

# Számítógépes hálózatok

## #03 – VLAN, IP, ARP, RARP

---

2024. szeptember 27.

**Naszlady Márton Bese**

*[naszlady@itk.ppke.hu](mailto:naszlady@itk.ppke.hu)*

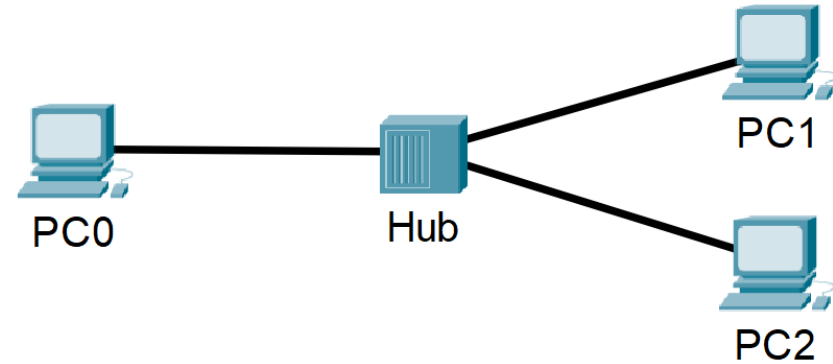
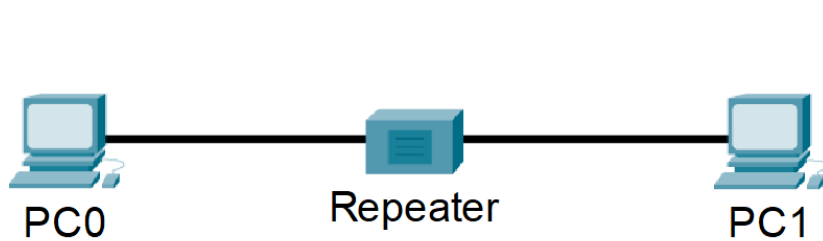
# **#03/1 – L1 és L2 szintű hálózatok**



# L1 és L2 szintek

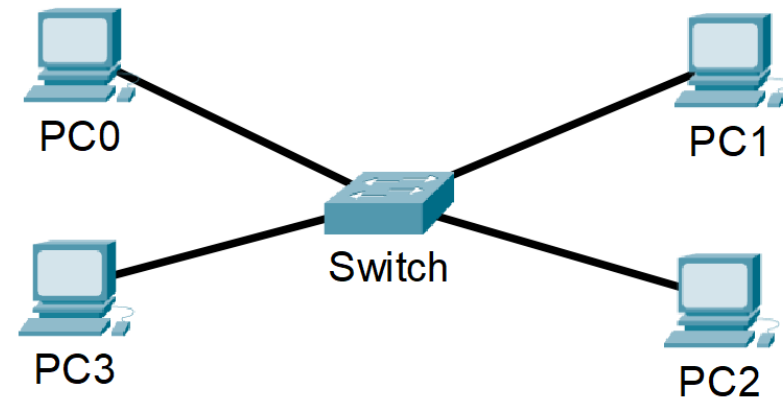
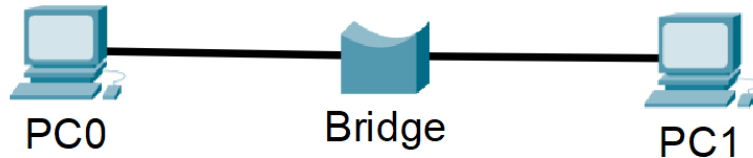
## L1 szint – physical layer

Feladata az eszközök fizikai összeköttetéséhez szükséges feltételek megteremtése.



## L2 szint – data link layer

Feladata az adat linken keresztüli átvitele két szomszédos eszköz között.



# Forgalomirányítás

Physical layer szintjén nincs forgalomirányítás.

Data link layer szinten a MAC címekkel való címzés csak azt mondja meg, hogy **kicsoda** a címzett, azt nem, hogy **merre** keressük.

Az egyes üzenetek célba juttatásához azokat akár mindenki számára is elküldjük, „*hátha közte van a címzett*”.

