



10

**előadás**

# Számítógépes hardver 3

## KIN/PS/IN/12

# FORGALOMIRÁNYÍTÁS

---

Ing. Ondrej Takáč, PhD.

Informatika Tanszék

Gazdaságtudományi és Informataikai Kar

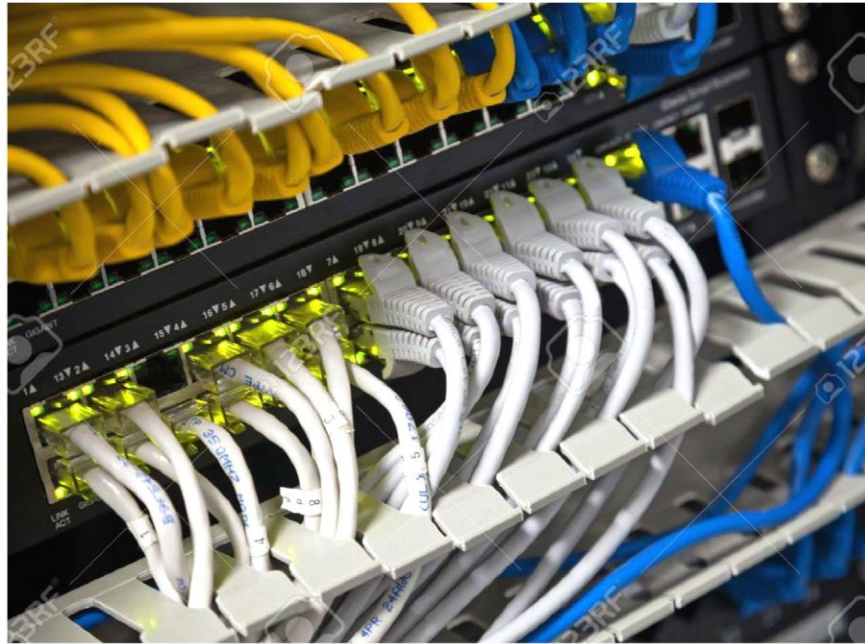
Selye János Egyetem

Tel.: +421 35 32 60 629

Email: [takac.ondrej@gmail.com](mailto:takac.ondrej@gmail.com)

**streda, 28. apríla 2021**





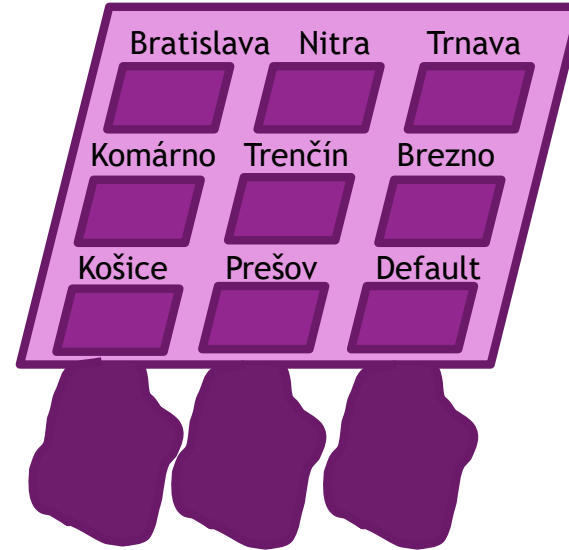
## ROUTING

# ROUTING



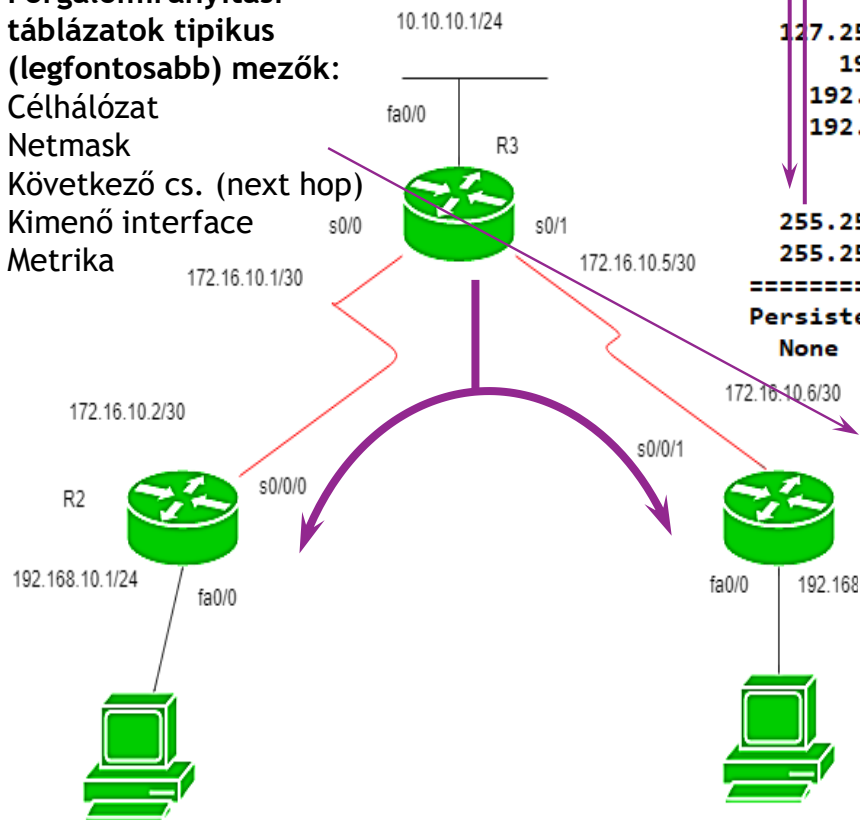
Mail sorting office in Wellington General Post Office, New Zealand c.1900

[https://en.wikipedia.org/wiki/Sorting\\_office](https://en.wikipedia.org/wiki/Sorting_office)



# ROUTING

Forgalomirányítási  
táblázatok tipikus  
(legfontosabb) mezők:  
Célhálózat  
Netmask  
Következő cs. (next hop)  
Kimenő interface  
Metrika



## PowerShell : netstat -r

### IPv4 Route Table

#### Active Routes:

| Network | Destination     | Netmask         | Gateway     | Interface     | Metric |
|---------|-----------------|-----------------|-------------|---------------|--------|
|         | 0.0.0.0         | 0.0.0.0         | 192.168.0.1 | 192.168.0.102 | 50     |
|         | 127.0.0.0       | 255.0.0.0       | On-link     | 127.0.0.1     | 331    |
|         | 127.0.0.1       | 255.255.255.255 | On-link     | 127.0.0.1     | 331    |
|         | 127.255.255.255 | 255.255.255.255 | On-link     | 127.0.0.1     | 331    |
|         | 192.168.0.0     | 255.255.255.0   | On-link     | 192.168.0.102 | 306    |
|         | 192.168.0.102   | 255.255.255.255 | On-link     | 192.168.0.102 | 306    |
|         | 192.168.0.255   | 255.255.255.255 | On-link     | 192.168.0.102 | 306    |
|         | 224.0.0.0       | 240.0.0.0       | On-link     | 127.0.0.1     | 331    |
|         | 224.0.0.0       | 240.0.0.0       | On-link     | 192.168.0.102 | 306    |
|         | 255.255.255.255 | 255.255.255.255 | On-link     | 127.0.0.1     | 331    |
|         | 255.255.255.255 | 255.255.255.255 | On-link     | 192.168.0.102 | 306    |

#### Persistent Routes:

None

Sorokból több ezer is lehet - Cél IP \* maszk - 0.0.0.0

| Sít         | Maska         | Next Hop         | Sítové rozhraní | Metrika |
|-------------|---------------|------------------|-----------------|---------|
| 192.168.1.0 | 255.255.255.0 | 192.168.254.5    | Seriál 1        | 4       |
| 10.1.2.0    | 255.255.255.0 | Lokální rozhraní | Ethernet        | 0       |
| 10.5.1.0    | 255.255.255.0 | 10.10.10.2       | Seriál 2        | 3       |
| 10.5.0.0    | 255.255.0.0   | 10.5.5.5         | Seriál 1        | 2       |
| ...         |               |                  |                 |         |
| 0.0.0.0     | 0.0.0.0       | 10.10.10.2       | Seriál 2        | 1       |

**Forgalomirányítás (routing):** Csomagok (IP datagramok) továbbítási irányának meghatározásával kapcsolatos döntések meghozatala.

# HÁLÓZATI PROTOKOLLOK

## FORGALOMIRÁNYÍTÁSI FELOSZTÁSA

- ◉ **Forgalomirányított protokoll (routed protocol):** Olyan hálózati réteghez kötődő általános adatszállító protokoll, amelyet a forgalomirányító (router) irányítani képes (pl. IP, IPX).
- ◉ **Forgalomirányítási protokoll (routing protocol):** A forgalomirányítási táblázat(ok) felépítéséhez szükséges információk továbbítását (routerek közötti cseréjét) leíró protokoll (pl. RIP, OSPF, BGP).
- ◉ **Egyéb protokoll:** Az előzőekhez nem sorolható hálózati protokoll (pl. ICMP).

# FORGALOMIRÁNYÍTÓK (ALAPVETŐ) MŰKÖDÉSE

- ◉ A router az input interfészen érkező csomagot fogadja.
- ◉ A router a csomag célcímét illeszti a routing táblázat soraira.
- ◉ Ha a célcím több sorra illeszkedik, akkor a leghosszabb prefixű sort tekintjük illeszkedőnek.
- ◉ Ha nem létezik illeszkedő sor, akkor a cél elérhetetlen, a csomag nem továbbítható.
- ◉ A csomagot a router eldobja és ICMP hibajelzést küld a feladónak.
- ◉ Ha létezik illeszkedő sor, akkor a csomagot az ebben szereplő kimeneti interfészen továbbítjuk (adatkapcsolati rétegbeli beágyazással) a következő hopként megadott szomszédhoz, ill. a célállomáshoz, ha már nincs több hop.



# IP CÍM ILLESZTÉS

- ◉ A routing tábla sorait prefix hossz szerint csökkenő sorrendbe rendezzük.  $N=1$ .
- ◉ Ezzel biztosítjuk, hogy több illeszkedő sor esetén a leghosszabb prefixűt fogjuk eredményként kapni.
- ◉ Ha nem létezik a táblázatban az  $N$ . sor, akkor nincs illeszkedő sor, és vége.
- ◉ A csomag célcíme és az  $N$ . sor hálózati maszkja között bitenkénti AND műveletet hajtunk végre.
- ◉ Ha a bitenkénti AND művelet eredménye megegyezik az  $N$ . sor célhálózat értékével, akkor a cím az  $N$ . sorra illeszkedik és vége.
- ◉  $N=N+1$ ; folytassuk a 2. pontnál.



