobni
    c. 2: MF – More Fragment
        i. **1**: ha nem az utolsó töredék
        ii. **0**: utolsó töredék vagy nem tördelt csomag
7. **Fragment Offset** (13 bit)
    a. Az eredeti csomagban lévő kezdőpozícióját adja meg e töredékben lévő adatnak
8. **TTL** – Time To Live
    a. Csomag élettartama
    b. Eredetileg: másodperc
    c. Gyakorlailag: hopszám
9. **Protocol**
    a. Az adatrészben lévő protokoll azonosítója
10. **Header Checksum** (16 bit)
    a. Fejrész ellenőrző összege
    b. Csomag érkezéskor: Ellenőrzés
    c. Továbbításnál újra számolni (TTL változása, fregmentáció)
11. **Source / Destination Address**
    a. Feladó és Címzett IP címe
    b. NAT miatt lehet nem a feladó tényleges címe
12. **IP Options**
    a. Ritkán használt opcionális része a fejlécnek
13. **Padding**
    a. Az IP Options részt egészíti ki 4 bájt többszörösére



IP – Útvonalválasztás

1. **Kinek küldjük tovább**?
    a. Cél címe alapján ➔ csomag tartalmazza
    b. Saját ismeret alapján ➔ **útvonal-irányítási táblázat** (routing table)

Routing

---

2. **Hogyan küldjük tovább?**
    a. Mekkora egységekben?
    b. Mekkora érkezik? ➔ csomag határozza meg
    c. Mekkora továbbítható? ➔ a **következő hálózat** **MTU**-ja (Maximum Transmission Unit) **határozza meg**

Tördelés

---

3. **Milyen QoS biztosításával?**
    a. „Best effort" – nincs garancia
    b. Opcionális megoldás: ToS mező

QoS

Útválasztó tábla (Routing table)

Merre megy?		Merre küldjük?		
Hálózat címe	Alhálózati maszk	Interfész	Közvetlenül kapcsolódó	Következő csomópont
<IP-cím>	</N>	<azonosító>	<igen/nem>	<IP-cím>
<IP-cím>	</N>	<azonosító>	<igen/nem>	<IP-cím>
<IP-cím>	</N>	<azonosító>	<igen/nem>	<IP-cím>

Hálózat kiválasztása: Hálózati mask ➔ Bitenkénti AND ➔ Összehasonlítás bejegyzésekkel

Ha több hálózatra is illeszkedik: **leghosszabb egyezőséget** kell választani!

Az útválasztó tábla összes bejegyzését végig kell nézni ➔ **rosszul skálázódik**

Alapértelmezett útvonal (Default route)

Erre megy a csomag, ha nem ismeri a küldő a célhálózatot

Végpontokon gyakran csak két bejegyzés szerepel:

- **Helyi hálózat** (helyi végpont közvetlenül elérhető)
- **Alapértelmezett útvonal** (minden más távoli hálózaton)

Teendők helyi és távoli hálózat esetén

Közvetlenül kapcsolódó (**helyi**) hálózat:

- A címzettnek közvetlenül küldeni
  - ➔ adatkapcsolati rétegbeli (**MAC**) **címére** van szükség

Nem közvetlenül kapcsolódó (**távoli hálózat**):

- Az útválasztónak kell küldeni
  - A csomag címzettjének IP-címét **tilos módosítani**
  - Csak adatkapcsolati rétegben kell az útválasztónak címezni

IP – ARP/RARP

ARP – Address Resolution Protocol

Ha van IP de kellene MAC

Az ARP üzeneteket közvetlenül az adatkapcsolati réteg protokolljának küldjük (**Nem IP-csomag**!)

Címfeloldás ARP-vel

Füzetben 1. mérés jegyzet

RARP – Reverse ARP

Van MAC kellene IP

Nem használják, helyette:

- Bootstrap Protocol (BOOTP)
- Dynamic Host Configuration Protocol (DHCP)

IP – Útvonalválasztó protokollok

Útvonalválasztó protokollok funkcionális osztályozása

1. **Ad hoc** routing protokollok
    a. Kis és gyorsan változó hálózatokra
2. Interior Gateway Protocols (**IGPs**)
    a. Autonóm rendszereken (AS) belül
        i. Kisebb hálózatokon
3. Exterior Gateway Protocols (**EGPs**)
    a. Autonóm rendszerek (AS) között
        i. Az Internet routing protokollja