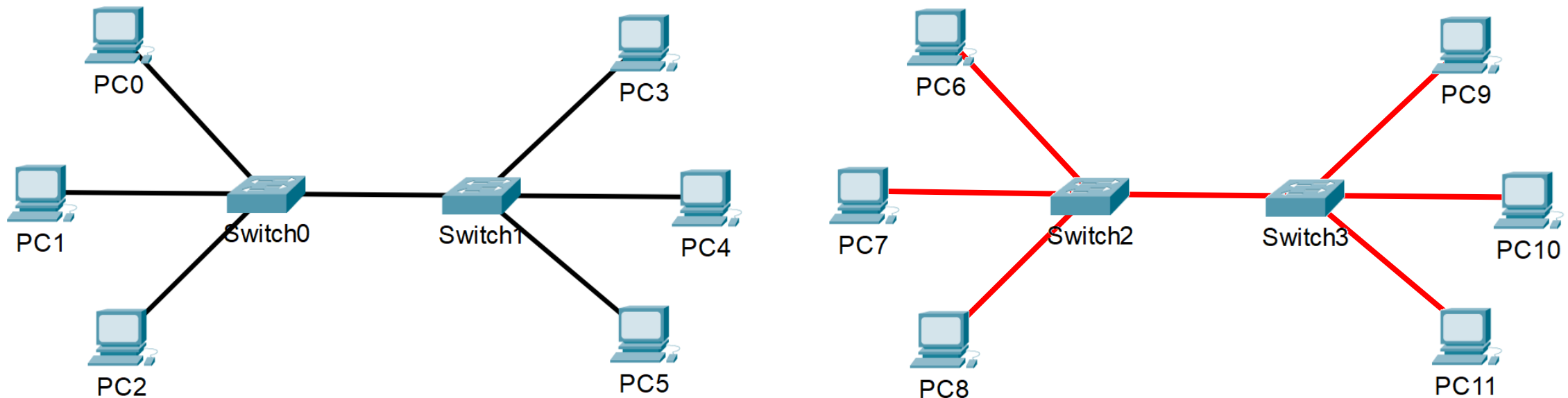
# Szeparáltan működő hálózatok

Felmerül az igény, hogy több hálózatot egymástól függetlenül, különállóan üzemeltessünk.