# IP ALHÁLÓZATOK 1

- ◉ Az intézmények logikai működésük, vagy térbeli elhelyezkedésük alapján kisebb (azonos méretű) részekre oszthatják a hálózati címtartományukat. A felosztás eredményeként kisebb, könnyebben kezelhető üzenetszórési tartományokat tudunk kialakítani.
- ◉ **Alhálózatok kialakítása (subnetting):** Az IP cím host részének legmagasabb helyiértékű bitjeiből néhányat az alhálózat (subnet) azonosítására használunk. Az új hálózat-csomópont azonosító határvonal pozícióját a hálózati maszk megadásával jelöljük.
- ◉ Alhálózat kialakítási példa:

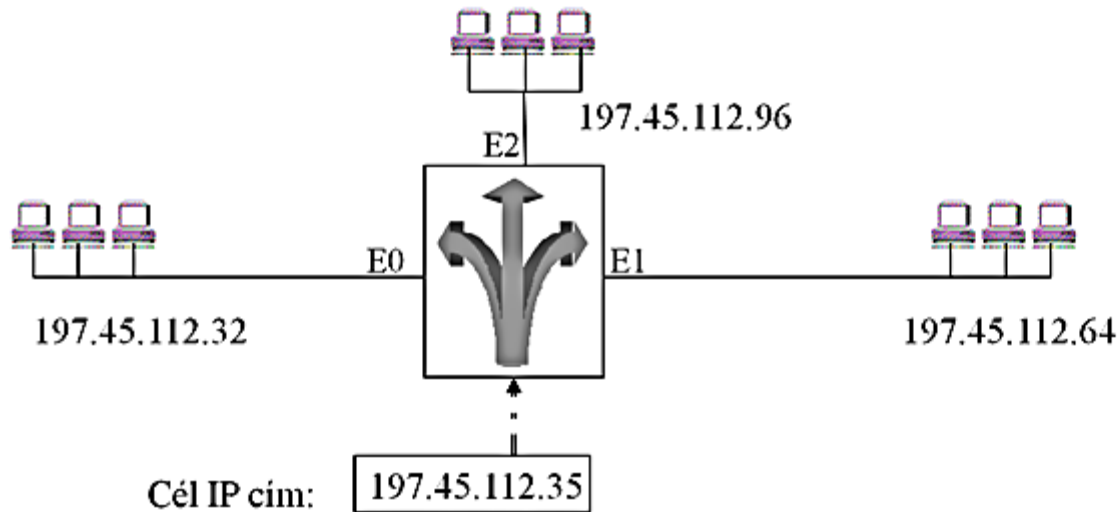
# IP ALHÁLÓZATOK 1

- Hálózat IP címe: 197.45.112.0
- Alapértelmezett hálózati maszk: 255.255.255.0
- Használjunk 3 bitet alhálózat azonosításra.
- Hálózati maszk: 255.255.255.224
- Összesen 8 alhálózat kialakítására van lehetőség.

| Sorszám | Alhálózat címe | Alhálózati gépcímek |
|---------|----------------|---------------------|
| 1.      | 197.45.112.0   | 197.45.112.1-30     |
| 2.      | 197.45.112.32  | 197.45.112.33-62    |
| 3.      | 197.45.112.64  | 197.45.112.65-94    |
| 4.      | 197.45.112.96  | 197.45.112.97-126   |
| 5.      | 197.45.112.128 | 197.45.112.129-158  |
| 6.      | 197.45.112.160 | 197.45.112.161-190  |
| 7.      | 197.45.112.192 | 197.45.112.193-222  |
| 8.      | 197.45.112.224 | 197.45.112.225-254  |

# FORGALOMIRÁNYÍTÁS ALHÁLÓZATOK KÖZÖTT

11

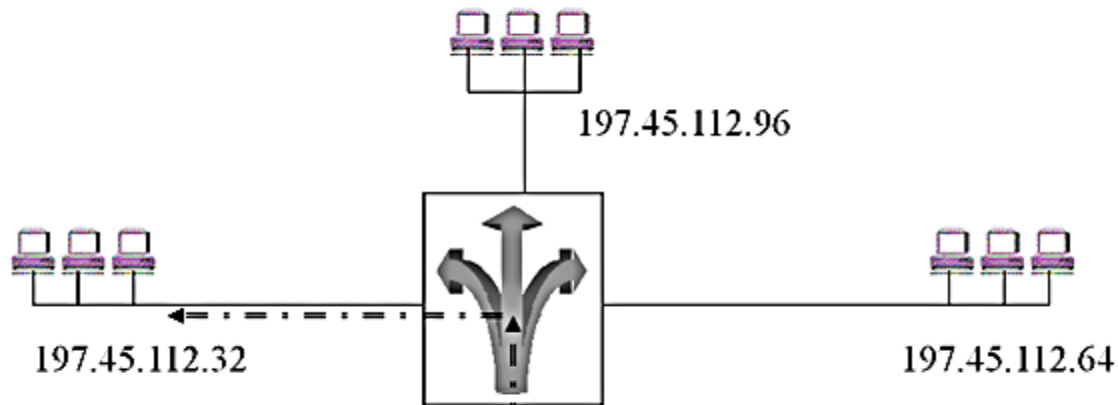


**Forgalomirányítási tábla:**

| Cél           | Netmask         | Interfész | Next-hop | Metrika |
|---------------|-----------------|-----------|----------|---------|
| 197.45.112.32 | 255.255.255.224 | E0        | 0.0.0.0  | 0       |
| 197.45.112.64 | 255.255.255.224 | E1        | 0.0.0.0  | 0       |
| 197.45.112.96 | 255.255.255.224 | E2        | 0.0.0.0  | 0       |

# FORGALOMIRÁNYÍTÁS ALHÁLÓZATOK KÖZÖTT

| Cél           | Netmask         | Interfész | Next-hop | Metrika |
|---------------|-----------------|-----------|----------|---------|
| 197.45.112.32 | 255.255.255.224 | E0        | 0.0.0.0  | 0       |
| 197.45.112.64 | 255.255.255.224 | E1        | 0.0.0.0  | 0       |
| 197.45.112.96 | 255.255.255.224 | E2        | 0.0.0.0  | 0       |



Cél IP cím: 197.45.112.35  
Hálózati maszk: 255.255.255.224

→ & → 197.45.112.32

# Útválasztás

Dinamikus útválasztás

Statikus  
útválasztás

IGP  
(Interior Gateway Protocol)

EGP  
(Exterior  
Gateway  
Protocol)

Távolságvektor alapú  
forgalomirányítás (Routing  
Vector Protocols)

Kapcsolat-állapot (link-  
állapot) alapú  
forgalomirányítás (Link State  
Routing)

Hibrid

Path vector

RIP

IGRP

OSPF

ISIS

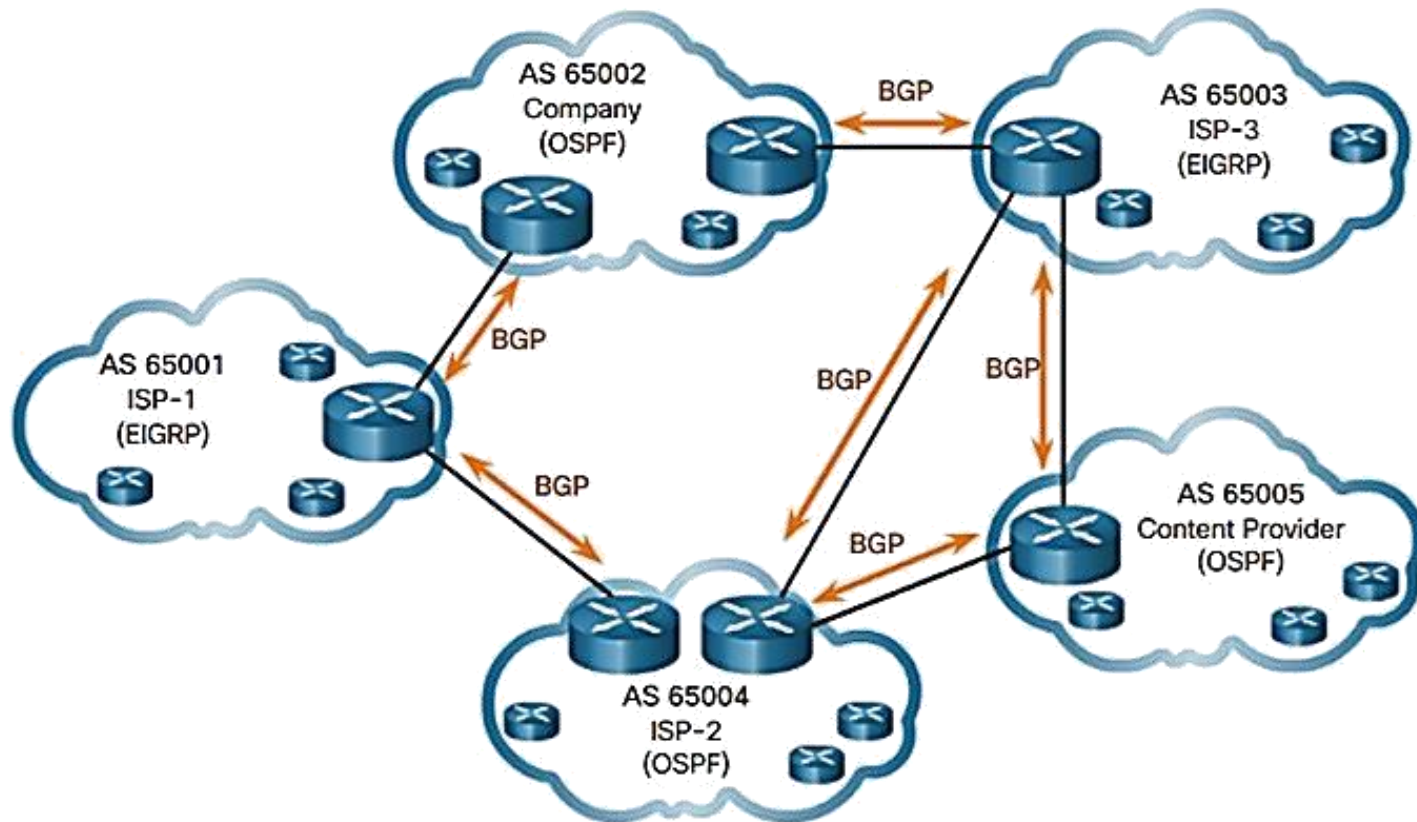
EIGRP

BGP

# FORGALOMIRÁNYÍTÁSI KONFIGURÁCIÓK OSZTÁLYOZÁSA

- ◉ **Minimális routing:** Teljesen izolált (router nélküli) hálózati konfiguráció. Forgalomirányítási döntés nem csak a forgalomirányítókön történik, hanem minden csomóponton a csomag küldése előtt.
- ◉ **Statikus routing:** A forgalomirányítási táblázatot a rendszeradminisztrátor tartja karban. Tipikus példa a végfelhasználói csomópontokon az alapértelmezett útválasztó (default router, vagy más néven default gateway) beállítása.
- ◉ **Dinamikus routing:** A forgalomirányítási táblázat(ok) valamilyen routing protocol segítségével kerülnek karbantartásra.
  - Belső forgalomirányítási protokollok (IGP). Egy autonóm rendszeren belül működik, legfőbb alapelv a
    - „legjobb útvonal” meghatározása ún. távolságvektor alapú vagy
    - kapcsolat-állapot (link-állapot) alapú módszerrel
  - Külső forgalomirányítási protokollok (EGP). A két autonóm rendszer számára egy autonóm rendszertől a másikig működik, és fordítva.
- ◉ Az **autonóm rendszer** olyan logikai határ, amely egyetlen közös adminisztráció alatt működik.