Interior Gateway Protocols (IGPs)

Autonóm rendszereken (AS) belül

Távolság-vektor:

- **RIP** (Routing Information Protocol)
- **IGRP** (Interior Gateway Routing Protocol)
- **EIGRP** (Enhanced Interior Gateway Routing Protocol)

Összekötés-állapot:

- **OSPF** (Open Shortest Path First)
- **IS-IS** (Intermediate System to Intermediate System)

Távolság-vektor protokollok: RIP

Metrika: **hopszám**

Lassan konvergál

Nem skálázódik jól: **nincs hierarchia**

Hopszám limit korlátozza a hálózat méretét (max 15)

Végtelenig számolás ellen: **split horizon** + **route poisoning**+ **holddown timer**

Könnyen konfigurálható!

Távolság-vektor protokollok: EIGRP

IGRP: Cisco specifikus

- Javította RIP hibáit
- Max 255 hop
- **Többféle metrika** figyelembe vétele

EIGRP: IGRP javítása - Variable Length Subnet Mask használata

- Gyorsabb konvergencia
- Hurokmentes működés
- 5 metrika, alapból csak 2 metrika kombinációja (sávszélesség + útvonal késleltetése)

Összeköttetés-állapot protokollok: **OSPF**

- Link state database (LSDB) üzenetek **elárasztással**
- Minden csomópontban **topológia térkép**
- Ebből **legrövidebb útvonalú fa** Dijkstra algoritmussal
- **IP** üzenetekben
- Többféle metrika
- **Autentikált** OSPF üzenetek (támadás ellen)
- **Hierarchikus**: gerinc - routing tartományok, köztük határ routerek

OSPF vs. IS-IS

OSPF (2-es verzió):

- IPv4-re tervezték: népszerűbb
- Át kellett specifikálni IPv6-ra

IS-IS:

- OSI fejlesztés, nem IP csomagokban
- Nagyobb kiterjedésű hálózatokra
- könnyebb átállás IPv6-ra

Exterior Gateway Protocols (EGPs)

Autonóm rendszerek (AS) között

- **EGP** (Exterior Gateway Protocol)
  - Sokáig az Internet EGP-je, ma már nem használt
- **BGP** (Border Gateway Protocol)

Címaggregáció (supernet)

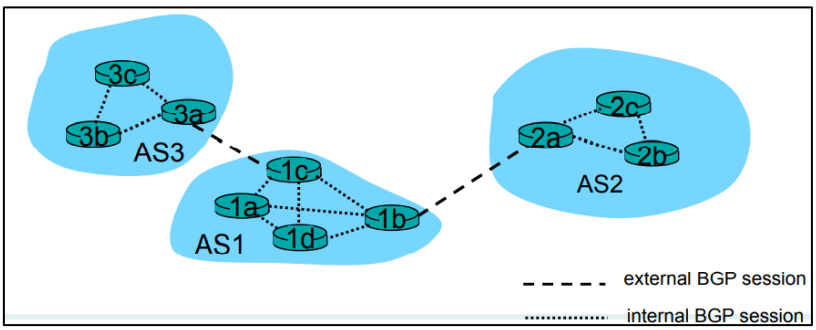
- Kisebb routing táblák
  - Gyorsabb keresés
  - Gyorsabb terjesztés

BGP alapok

Routerek között (BGP peer-ek) TCP kapcsolatok: BGP session

A hirdetett célpont: nem terminál, hanem hálózati prefix

AS2 hálózati prefix-et hirdet AS1-nek: AS2 megígéri, hogy kézbesít minden, a prefix-nek küldött csomagot



Útvonal attribútumok & BGP útvonalak

prefix + attribútumok = útvonal – *Ezt hírdetik egymásnak*

Két fontos attribútum:

- **AS-PATH**: azon AS-ek sorozata, amelyeken áthaladt a prefix hirdetés
- **NEXT-HOP**: a következő lépésbeli AS-hez vezető, AS-n belüli, router, tőle indul az AS-PATH