**Cél:** ne jusson át adat ellenőrizetlenül az egyikből a másikba

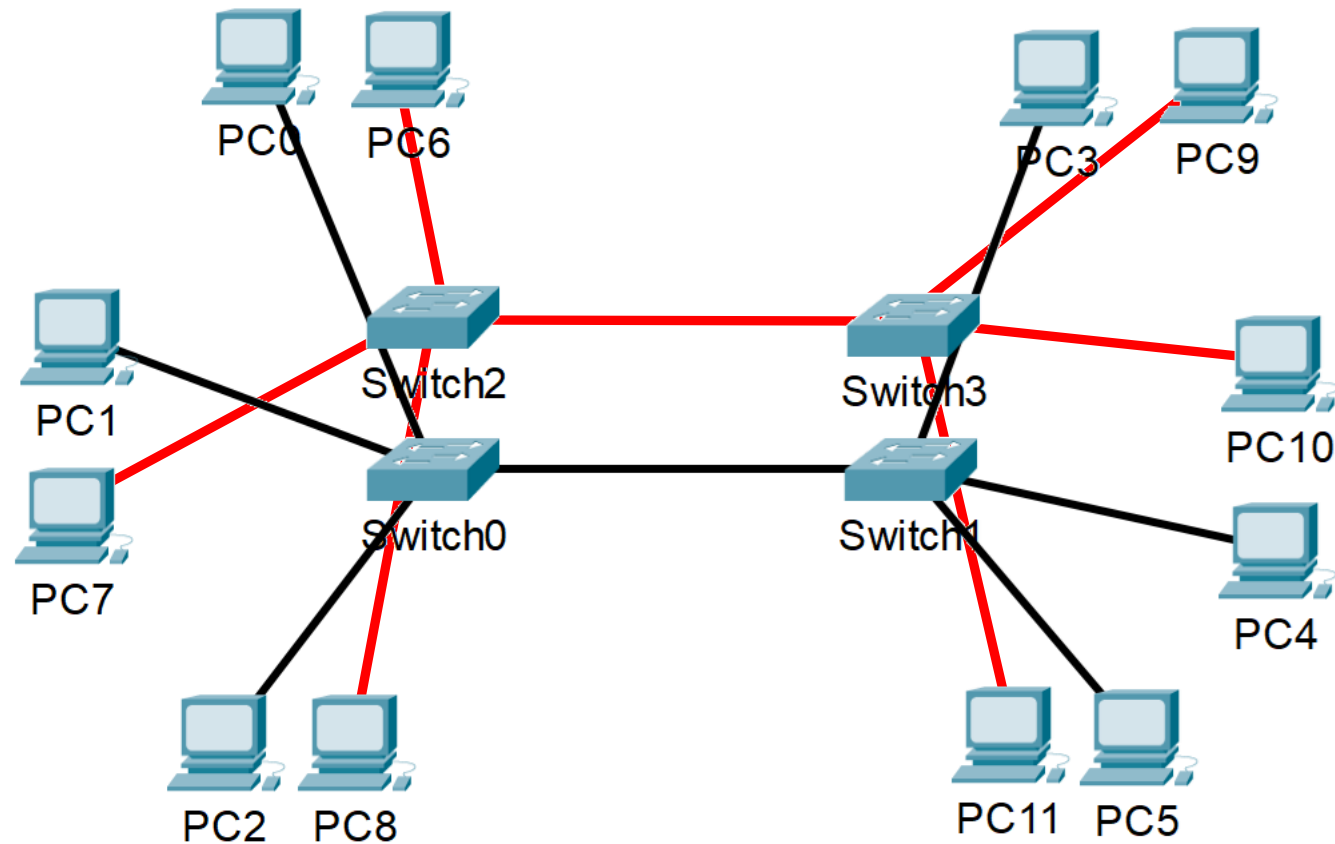
# Szeparáltan működő hálózatok

Lehetséges megoldás: teljesen külön hardverek



# Szeparáltan működő hálózatok

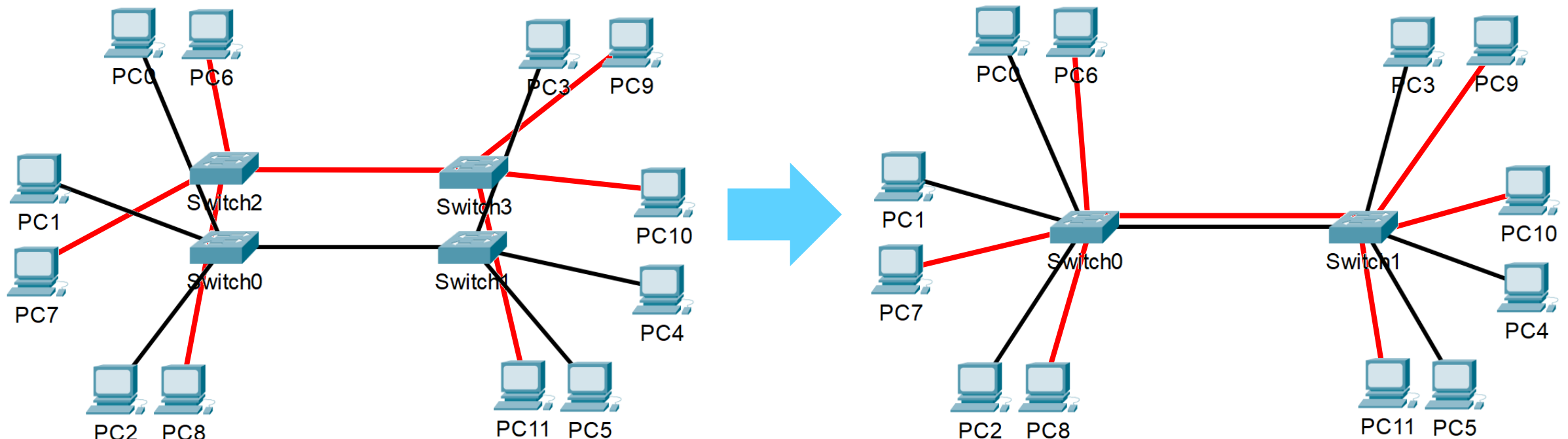
Mi van akkor, ha földrajzilag más a helyzet?



# Szeparáltan működő hálózatok

Előfordulhat, hogy logikailag különböző, de fizikailag közel lévő eszközöket találunk.

Ha ez beállításokkal lehetséges, akkor ugyanaz a switch kezelhetné a piros és fekete hálózat eszközeit is, megoldva a szeparációt.

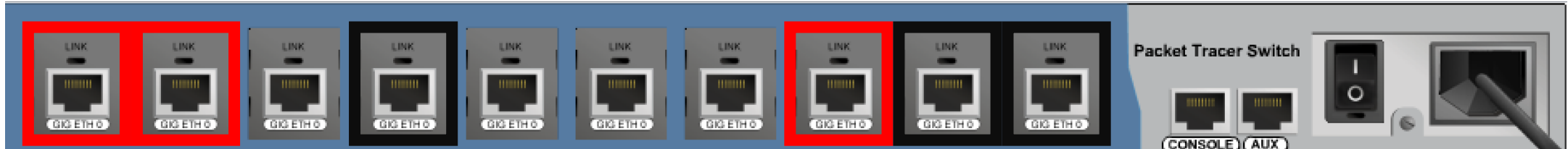




# Virtual LAN



Beállítások révén a switchek egyes physical interface-eit virtuálisan külön hálózatba tartozónak tekinthetjük.



Az adott VLAN hálózatba tartozó interface-en beérkezett frame-et csak az ugyanebbe a VLAN-ba tartozó más interface-ek számára tesszük elérhetővé.