# IGP ÉS EGP





Digital Attack Map Top daily DDoS attacks worldwide

April 17 2021

Showing All Countries Show Attacks

Large attacks on United States, Brazil, and 2 others

Color Attacks By

Type Source Port

Duration Dest. Port

● TCP Connection  
● Volumetric  
● Fragmentation  
● Application

Size (Bandwidth, in Gbps)

25 5 1

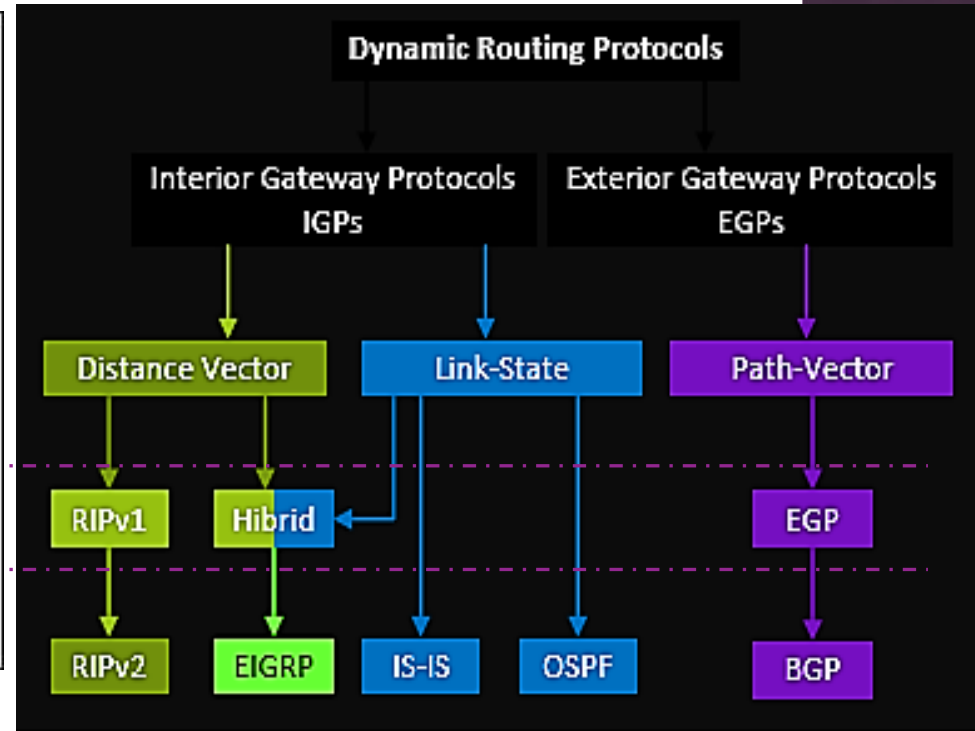
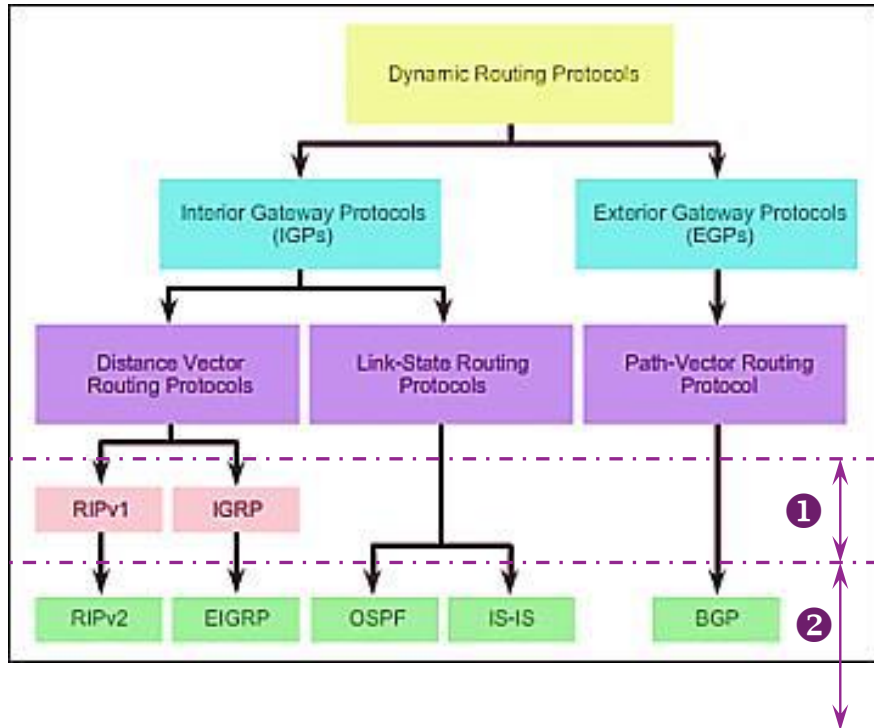
Shape (source + destination)

between two countries

internal

# ROUTING

① - Classfull ② - Classless



# ROUTING

## ◉ Távolság vektor

- A távolsági vektor útvonalválasztó protokoll a távoli hálózathoz vezető legjobb utat találja a relatív távolság használatával.
- Minden alkalommal, amikor egy csomag áthalad egy útvásztón, ugrásnak nevezzük.
- A legjobb útvonal az az útvonal, amely a legkisebb számú komlóval rendelkezik a hálózathoz.

## ◉ Link State

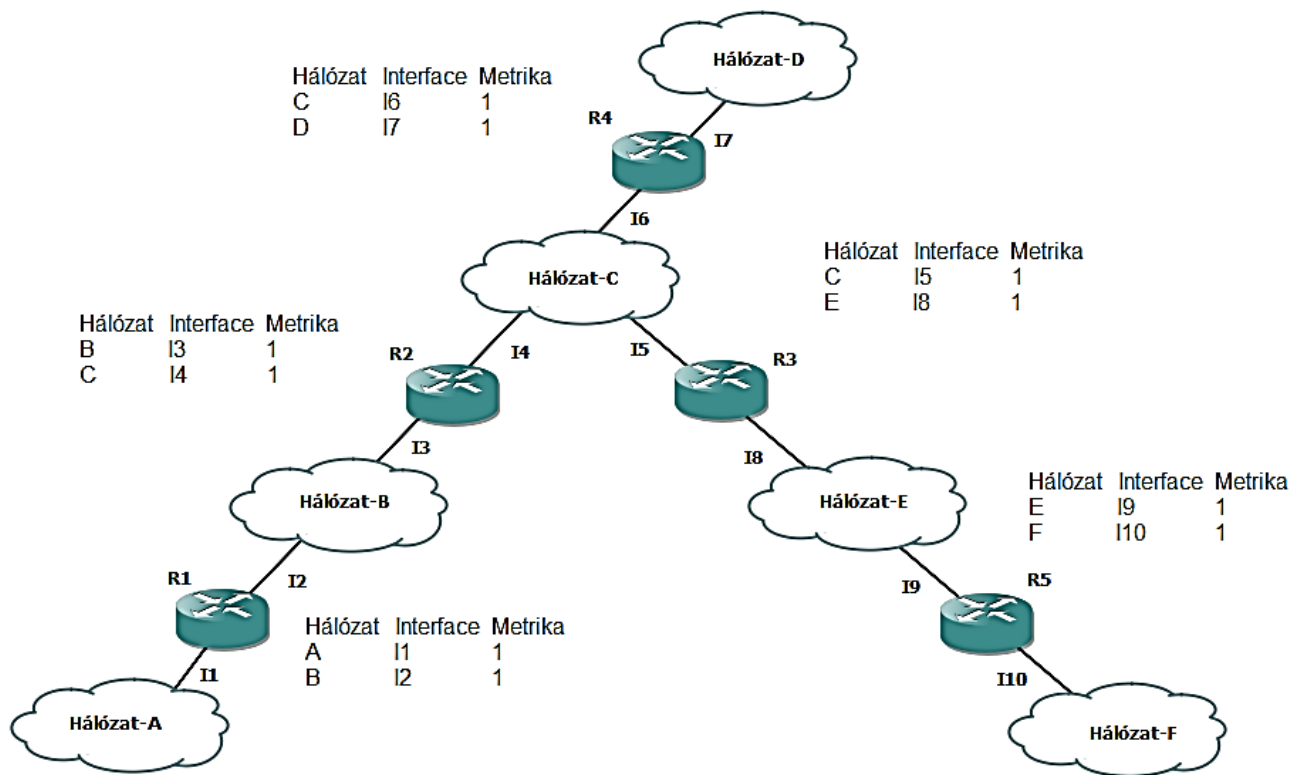
- A legrövidebb út is az első, ahol minden útvásztó három külön táblát hoz létre. Minden táblázat elvégzi a különböző funkciókat, mint például a közvetlenül csatlakozó szomszédok nyomon követését, a második pedig a teljes internetes munka topológiáját, a harmadik pedig az útvásztási táblázatot.

## ◉ Hibrid

- A távolságvektorra és a kapcsolatállapotra, például az EIGRP-re jellemző.

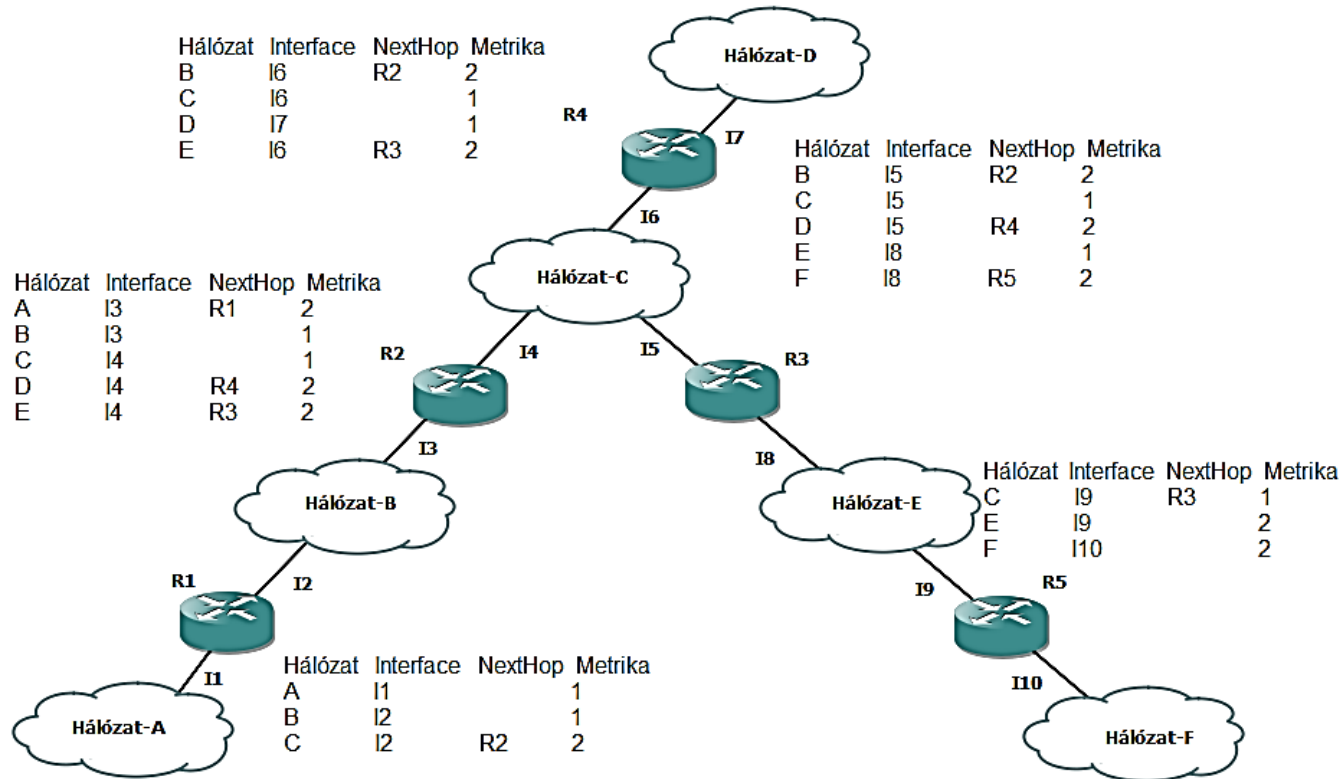
# ROUTING VECTOR PROTOCOLS (RVP) - BELLMAN-FORD ALGORITHM.