Gateway router: **import policy** alapján dönti el, elfogadjae az útvonalat

BGP útvonal választás

Ha a router egynél több útvonalat kap egy prefix-hez

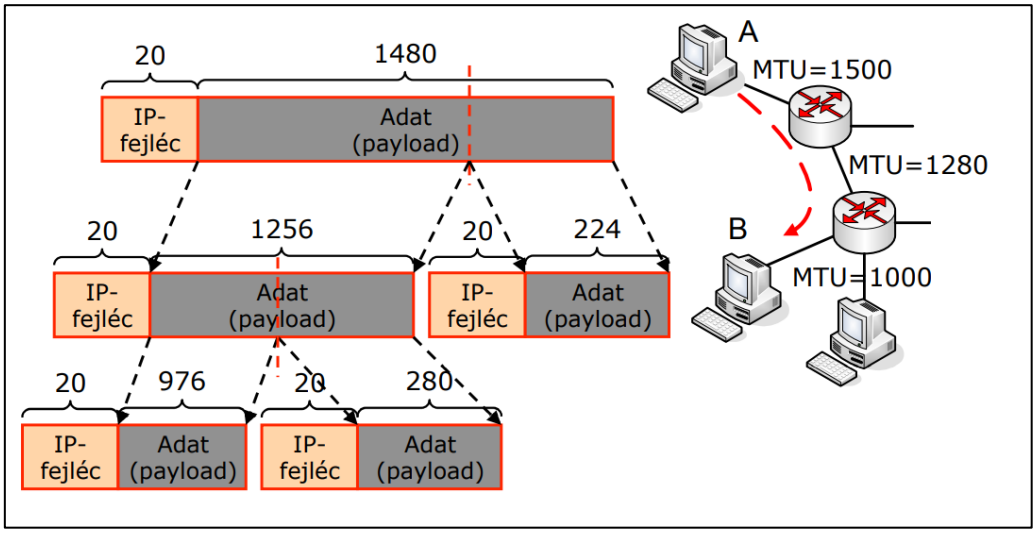
Döntési sorrend:

1. Az útvonal helyi preferencia értéke: **policy**
2. **Legrövidebb AS-PATH**
3. **Legközelebbi NEXT-HOP router**: hot potato routing
4. További szempontok

IP – Tördelés

Hálózatok meghatározzák a keret max méretét

Az adatkapcsolati réteg fej- és farokrészét leszámítva ez az **MTU** (Maximum Transmission Unit)



IP fejléc hossza vagy a min **20** vagy a max **60 bájt**!
AZ ADAT RÉSZ MÉRETÉNEK 8-AL OSZTHATÓNAK KELL LENNIE!

Ha tördelés szükséges:

1. **Total Length mezőt** a töredék méretére állítani
2. Ha nem volt **Identification** mező érték, akkor generálni kell
3. Adat tördelése **8 bájtos egységekre**
4. A tördelésnek megfelelő **fragmentation offset**-et beállítani az új csomagokban
5. Minden csomagban az **MF bit**et 1-re kell állítani, kivéve annak a csomagnak az utolsó töredékénél
6. **Ellenőrző összeg** változása

A töredékek összeállítása

Csak a címzett végezheti el (Kivéve pl. NAT)
Ha valamelyik nem érkezik be ➔ **Többit is eldobja**
Csak teljesen összeállított csomagokat továbbít

A router feladatai

- Hibás-e a csomag (fejléce)?
- Nekem címezték-e?
- Ismerem-e a címzett hálózatát?
- A TTL érték csökkentés után >0?
- Kell-e tördelni? Lehet-e tördelni?
- Kell-e visszajelzést küldeni?

ICMP

Jelzés- és menedzsmentüzenetek

Üzenettípusok

Hibaüzenetek

Kérdések

Válaszok

Echo request – Echo reply ➔ Ping (RTT)

Echo request – Echo reply + TTL ➔ Traceroute

ICMP csomag és fejléce

IP csomagba ágyazva

Best-effort szolgáltatás: de **különleges elbánást** kap a feldolgozásnál

Fejléc:

- **Típus**on belül **kód**
- **Ellenőrzőösszeg** egész ICMP üzenetre

Hibaüzenetnél: a hibás üzenet fejléce és adatmezőjéből az első 8 byte

10- előadás

SZÁLLÍTÁSI PROTOKOLLOK (UDP és TCP)

A hálózati és a szállítási réteg

hálózati réteg: hostok közötti logikai kapcsolatok

szállítási réteg: alkalmazások közötti logikai kapcsolatok

Feladatai:

- forgalomszabályozás és torlódáskezelés
- multiplexelés
- hibadetektálás, javítás
- sorrendhelyes átvitel

A szállítási réteg

Logikai kapcsolatok a végpontokban futó **alkalmazások között**

A **szállítási protokollok a végpontokban futnak**, a csomópontokban nem

Szállítási protokollok: UDP és TCP

UDP – User Datagram Protocol

TCP – Transmission Control Protocol

Közös képességek:

- Portok kezelése
- Multiplexelési képesség

Különbségek:

- **UDP** összeköttetésmentes, megbízhatatlan
- **TCP** összeköttetés-alapú, megbízható

Az UDP és TCP közös képességei

Portok kezelése:

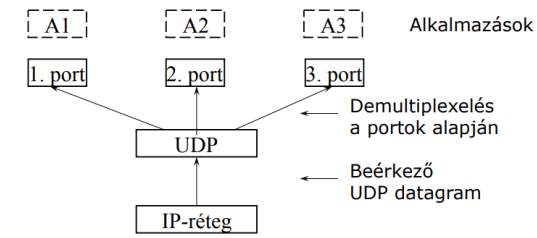
Végpontokon belül több folyomat ➔ Megkülönböztetés: Portokkal

System/User/Dynamic ports

Multiplexelés /demultiplexelés:

- A portmechanizmus segítségével

Multiplexelés és demultiplexelés
Példa:



Socket: magyarázat

interfész, „ajtó” az alkalmazás és a hálózat között (*socket = IP cím + portszám***)**

A socket-et az alábbiak jellemzik:

- transzport protokoll (UDP v. TCP)
- saját IP cím
- saját portszám

Helyi socket cím

- (opcionális) távoli host IP címe
- (opcionális) távoli alkalmazás portszáma

Távoli socket cím

Socket pár

Socket típusok

- **Datagramm socket**
  - Összeköttetésmentes socket, UDP használja
- **Stream socket**
  - Összeköttetés-alapú socket, TCP használja
- **Raw (IP) socket**

TCP kliens-szerver socket kezelés

- Szerver létrehoz „**hallgató**” módban lévő **socket**eket
- **Kapcsolatfelvétel után**: **dedikált socket** minden kapcsolathoz (TCP kapcsolat, duplex bájt folyam)
- Szerver különböző TCP socketeket hozhat létre ugyanazzal a helyi IP címmel és port számmal
- UDP-ben nincs dedikált socket

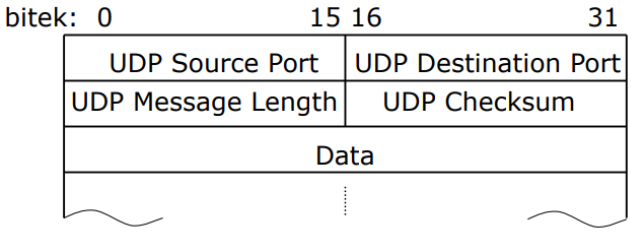
Egyszerű (raw) socket

Közvetlenül az alkalmazásnak továbbítja a csomagot a fejléccel (nincs TCP/IP feldolgozás)

Félelem TCP reset támadástól: „lelövi” a TCP kapcsolatot

UDP (User Datagram Protocol)

Nincs kapcsolatfelépítés, kézfogás



Nincs garancia:

- a csomagok sorrendjére
- elvesztésének detekciójára

megbízható átvitel garantálását az alkalmazási rétegre bízza

Az ellenőrző összeget „pseudo-header” hozzáadásával együtt számolja (IPv4-re)

Miért használunk UDP-t?

1. Kisebb késleltetés
2. Egyszerű: nincs kapcsolat állapot kezelés
3. Kis méretű fejléc
4. Nincs torlódásvezérlés: gyorsan eléri a maximális adatátviteli rátát

UDP – User Datagram Protocol Összefoglalás: funkcionalitás és a költsége

- **Portkezelés**
  - a különböző alkalmazások/folyamatok megkülönböztethetők
- **Több alkalmazás egyidejű kezelése**
  - port-hozzárendeléssel és multiplexeléssel/demultiplexeléssel
- **Hibajelzés az UDP datagram tartalmára és az IP csomag további részeire**

Ezek költsége: **min 8 oktettnyi overhead**

UDP alkalmazása

Média: streaming, valós idejű játékok, VoIP, IPTV

Gyors és rövid lekérdezések:

- DNS
- DHCP
- RIP
- NTP

Tipikusan a hálózati forgalom pár százaléka

TCP –Transmission Control Protocol Fő jellemzői

- Pont-pont
- Stream–típusú szolgáltatás
- Pufferelt átvitel

Duplex kapcsolatok

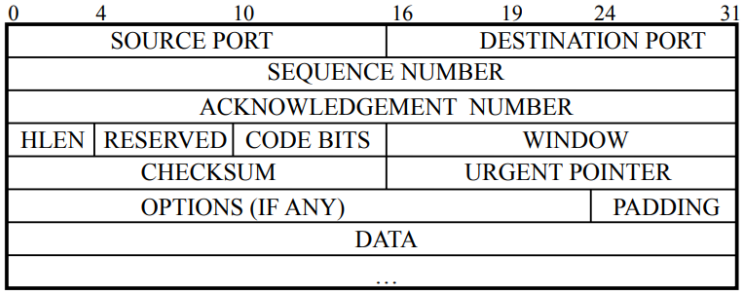
- két független stream

Összeköttetés alapú

- összeköttetés épül fel és marad fenn a kommunikáció tartamára
- 3 utas kézfogás

Vezérlő információk küldése (piggypacking)

A PDU – Protocol Data Unit (a TCP-ben: „segment”) struktúrája



1. **Sequence no**.: a szegmensben levő adat első byte-jának pozíciója
2. **Ack no**.: ezt a sorszámú szegmenst várja a forrás
3. HLEN (**Data offset**): fejléc mérete (min 20 max 60 byte)
4. Code bits (**flags**): CWR, ACK, RST, SYN, FIN, …
5. **Window**: vételi puffer mérete
6. **Checksum**: mint az UDP-ben (pszeudoheader)
7. **Urgent pointer**: arra mutat ami már NEM fontos/urgent

MSS (Maximum Segment Size)

Legnagyobb adatméret bájtban, amit a TCP hajlandó küldeni egy szegmens adat részében

Egyeztetni kell az adatkapcsolati réteg MTU-jával

Kapcsolat kiépítésnel lehet jelezni **MSS opció fejlécben**

Várakozás nyugtázásra

Várakozás ACK-ra: time–out

RTT-hez igazítani, **adaptívvá** tenni

$$EstimatedRTT = (1- \alpha)*EstimatedRTT + \alpha*SampleRTT$$

SampleRTT = Pillanatnyi RTT

Alfa jellemzően = 0.125

Time-out kétszerezés

amennyiben lejár a timer és újraadás van ➔ **2 * time-out értéke**

Nyugta vétele után visszaáll az EstimatedRTT és DevRTT kombináció

Torlódásvezérlés egyik eszköze

Gond összevont (kumulatív) nyugtával

Megoldás: **szelektív nyugtázás (SACK)**

SACK-ban meg tudja mondani mettől-meddig érkezett meg

Gond sorrend keveredéssel

Megoldás: **D-SACK (duplikált nyugta)**

Fogadó szól, hogy az újraküldött csomag duplikátum ➔ visszagyorsul a forrás

Fast retransmit

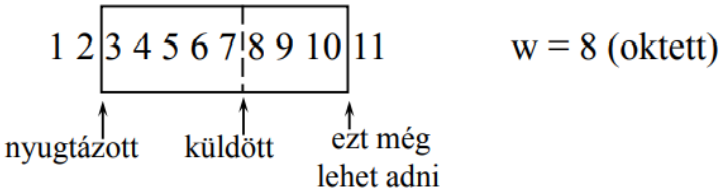
A time-out idő gyakran túl hosszú:

- nagy késleltetés mielőtt az elveszett csomagot az adó újra tudná küldeni

Fast retransmit szabály: ha az adó 3 egymást követő ACK-t kap (plusz az eredeti ACK) ugyanarra a szegmensre, feltételezi, hogy az azt követő szegmens elveszett és

- **újraküldi azt még mielőtt lejárna a timeout**

Forgalomszabályozás: a „sliding window” módszer elve



Méret(w) = A max megengedett nem nyugtázott csomagok száma

TCP forgalomszabályozás: hogy működik?