A VLAN hálózatokat 1-től 4094-ig számozzuk.

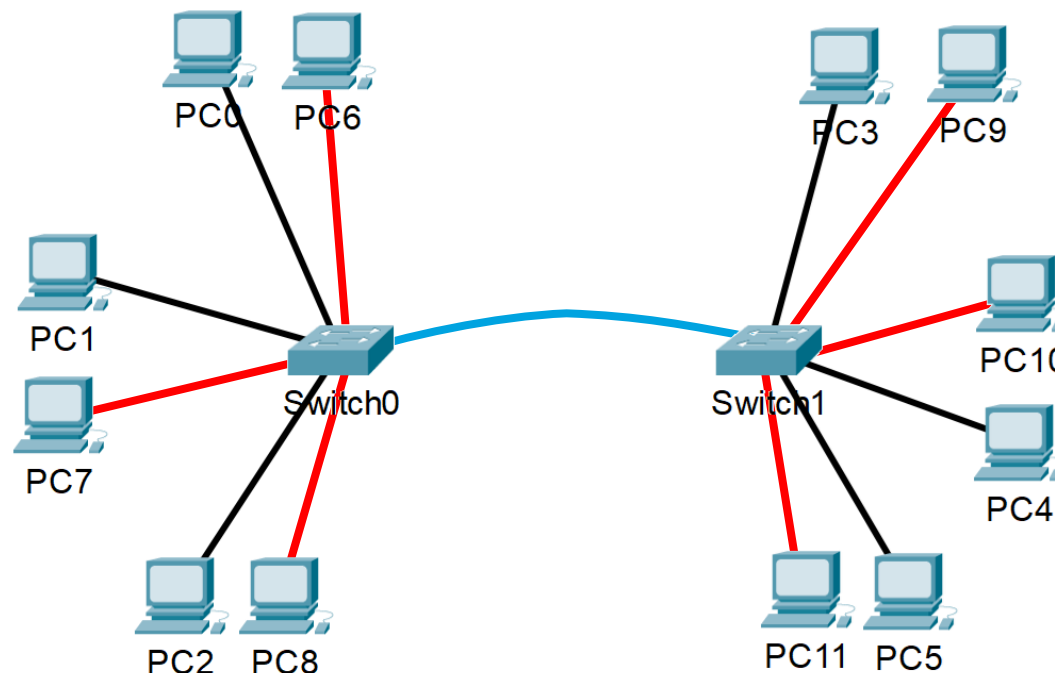
# Switchek közötti összeköttetés



Kellemetlen, hogy a két switch között párhuzamosan halad az összes VLAN számára egy vezeték.

A vezetékek összevonhatók, egy fizikai összeköttetésen egyszerre több logikai összeköttetés és megtörténhet.

Az így kapott összeköttetés neve **trunk**.





# Switchek közötti összeköttetés

Az egy adott VLAN-hoz tartozó interface-en az Ethernet frame-ek változtatás nélkül kerülnek továbbításra.

A trunk linken keresztül átküldött Ethernet frame-ekbe beleírjuk, hogy az adott frame melyik VLAN-ba tartozik.

|                    |               |              |              |                |                    |                        |               |               |
|--------------------|---------------|--------------|--------------|----------------|--------------------|------------------------|---------------|---------------|
| preamble<br>7 byte | SFD<br>1 byte | DA<br>6 byte | SA<br>6 byte | VLAN<br>4 byte | len/type<br>2 byte | data<br>max. 1500 byte | pad<br>n byte | CRC<br>4 byte |
|--------------------|---------------|--------------|--------------|----------------|--------------------|------------------------|---------------|---------------|

A fogadó oldali switch értelmezi ezt az adatot, és a megfelelő interface-ek felé küldi csak tovább a frame-et.

# #03/1 – Összefoglalás

|                 |  |
|-----------------|--|
| <b>Elvek</b>    | Egymástól független hálózatok iránti igény<br>Switch physical interface-ének VLAN-hoz rendelése<br>Trunk |
| <b>Képesség</b> | Megmondani, hogy az adott interface-en lesz-e VLAN ID<br>Megérteni a bővített Ethernet frame struktúrát  |

## **#03/2 – A network layer**

# Forgalomirányítás

Physical layer szintjén nincs forgalomirányítás.

Data link layer szinten a MAC címekkel való címzés csak azt mondja meg, hogy **kicsoda** a címzett.

Egy nagy hálózaton nem gazdaságos a próbálkozás; tudni akarjuk, hogy **merre** küldjük tovább az üzenetet.

# A network layer szerepe



Feladata a data link layer által készített frame-ek forrás és cél közötti útvonalának meghatározása és az adat több közbűlső szereplőn keresztüli célbajuttatása.