19



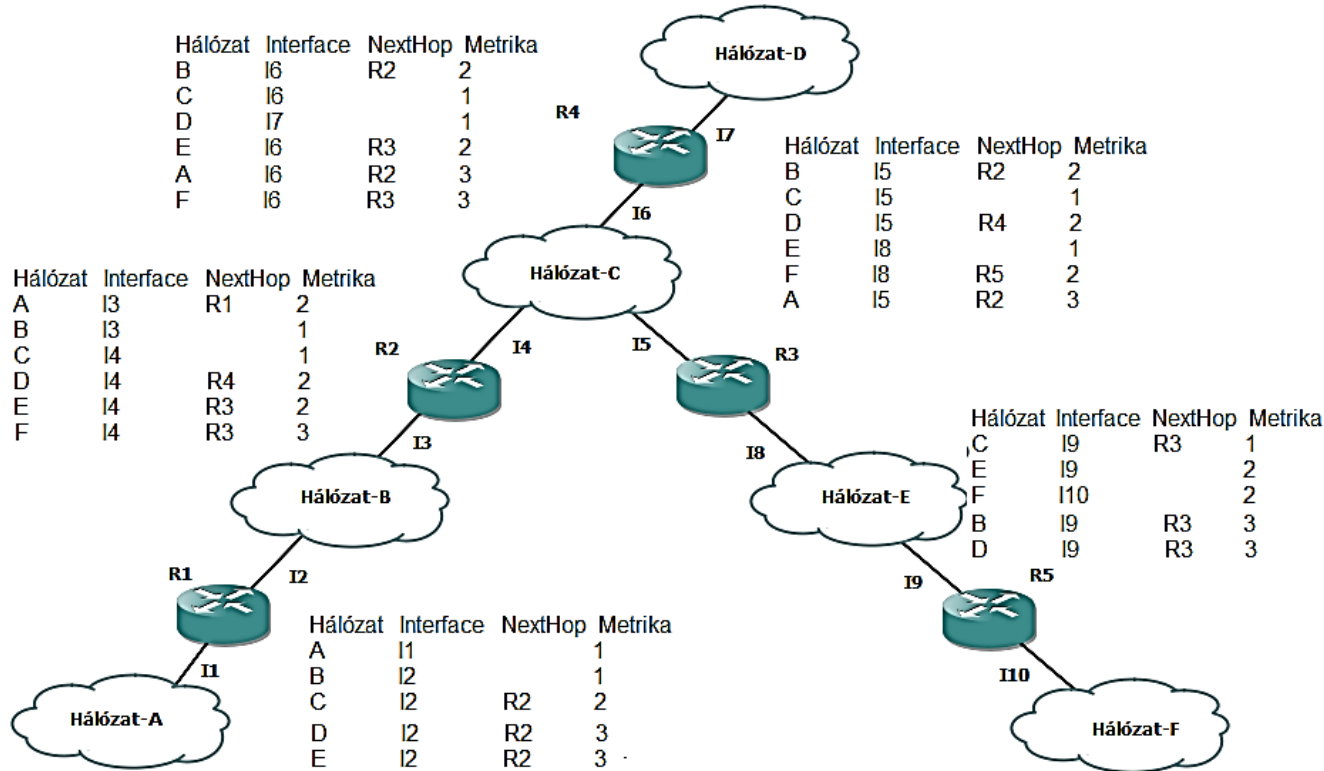
# ROUTING VECTOR PROTOCOLS (RVP) - BELLMAN-FORD ALGORITHM.

20



# ROUTING VECTOR PROTOCOLS (RVP) - BELLMAN-FORD ALGORITHM.

21

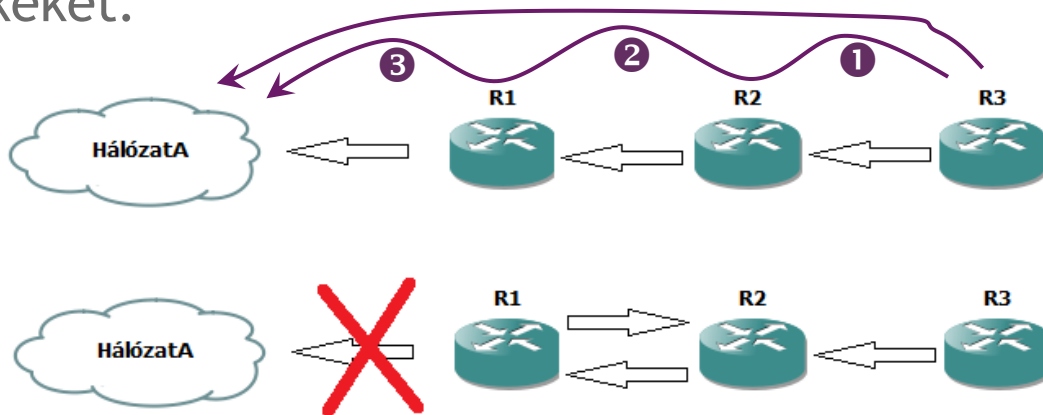


# A TÁVOLSÁGVEKTOR ALGORITMUS ALAPPROBLÉMÁJA

22

## ◉ A végtelenig számolás

- egyre növelik egy adott csomóponthoz való távolságukat, elvileg végtelenig, konkrét megvalósításoknál véges számig (15 - OK, 16 - nem elérhető)
- Következmény: egymásnak irányítják az el nem érhető csomóponthoz tartó forgalmat, ezzel túlterhelnek linkeket.

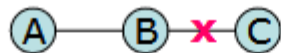




# A TÁVOLSÁGVEKTOR ALGORITMUS ALAPPROBLÉMÁJA

23

- Példa a végtelenig számolásra:



- A B-C link megszakad
- B kijavítja a bejegyzését
- B és A kicserélik elképzeléseiket és korrigálnak:
- Ha ismét cserélnek és korrigálnak:

|   | C táv    | Köv. lépés |
|---|----------|------------|
| A | 2        | B          |
| B | 1        | C          |
| A | 2        | B          |
| B | $\infty$ | -          |

|   |          |   |
|---|----------|---|
| A | $\infty$ | - |
| B | 3        | A |

|   |          |   |
|---|----------|---|
| A | 4        | B |
| B | $\infty$ | - |

# A TÁVOLSÁGVEKTOR MÓDSZER JAVÍTÁSI LEHETŐSÉGEI

## ◉ Split horizon:

- megtiltja a csomópont számára az útvonal hirdetését azon interfész felé, ahonnan tanulta

## ◉ Split horizon with poison reverse:

- amennyiben a csomópont egy másik csomóponton keresztül ér el egy harmadikat, akkor ennek a másiknak azt hirdeti, hogy tőle végtelen távolságra van a harmadik.

## ◉ Holddown timer:

- router elindít egy időzítőt, amikor olyan értesítést kap, hogy egy csomópont elérhetetlen. Amíg le nem jár, eldob minden értesítést az ebbe a csp-ba vezetőútvonalakról.

# RIP

- ◉ Az útválasztási információs protokoll a távolsági vektor útvonalválasztás egyenes megvalósítása a helyi hálózatok számára.
- ◉ 30 másodpercenként a teljes útválasztási táblát minden aktív interfészre szállítja.
- ◉ Távolságvektor alapú IGP protokoll. Régi, de folyamatosan fejlesztik, javítják.
- ◉ A **Hop szám** az egyetlen metrika, amely leírja a távoli hálózat legjobb útját, **de maximum 15 lehet** (16 = végtelen távolság).
- ◉ Megakadályozza az útvonali hurkok létrehozását azáltal, hogy korlátozza az útvonalon engedélyezett komlószerkezetek számát.

# RIP, RIPNG - IPV6

## Jellemzők

Osztálytámogatás

RIPv1

C csoportnyi

RIPv2

Osztály nélküli

Támogatja a változó hosszúságú alhálózati maszkot (VLSM)

Nem

Igen

Elküldi az alhálózati maszkot az útválasztási frissítéssel együtt

Nem

Igen

Kommunikál más RIP útválasztóval az alábbi címtípuson keresztül

Adás

multicast

RFC meghatározás

RFC 1058

1721, 1722 és 2453 RFC-k

Támogatja a hitelesítést

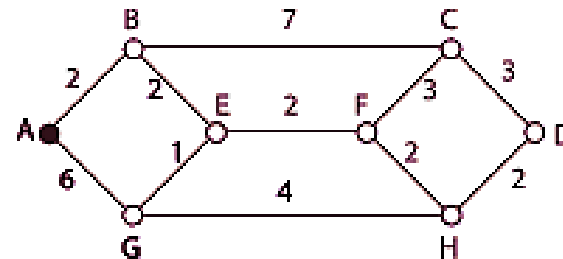
Nem

Igen

# LINK STATE PROTOCOLS (LSP) - DIJKSTRA ALGORITMUS

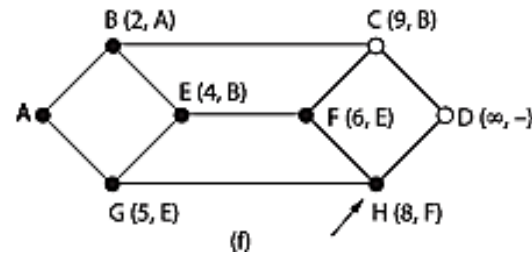
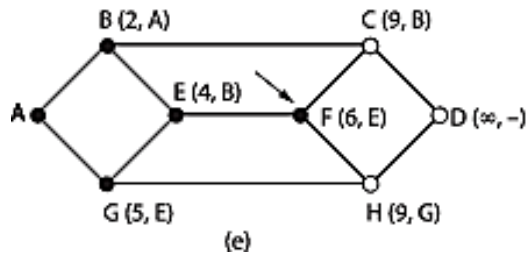
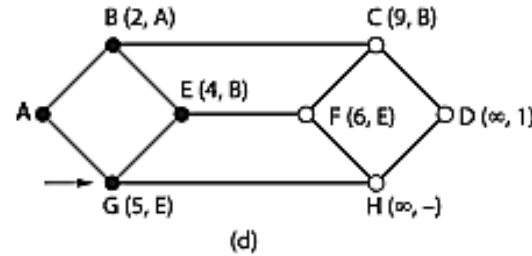
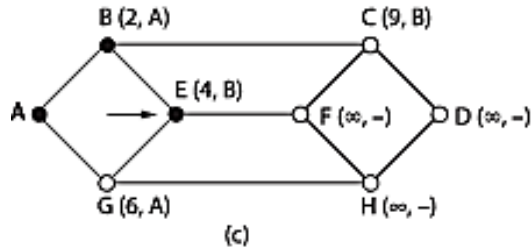
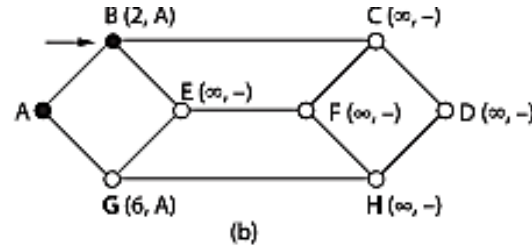
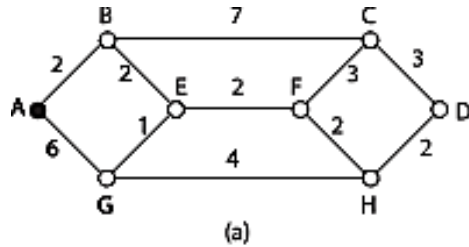
27