1. Vevő közli a szabad helyének a méretét: RcvWindow
2. Az Adó legfeljebb RcvWindow nyugtázatlan adatot küld

Nem szabad összekeverni a torlódási ablakkal (CongWin)

Csúszóablakos problémák

Ha a vevő **nullás csúszóablak méretet** hirdet: adó leáll a küldéssel

- Megoldás: adó egy timert indít, lejárta után felkéri a vevőt, hogy küldjön ACK-ot az új ablakméretről

Buta ablak jelenség

- Kicsi ablakot nyit ➔ Erőforrás pocséklás
- A vevő nem nyitja az ablakot, csak akkor, ha MSS nagyságrendűt nyithat

Torlódásvezérlés a TCP-ben

Hálózat által segített (Network assisted):

- Hálózati elemek szólnak az adónak a terhelésről

Végpontok közötti (end-to-end):

- Nincs visszacsat nincs hálózati elemektől, végpont következtet

TCP a **Végpontok közötti torlódásvezérlést** használja!

Módszer:

- Van elég átbocsátóképesség? ➔ Adatsebesség növelése
- Van torlódásra utaló jel? ➔ Akkor csökkentünk

Hogyan (*AIMD*):

- Torl. ablak-ot növeljük MSS-al minden RTT alatt amíg nincs vesztés
- Torl. ablak csökkentése felére minden vesztéskor

Hogyan érzékeljük a torlódást (vesztést)?

- timeout letelt, nem jött nyugta
- többszörös nyugta érkezett ugyanarra a szegmensre

Slow Start:

- kapcs. elején sebesség exponenciális növelése első vesztésig/korlátig
- utána AIMD

Csomagvesztés

3 duplikált ACK után:

1. Torl. ablak felére csökken
2. Utána lineáris növelés
    a. **Torlódás elkerülése fázis**

Timer letelte után:

1. Torl. ablak 1 MMS lesz
2. Utána exponenciálisan korlátig
3. Utána lineárisan
    a. **Torlódás elkerülése fázis**

Összefoglaló: TCP torlódás vezérlés

- Ha a **CongWin** egy korlát alatt van, adó **slow-start fázisban**, exponenciális ablak növelés
- Ha **CongWin** a korlát felett, az adó **torlódás elkerülési fázisban**, lineáris ablak növelés
- Ha **3 duplikált ACK**: a korlát CongWin/2 lesz és a CongWin pedig a korlát
- Ha **timer lejárt**: a korlát CongWin/2 lesz, míg a CongWin 1 MSS lesz

TCP implementációk

3 duplikált ACK a 9.átvitelnél:

- **Tahoe**: Mintha timer járt volna le (1MMS-ről, exp. korlátig, lineáris)
- **Renoe**: Torl. ablak/2-ről (Timeout ➔ Tahoe)
- Vegas
  - Minden szegmensnél Timer, RTT mérése
  - RTT nő ➔ Torlódás kialakulása ➔ Lineáris növelés/csökkentése
- Cubic
  - Torl ablak harmad fokú függvény

TCP a valóságban:

- Kisebb RTT-vel rendelkező kapcsolatok gyorsabban növelik a CongWin-t ➔ Nagyobb átviteli sebesség
- UDP kiszorítja a TCP-t
- Párhuzamos TCP kapcsolatok
  - Több kapcsolat ➔ Nagyobb átviteli sebesség

TCP és UDP: összefoglalás

- Mindkettő **szállítási rétegbeli** protokoll
- Mindkettő portokat kezel
  - multiplexelés/demultiplexelés
  - ezáltal interfész nyújtása az alkalmazási folyamatok felé
- Az **UDP** összeköttetés-mentes, best effort szolgáltatás
  - nem garantál célba juttatást, csak hibajelzést nyújt
  - gyorsan célba juttat
- A **TCP** összeköttetés-alapú, megbízható transzport szolgáltatás
  - sorrendhelyes, hibamentes szállítást nyújt
  - ***ára: késleltetés***

Alkalmazás	***Alkalmazási rétegbeli protokoll***	***Használt transzport-protokoll***
E-mail	SMTP	TCP
Távoli elérés	Telnet	TCP
Web-elérés	HTTP	TCP
File-átvitel	FTP	TCP
Routing	RIP	UDP
Hálózatmenedzsment	SNMP	UDP, TCP
VoIP, média-streaming	Többnyire nem szabványos	UDP, TCP