*Tipikusan ennek a rétegnek a része:*

- a magasabb szintű címezési hierarchia kialakítása
- forgalomirányítás
- ha szükséges, a nagy üzenetek több részre bontása

# A network layer működése



A network layer szintjén úgy tekintjük, hogy független alhálózatok vannak, melyek között átjárási pontokat határozunk meg.

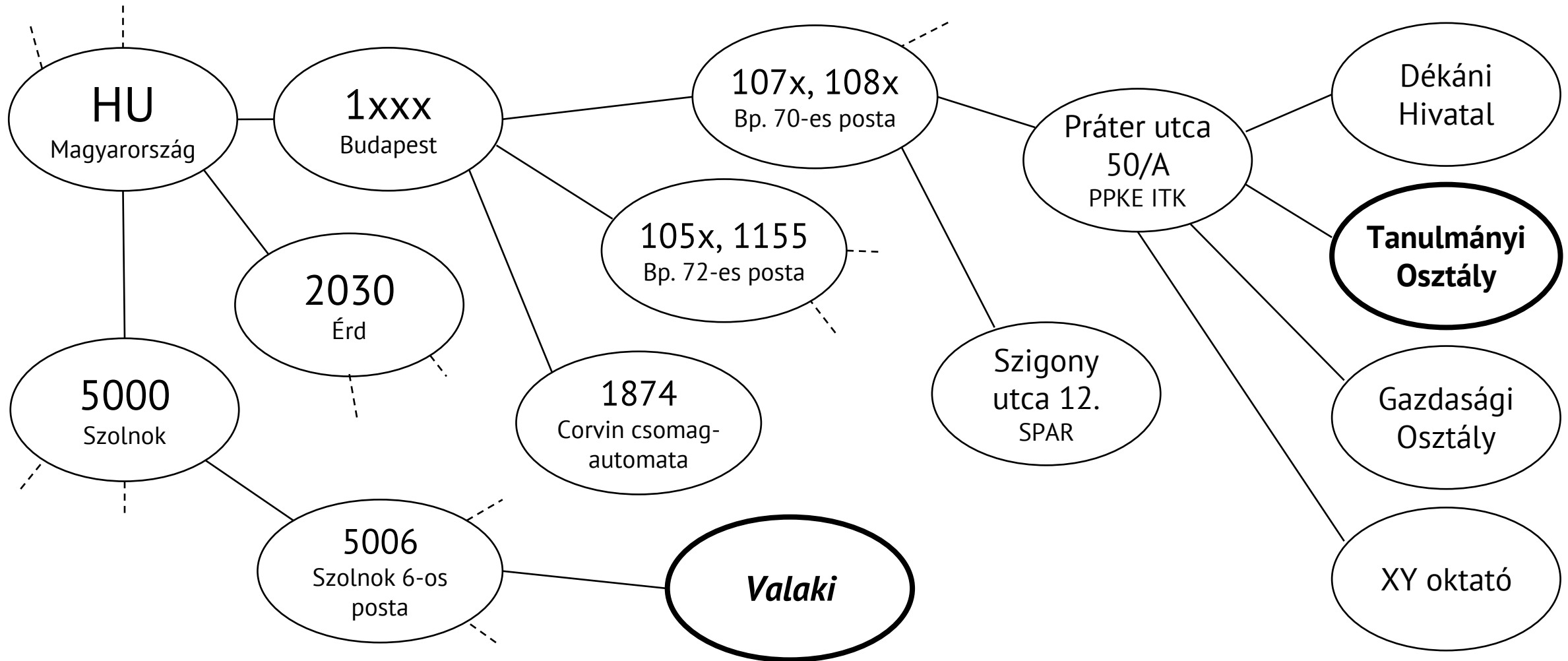
A hálózati eszközök network layer szintű címéből kiderül, hogy azok mely alhálózatba tartoznak.

Az alhálózatok között is szervezhető hierarchia.