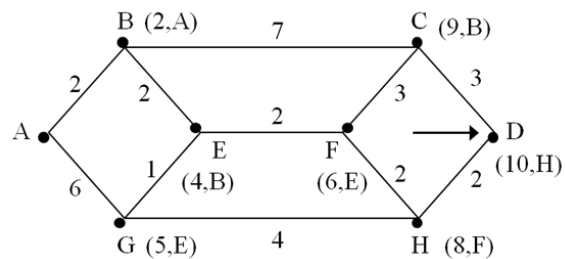
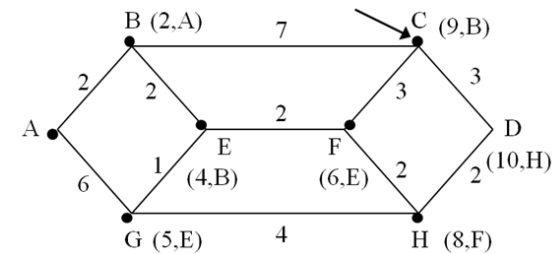
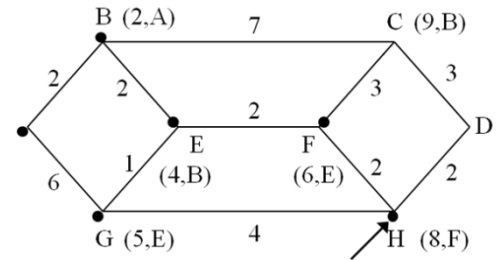
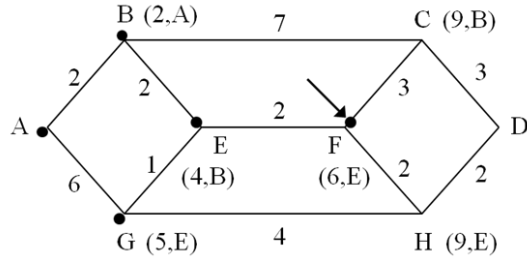
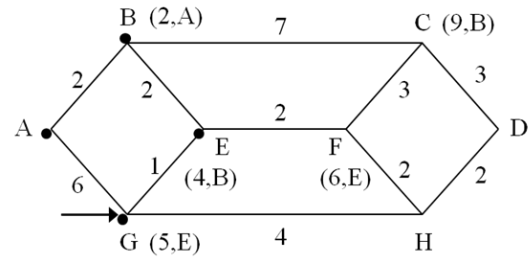
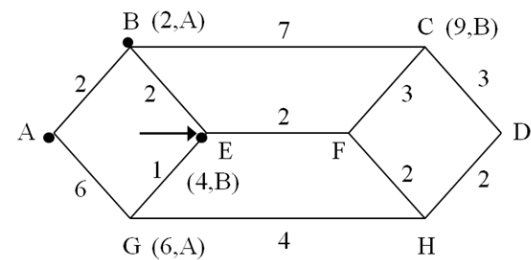
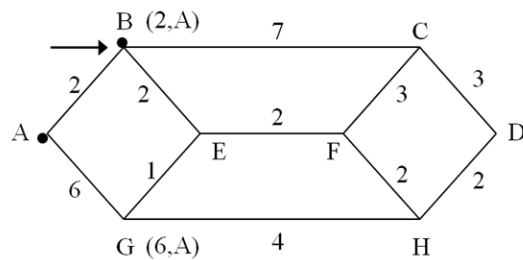
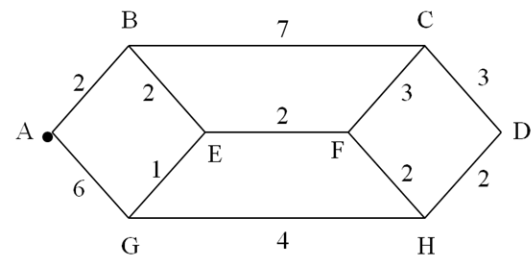
- ◉ Minden útválasztó egy csomópontnak, és minden él egy kommunikációs vonalnak (vagy adatkapcsolatnak) felel meg egy zárt területen belül.
  - Két adott útválasztó közötti útvonal kiválasztásához az algoritmus egyszerűen a köztük levő legrövidebb utat keresi meg a gráfban.
  - Egy csomópont lepingeli a szomszédjait és ezeket az utakat értékeli pl. metrika alapján. Router ezeket az információkat szétküldi az összes routernek. Így az összes csomópontnak megvannak ezek az adatok a többi csomópontból. Minden router onálón megkeresi a legrövidebb utat az egyes hálózatokba és eredményt bevisszi az útválasztási táblájába.
  - Egy út hosszát mérhetjük például a megtett ugrások számával.
    - ◉ *ABC* és *ABE* út ugyanolyan hosszú. Egy másik mérték lehet a földrajzi távolság kilométerekben, amikor is *ABC* nyilvánvalóan sokkal hosszabb, mint *ABE* (feltéve, hogy az ábra léptékhelyes).



# LINK STATE PROTOCOLS (LSP) - DIJKSTRA ALGORITHMUS

28





Az "A"-ból a "D"-be vezető optimális utat keressük. A ponttal jelzett gráfcúspont az adott gráfcúspont "lezárt" állapotát jelzi. Ez azt jelenti, hogy a gráfcúspontba vezető legjobb utat már megtaláltuk, s megvizsgáltuk, hogy az adott gráfcúspontból merre lehet továbblépni optimális úton haladva. A nyíllal jelölt gráfcúspont az ún. "aktuális" pont, erre vonatkozóan vizsgáljuk, hogy milyen (nem zárt) továbblépési lehetőségek vannak innen.



# ALGORITMUS KOMPLEXITÁSA

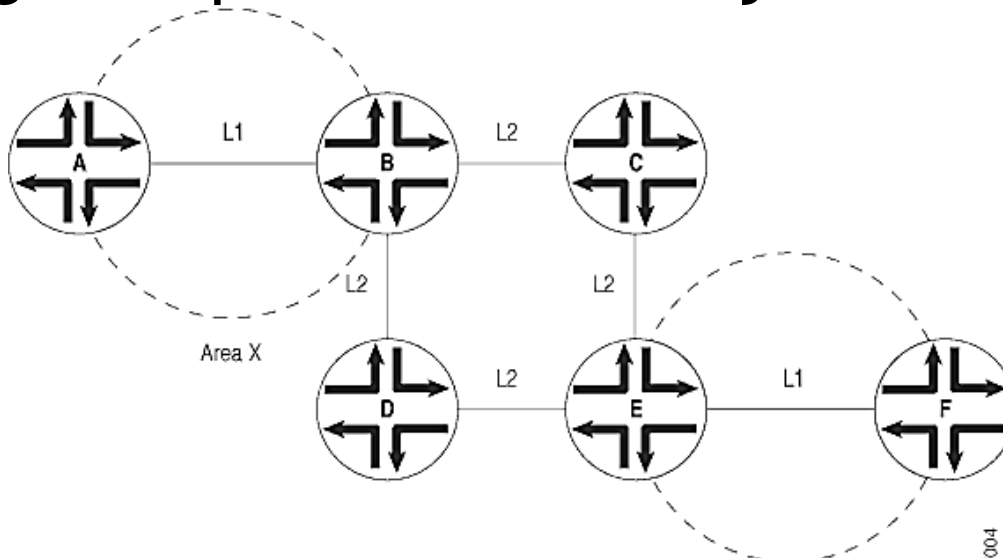
- ◉ Az információ elküldése mindenkinek elárasztással van megvalósítva. Minden csomópont rendelkezik a globális link-állapot információkkal az elárasztásnak köszönhetően.
  - A linkállapot-információk alapján a legrövidebb utak kiszámíthatók.
  - $k$  iteráció után, csomópont ismeri a legrövidebb utat  $k$  célállomás felé.
- ◉ Az algoritmus komplexitása:  $n$  csomópont esetén
  - Minden iterációban: minden csp-ra megnézni, melyik a legkisebb költséggel elérhető, ami nincs benne  $N'$  halmazban
  - $n(n+1)/2$  összehasonlítás:  $O(n^2)$
  - Hatékonyabb implementációval:  $O(n \log n)$
  - Gyakorlatban: hierarchikus linkállapot routing protokollok

$N'$ : azon csp-ok halmaza, amelyekhez megvan a legkisebb költségű útvonal

# IS-IS INTERMEDIATE SYSTEM TO INTERMEDIATE SYSTEM

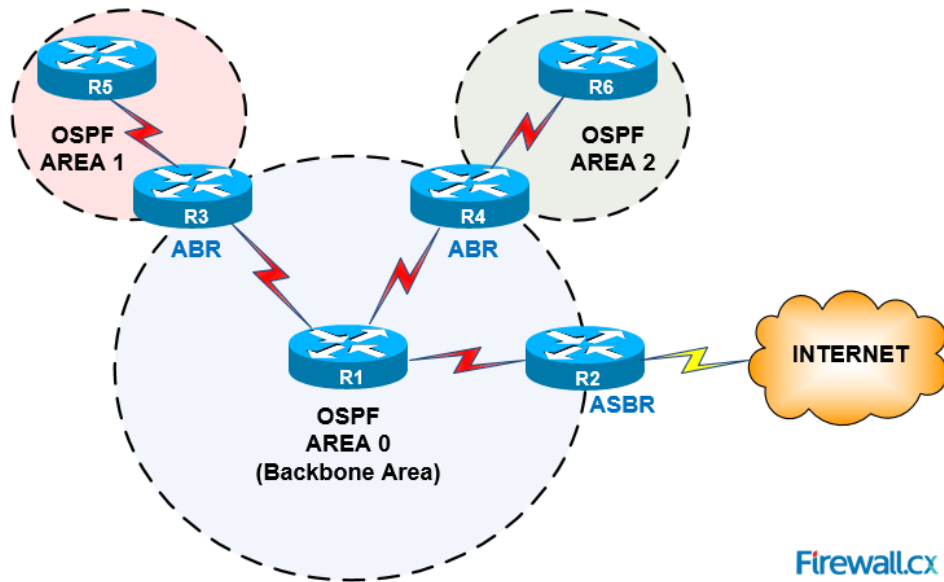
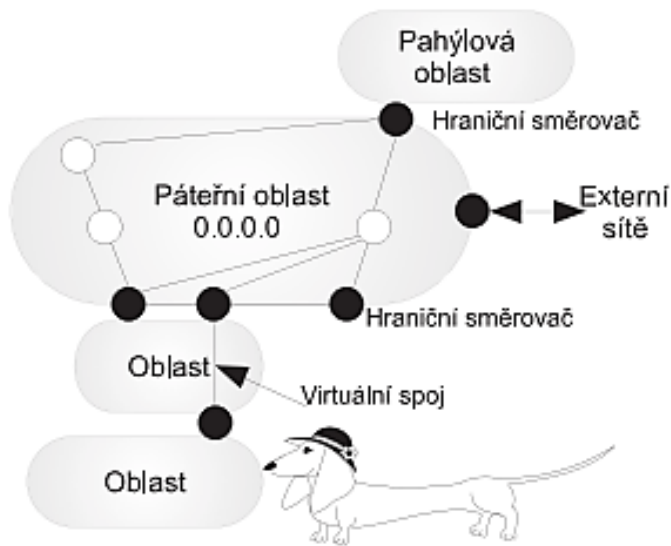
31

- ◉ L1 és L2 - adminisztrátor feladata felosztani a teret és definiálni a L2 routereket.
- ◉ Eléggé komplikát és nem terjedt el.



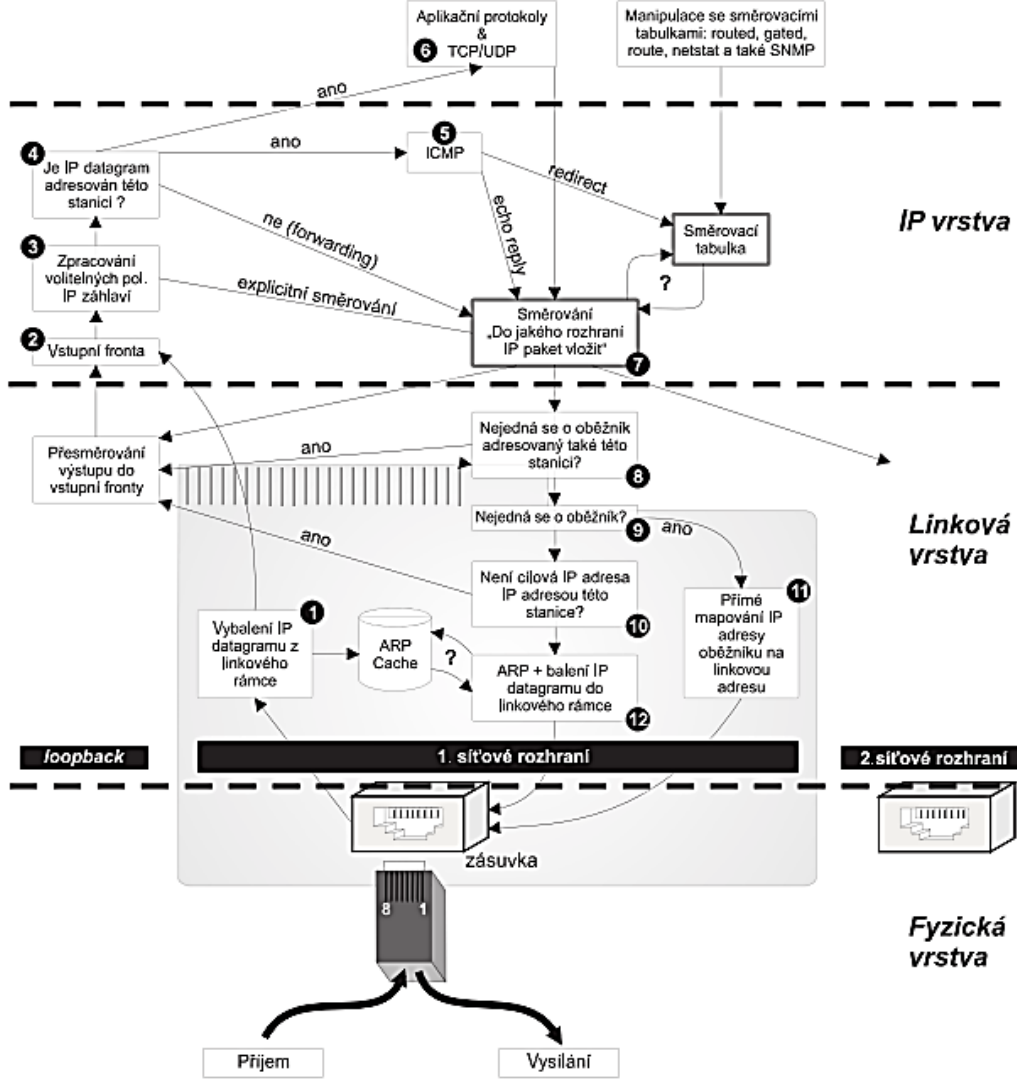
# OSPF - OPEN SHORTEST PATH FIRST

- Már adott a topológia - adminisztrátor feladata beálítani

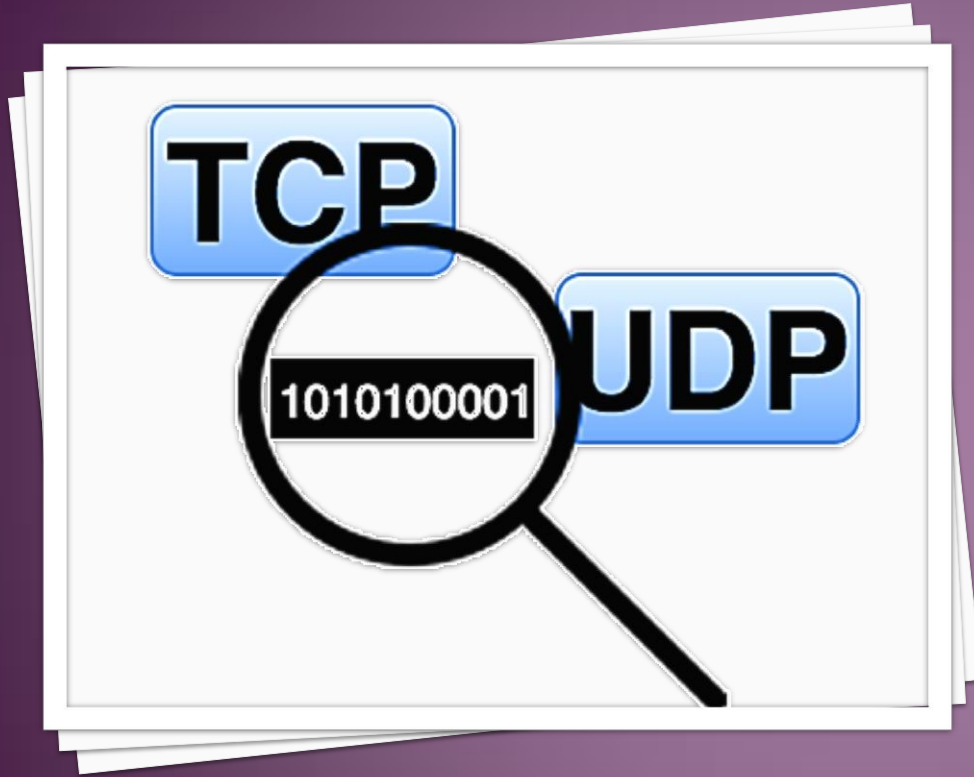




## FORWARDING VS. ROUTING



- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12



TCP UDP

# TCP/UDP

- ◉ A szállítási (transzport) réteg funkcionalitása a bitfolyam-átvitel biztosítása az applikációk számára.
- ◉ Feladatai közé tartozik az átvitel vezérlése ill. szabályozása (pl. torlódás elkerülés, torlódási helyzetek kezelése).
- ◉ Két protokollt vizsgálunk: az (egyszerűbb) UDP és az (összetettebb) TCP protokollt.

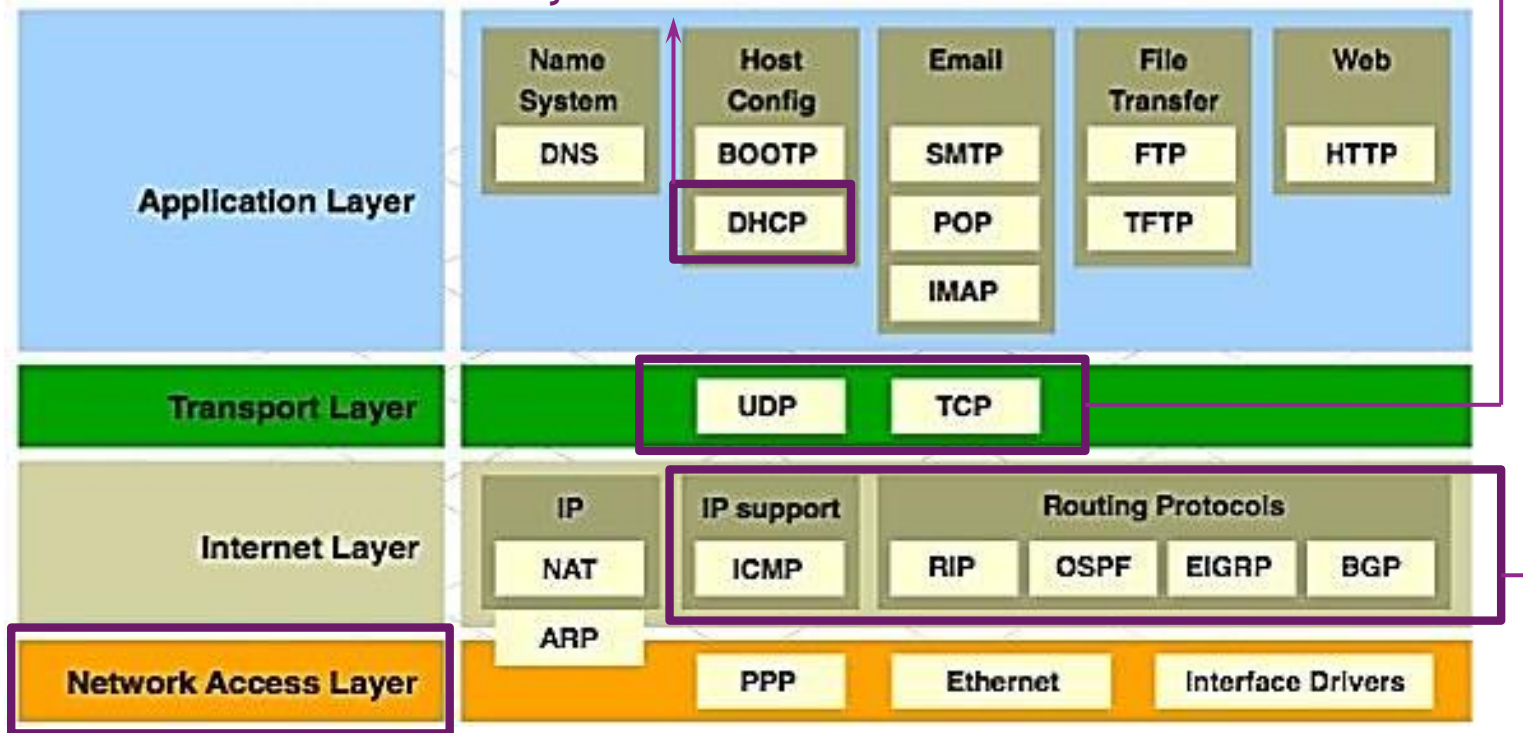


# TCP / UDP

Gyakorlat: cmd / PowerShell, TCP/IP csomagok struktúrája

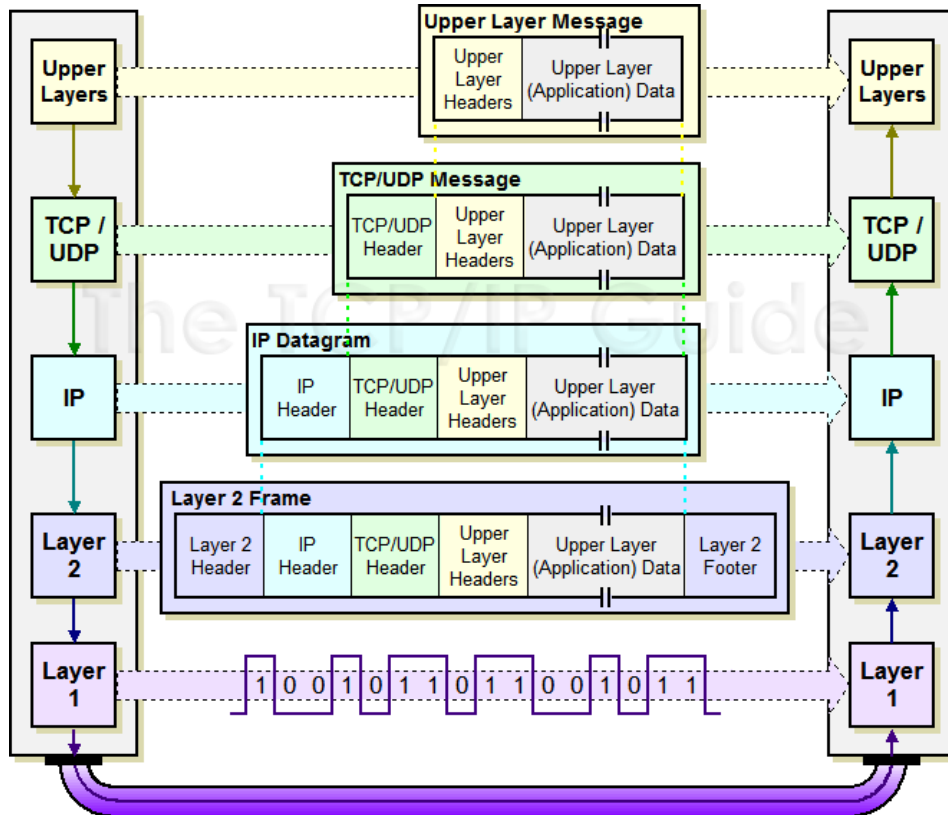
Gyakorlat: Útválasztás, OSPF, RIP, EIGRP, BGP