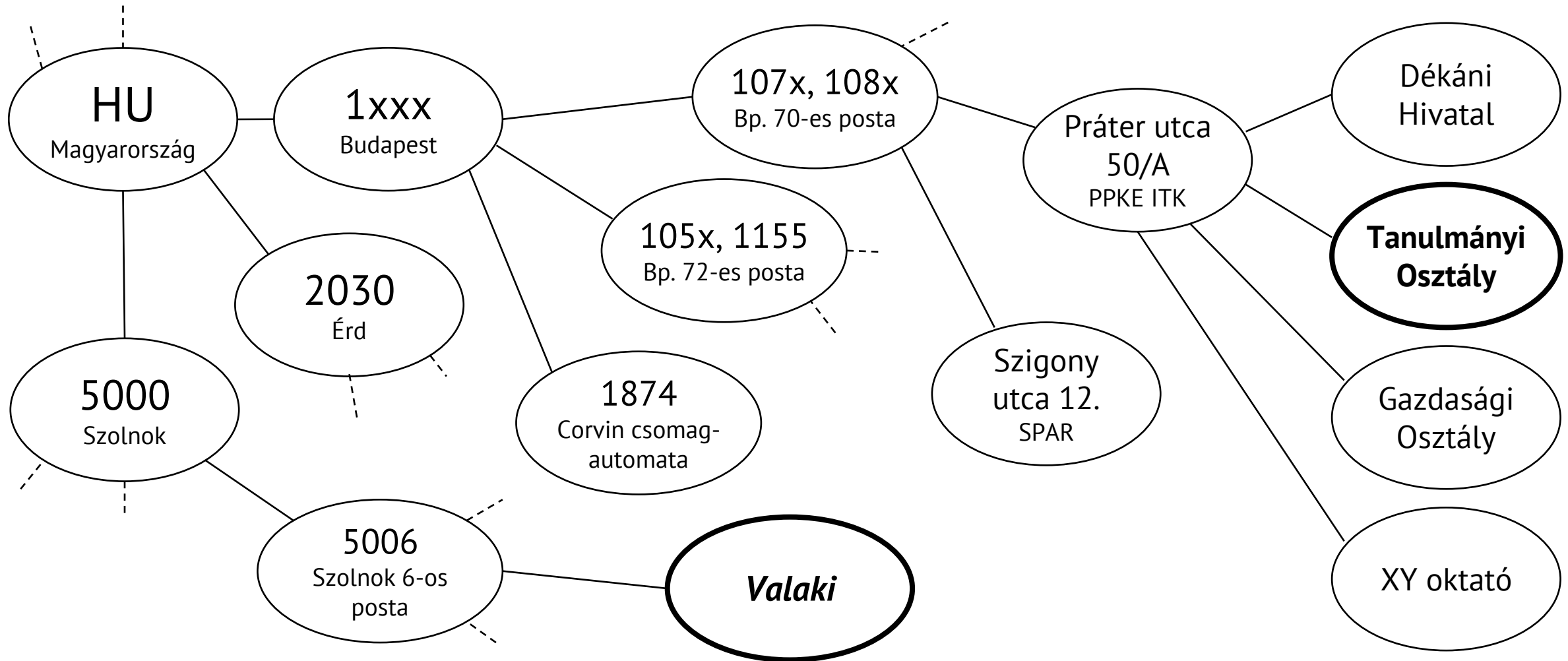
# Posta, mint network layer implementáció

Léteznek hierarchikusan szervezett lokális környezetek, melyen belül lényegében egyenrangú felek vannak.



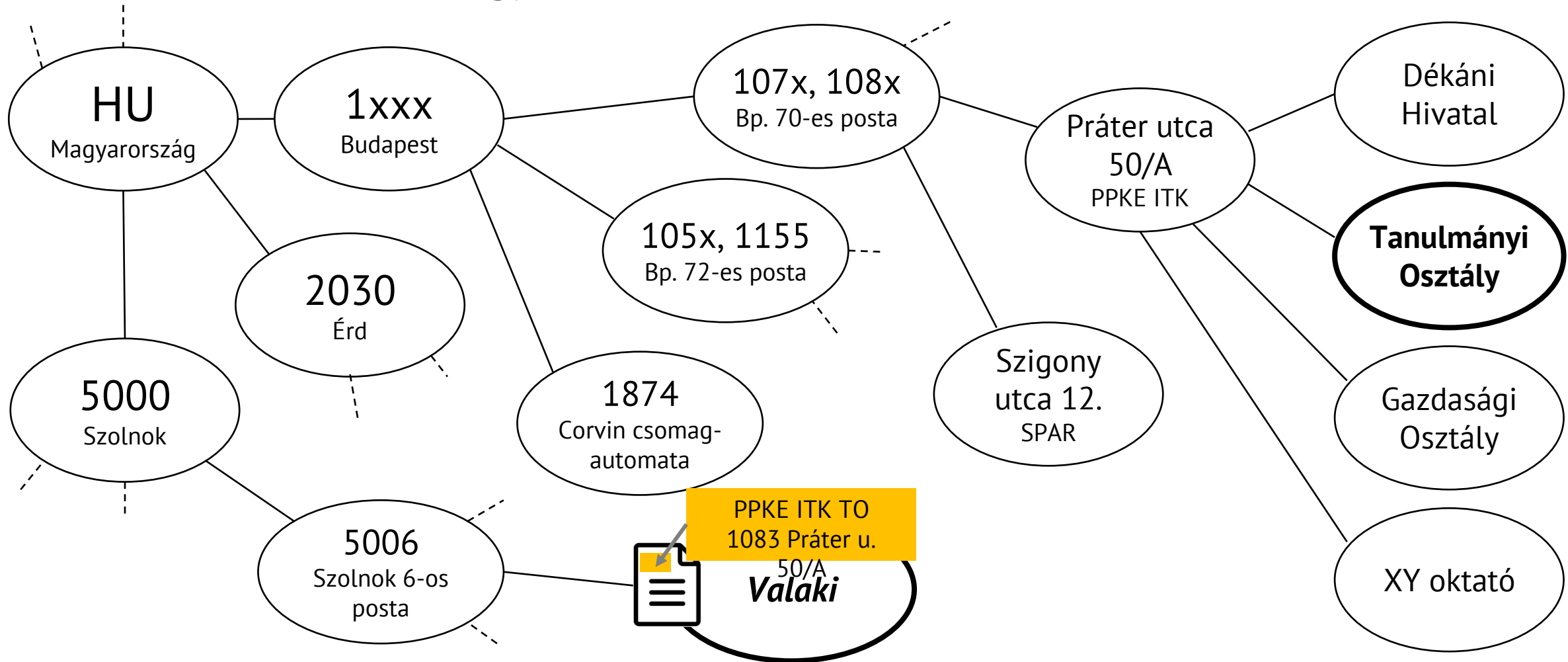
# Posta, mint network layer implementáció