Gyakorlat: DHCP



Gyakorlat: Modulációk, vonali kódolások vs. ethernet

# TCP / UDP

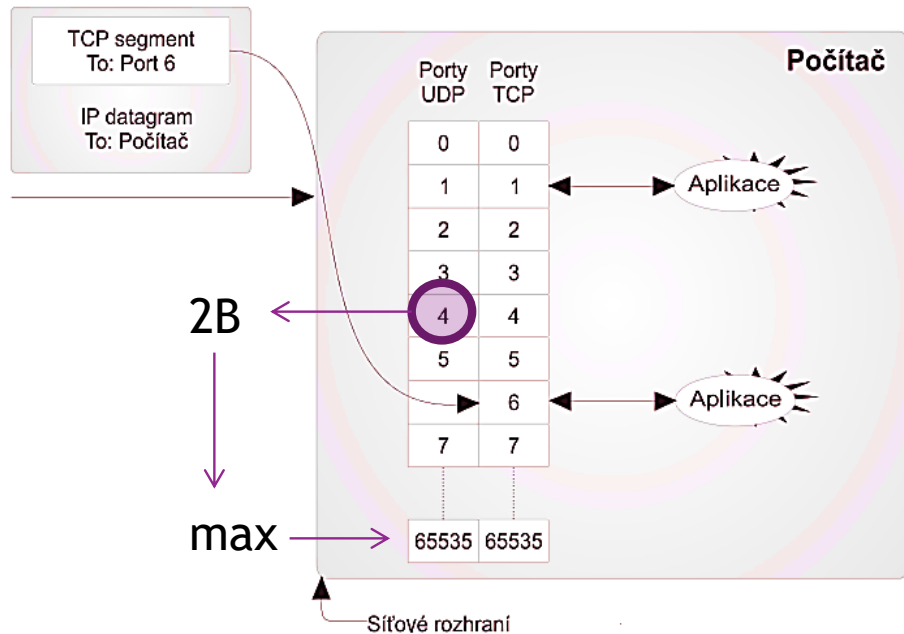


Mivel van identifikálva az adatfolyam?

1. IP source
2. IP destination
3. Port source
4. Port destination
5. Protocol TCP or UDP

Kliens - Szerver

# PORTOK



21: FTP (File Transfer Protocol) TCP  
 25: SMTP (Simple Mail Transfer Protocol) TCP  
 53: DNS (Domain Name System) TCP UDP  
 80: HTTP (HyperText Transfer Protocol) TCP  
 443: HTTPS (HyperText Transfer Protocol over SSL) TCP  
 ...



C:\Windows\System32\drivers\etc\services  
 Vagy IANA:

<https://www.iana.org/assignments/service-names-port-numbers/>

# UDP - TCP

- ◉ Az UDP protokoll (RFC 768) egyszerű, nem megbízható (nem nyugtázott), összeköttetés mentes szállítási réteg protokoll.
- ◉ Az TCP protokoll (RFC 793) megbízható (nyugtázott), összeköttetés alapú szállítási réteg protokoll. Az adatkommunikáció megkezdése előtt kapcsolat (TCP összeköttetés) épül ki a felek között. A kommunikáció során pedig folyamatos áramlásszabályozást lát el.

## Pseudoheader

| Offsets | Octet | 0           |   |   |   |   |   |   |   | 1 |   |    |    |    |    |    |                  | 2  |    |    |    |    |    |    |    | 3  |    |    |    |    |    |    |    |
|---------|-------|-------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Octet   | Bit   | 0           | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15               | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 0       | 0     | Source port |   |   |   |   |   |   |   |   |   |    |    |    |    |    | Destination port |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 4       | 32    | Length      |   |   |   |   |   |   |   |   |   |    |    |    |    |    | Checksum         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

| Offsets | Octet | 0   |   |   |   |                   |   |   |        | 1           |             |             |             |             |             |             |             | 2                           |   |   |   |   |   |   |   | 3 |   |   |   |   |   |   |   |
|---------|-------|---|---|---|---|-------------------|---|---|--------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Octet   | Bit   | 7   | 6 | 5 | 4 | 3                 | 2 | 1 | 0      | 7           | 6           | 5           | 4           | 3           | 2           | 1           | 0           | 7                           | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0       | 0     | Source port   |   |   |   |                   |   |   |        |             |             |             |             |             |             |             |             | Destination port            |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 4       | 32    | Sequence number   |   |   |   |                   |   |   |        |             |             |             |             |             |             |             |             |                             |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 8       | 64    | Acknowledgment number (if ACK set)  |   |   |   |                   |   |   |        |             |             |             |             |             |             |             |             |                             |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 12      | 96    | Data offset   |   |   |   | Reserved<br>0 0 0 |   |   | N<br>S | C<br>W<br>R | E<br>C<br>E | U<br>R<br>G | A<br>C<br>K | P<br>S<br>H | R<br>S<br>T | S<br>Y<br>N | F<br>I<br>N | Window Size                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 16      | 128   | Checksum  |   |   |   |                   |   |   |        |             |             |             |             |             |             |             |             | Urgent pointer (if URG set) |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 20      | 160   | Options (if <i>data offset</i> > 5. Padded at the end with "0" bytes if necessary.) |   |   |   |                   |   |   |        |             |             |             |             |             |             |             |             |                             |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| :       | :     |   |   |   |   |                   |   |   |        |             |             |             |             |             |             |             |             |                             |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 60      | 480   |   |   |   |   |                   |   |   |        |             |             |             |             |             |             |             |             |                             |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

# UDP - TCP

|    |     |  |                   |        |             |             |             |             |             |             |             |             |                             |
|----|-----|--|-------------------|--------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------------------------|
| 8  | 64  | Acknowledgment number (if ACK set)   |                   |        |             |             |             |             |             |             |             |             |                             |
| 12 | 96  | Data offset  | Reserved<br>0 0 0 | N<br>S | C<br>W<br>R | E<br>C<br>E | U<br>R<br>G | A<br>C<br>K | P<br>S<br>H | R<br>S<br>T | S<br>Y<br>N | F<br>I<br>N | Window Size                 |
| 16 | 128 | Checksum   |                   |        |             |             |             |             |             |             |             |             | Urgent pointer (if URG set) |
| 20 | 160 | Options (if data offset > 5. Padded at the end with "0" bytes if necessary.) |                   |        |             |             |             |             |             |             |             |             |                             |
| :  | :   |  |                   |        |             |             |             |             |             |             |             |             |                             |
| 60 | 480 |  |                   |        |             |             |             |             |             |             |             |             |                             |

Legfontosabb jelzőbitek:

- ◉ SYN: Kapcsolat kiépítés (szinkronizáció)
- ◉ FIN: Kapcsolat bontás (finish)
- ◉ ACK: Érvényes a nyugta sorszám mező értéke
- ◉ 16 bit Ablakméret - (A következőként várt szegmens maximális mérete)
- ◉ A szegmens hossza nincs benne a fejrészben, ez a TCP/IP interfészen adódik át az IP felé. A TCP minden bájtot sorszámoz, s a nyugtában jelzi vissza a társa felé a soron következőként várt bájt sorszámát.
- ◉ A TCP áramlásszabályozást is végez: az ablakméret mezőben jelzi vissza a társának, hogy az maximum milyen hosszú szegmenst küldhet legközelebb.

# TCP HÁROMUTAS KÉZFOGÁS

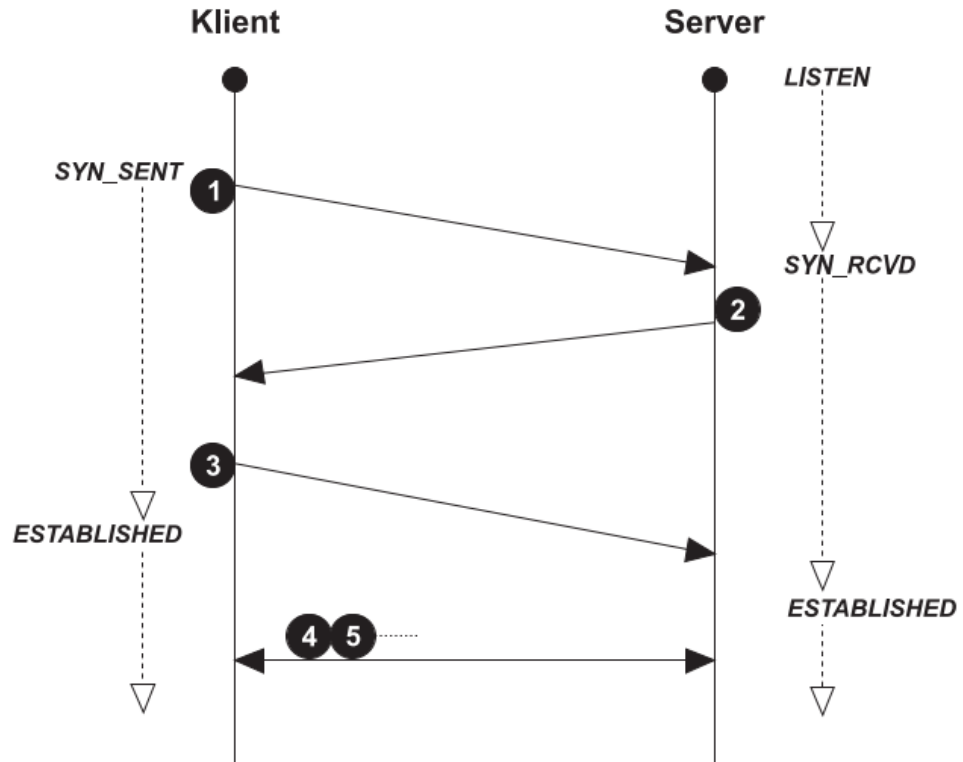
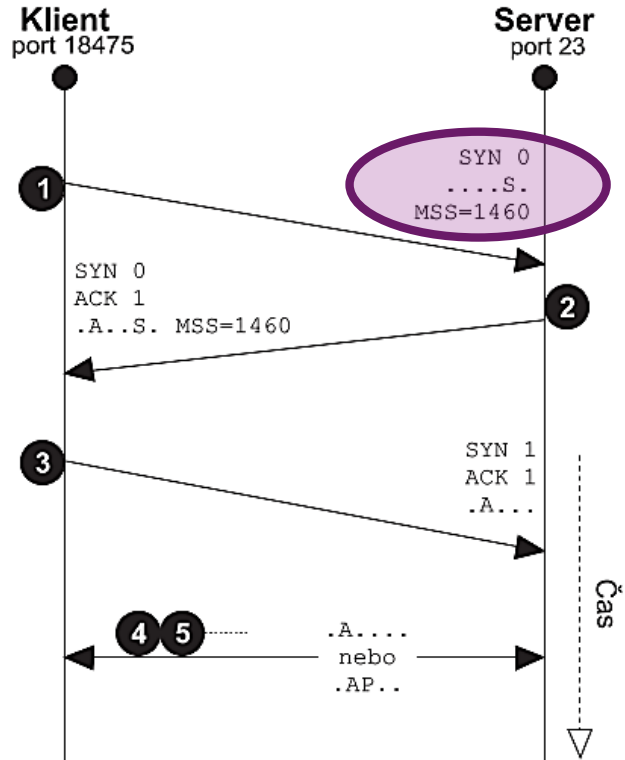
- A TCP az applikációk közötti adatátvitel megkezdése előtt egy TCP összeköttetést (TCP kapcsolatot) épít ki, ezt a kapcsolatkiépítést hívjuk TCP háromutas kézfogásnak (3 way handshake).
  - A kapcsolat kiépítést a kliens kezdeményezi.
    - A TCP fejrészben a portszámok megfelelően beállításra kerülnek;
    - a kezdősorszám egy (bizonyos feltételeknek eleget tevő) véletlen-szám lesz (pl. SEQ\_No=450).
    - A jelzőbiteknél SYN=1, ACK=0.
  - A szerver megkapja a kliens üzenetét. A TCP fejrészből (jelzőbitekből) látja, hogy új kapcsolat kiépítése indult. A szerver jóváhagyó válasz-üzenetet küld:
    - A válasz TCP fejrészben a kapott üzenet portszámait felcseréli;
    - beállítja a saját (véletlen) kezdősorszámát (pl. SEQ\_No=870),
    - a nyugta sorszámot a kapott SEQ érték rákövetkezőjére (ACK\_No=451) állítja.
    - A jelzőbiteknél SYN=1, ACK=1.
  - A kliens megkapja a szerver választ, s erre egy jóváhagyást küld a szerver felé.
    - A TCP fejrészben a kapott üzenet portszámait felcseréli;
    - beállítja a saját szegmes-sorszámát (SEQ\_No=451);
    - a nyugta sorszámot pedig a kapott SEQ érték rákövetkezőjére (ACK\_No=871) állítja.
    - A jelzőbiteknél SYN=0, ACK=1.
- A szerver megkapja a kliens választ, s ezzel a kapcsolat kiépült. Ezután megindul az applikációs rétegben használt protokollnak megfelelő adatátviteli kommunikáció.