Valaki küld egy postai levelet az ITK Tanulmányi Osztálynak.



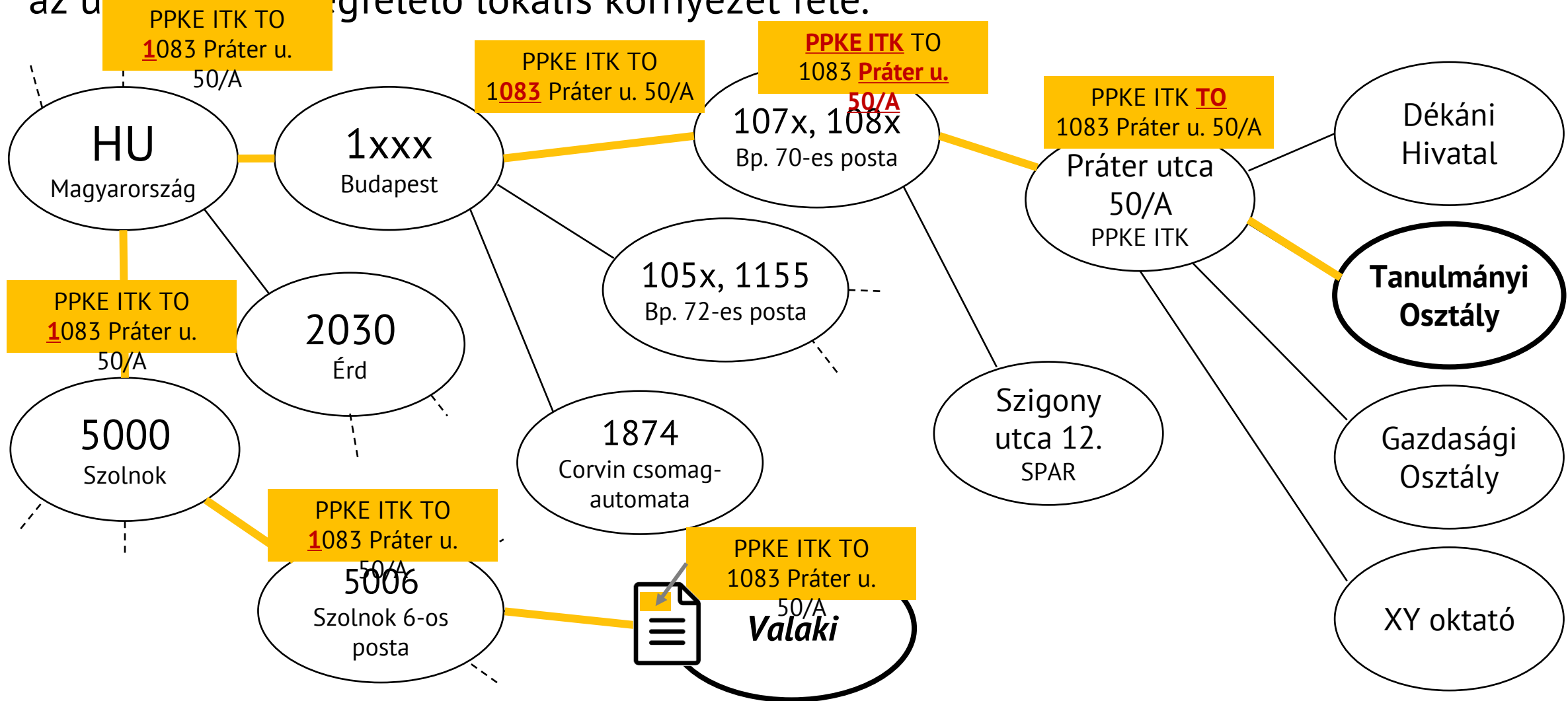
# Posta, mint network layer implementáció