# TCP HÁROMUTAS KÉZFOGÁS

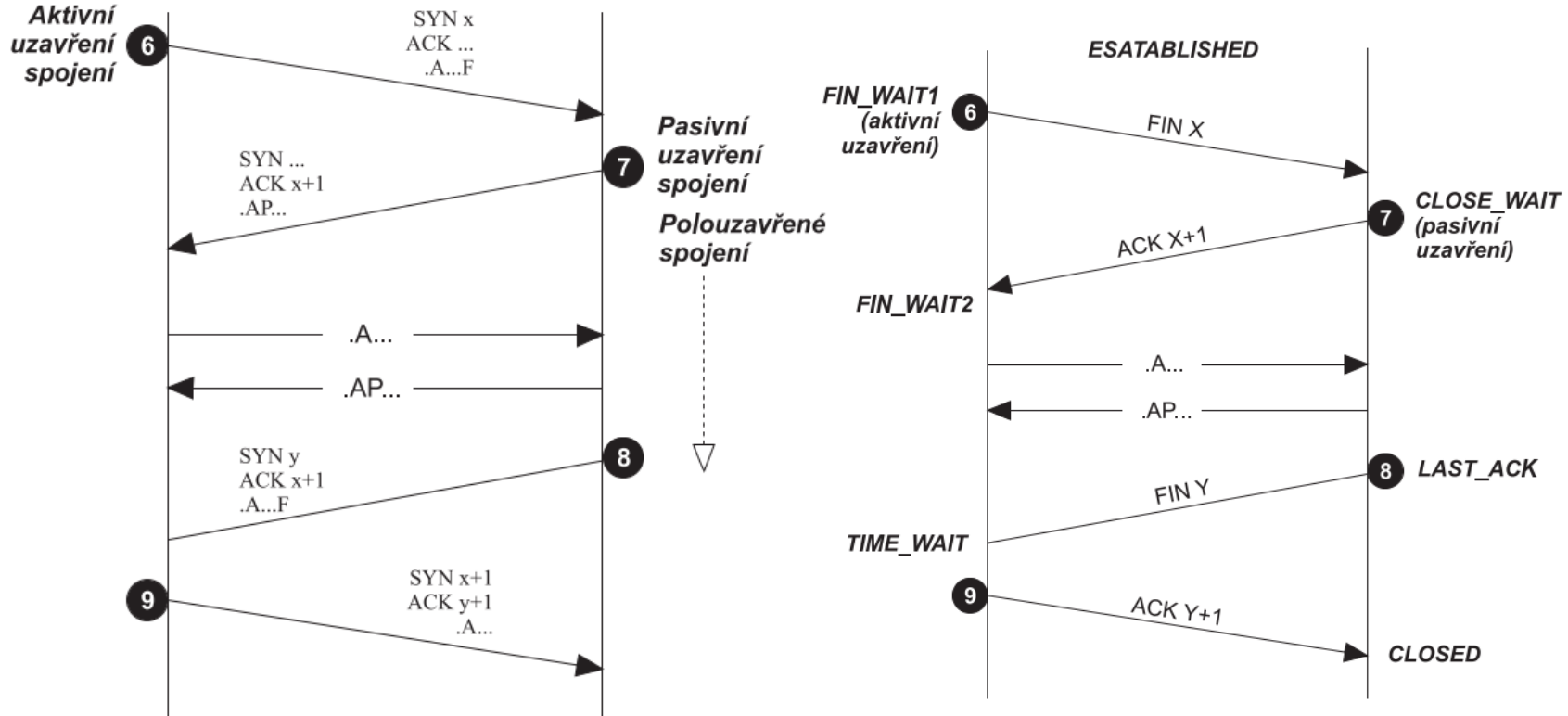
- ◉ Megjegyzés: A háromutas kézfogás üzenetei tipikusan nem szállítanak adatot; ha mégis, akkor azokat pufferelni kell, s a feldolgozásuk csak a kapcsolat kiépülése után kezdődhet el.



# TCP



# TCP KAPCSOLAT LEZÁRÁSA



# NETSTAT -N

PS C:\Users\user> netstat -n

## Active Connections

| Proto | Local Address       | Foreign Address     | State       |
|-------|---------------------|---------------------|-------------|
| TCP   | 127.0.0.1:64067     | 127.0.0.1:64068     | ESTABLISHED |
| TCP   | 127.0.0.1:64068     | 127.0.0.1:64067     | ESTABLISHED |
| TCP   | 192.168.0.102:53865 | 20.185.212.106:443  | ESTABLISHED |
| TCP   | 192.168.0.102:53874 | 52.26.20.242:443    | ESTABLISHED |
| TCP   | 192.168.0.102:139   | 5CG5490MT2:0        | LISTENING   |
| TCP   | 192.168.0.102:53894 | 91.198.174.208:443  | ESTABLISHED |
| TCP   | 192.168.0.102:53896 | 35.244.247.133:443  | ESTABLISHED |
| TCP   | 192.168.0.102:53901 | 185.94.157.11:80    | TIME_WAIT   |
| TCP   | 192.168.0.102:53903 | 204.79.197.200:443  | ESTABLISHED |
| TCP   | 192.168.0.102:53905 | 193.87.80.131:443   | ESTABLISHED |
| TCP   | 192.168.0.102:53906 | 52.114.132.91:443   | TIME_WAIT   |
| TCP   | 192.168.0.102:56876 | 23.97.215.12:443    | ESTABLISHED |
| TCP   | 192.168.0.102:56894 | 138.201.217.169:80  | ESTABLISHED |
| TCP   | 192.168.0.102:54758 | 10.16.90.101:2222   | SYN_SENT    |
| TCP   | 192.168.0.102:57804 | 91.228.167.172:8883 | ESTABLISHED |

PS C:\Users\user>

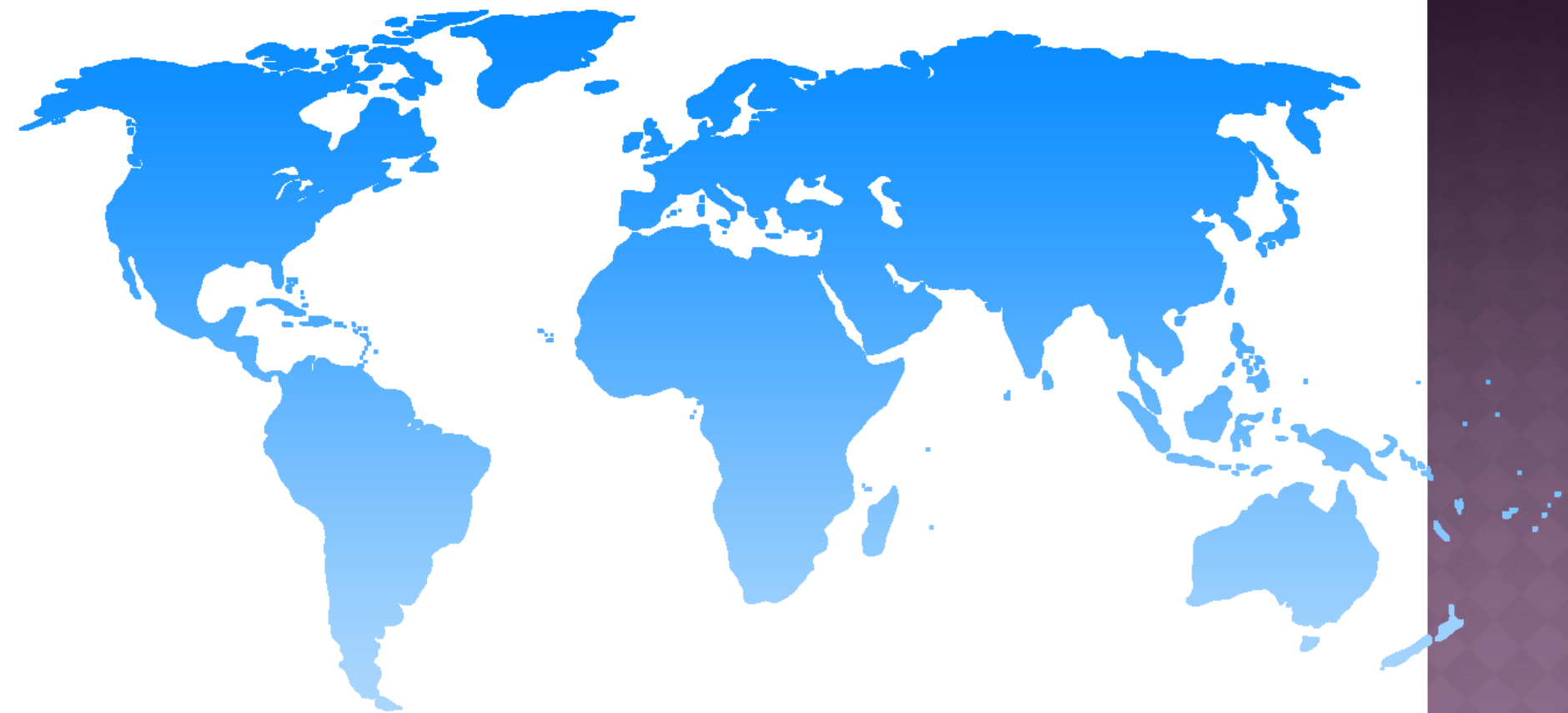
PS C:\Users\user>



# IDŐ A KÉRDÉSEKRE, DISZKUSSZIÓRA

Összefoglaló:

- 🌀 Útválasztás
- 🌀 IGP/EGP
- 🌀 UDP/TCP
- 🌀 Portok





Selye János Egyetem  
Informatika Tanszék  
Gazdaságtudományi és  
Informatakai Kar  
Hradná 21.  
945 01 Komárom

Számítógépes hardver 3  
(Számítógépes hálózatok)  
KIN/PS/IN/12  
Kreditszám: 4  
Tanulmány szintje: I.



**előadás**

# KÖSZÖNÖM A MEGTISZTELTŐ FIGYELMÜKET

Ing. Ondrej Takáč, PhD.  
Informatika Tanszék  
Gazdaságtudományi és Informatakai Kar  
Selye János Egyetem  
takac.ondrej@gmail.com  
+421 35 32 60 629