Az üzenet feladásakor azt ki kell egészíteni egy célba juttatást segítő **cím** adattal.  
Ezt a címzést **az adattal együtt** kell továbbítani.



# Posta, mint network layer implementáció

Az üzenet továbbításakor minden node **megnézi a címezést**, és ez alapján küldi tovább az üzenetet a megfelelő lokális környezet felé.



# #03/2 – Összefoglalás

**Elvek**

- Network layer feladata
- Alhálózatok
- Hierarchikus címzés

# **#03/3 – Az Internet Protocol címek**



Az IP a network layer legjellemzőbb implementációja.

Tulajdonságai:

- packet-switched
- nem kapcsolat-orientált
- „best effort”, azaz
  - lehet, hogy egy csomag elveszik
  - lehet, hogy egy csomag többször is megérkezik
  - lehet, hogy megváltozik a csomagok sorrendje

# IPv4 címzés



A cím 4 byteból áll:

decimális jelölés: 193.225.109.159

bináris adat: 11000001.11100001.01101101.10011111

Az IPv4 címek esetén:

- a hálózaton belüli egyediségéről gondoskodni kell
- nem a hálózati eszköznek, hanem a hálózati eszköz interface-ének van IP címe
- a cím első része az alhálózatot azonosítja
- a cím második része az alhálózaton belüli hálózati interface-t azonosítja



## IPv4 címzés

Az IP hajnalán az első byte azonosította az alhálózatot,  
a többi bit pedig az interface-t.

Az internet fejlődésével világossá vált,  
hogy a lehetséges kb. 200 hálózat kevés lesz.

Megoldás: **subnet mask** (alhálózati maszk)



# Subnet mask

A **subnet mask** (netmask) az a bithalmaz, amivel az ugyanabba az alhálózatba tartozó IP címeket maszkolva mindig ugyanazt az értéket kapjuk.

Tehát a netmaskban szereplő 1-es értékek jelzik azokat a bitpozíciókat, amik az alhálózaton belül nem változtatják értéküket.

|                           |                                     |
|---------------------------|-------------------------------------|
| decimális jelölés:        | 193.225.109.159                     |
| bináris adat:             | 11000001.11100001.01101101.10011111 |
| netmask:                  | 11111111.11111111.11111111.00000000 |
| maszkolt érték (AND):     | 11000001.11100001.01101101.00000000 |
| maszkolás után decimális: | 193.225.109.0                       |
| a netmask decimáliasan:   | 255.255.255.0                       |

# Classless Inter-Domain Routing (CIDR)



Mivel az IPv4-es címek első része azonosítja az alhálózatot, a subnet mask megadható úgy is, hogy az abban szereplő 1-es értékek számát (az alhálózaton belül minden IPv4 cím esetében megegyező bitek számát) írjuk le.

A Classless Inter-Domain Routing (**CIDR**) jelölés a következő:

193.225.109.128/26

Ahol a perjel előtti rész az alhálózat legkisebb címe, a perjel utáni pedig az alhálózaton belüli címekben megegyező bitek száma.

# Alhálózat példák

| Alhálózat          | Legutolsó cím   | Címek száma                 | Megjegyzés  |
|--------------------|-----------------|-----------------------------|---|
| 0.0.0.0/0          | 255.255.255.255 | $2^{32} = 4\,294\,967\,296$ | Az összes lehetséges IPv4 cím   |
| 19.0.0.0/8         | 19.255.255.255  | $2^{24} = 16\,777\,216$     | Az IANA által a Ford Motor Company számára kiadott IP címek                           |
| 172.16.0.0/12      | 172.31.255.255  | $2^{20} = 1\,048\,576$      | Az IPv4 specifikáció szerint lokálisan kiosztható címtartományok egyike               |
| 192.168.0.0/16     | 192.168.255.255 | $2^{16} = 65\,536$          | Az IPv4 specifikáció szerint lokálisan kiosztható címtartományok egyike               |
| 193.225.108.0/23   | 193.225.109.255 | $2^9 = 512$                 | A PPKE számára a KIFÜ által kiosztott IPv4 címek                                      |
| 202.11.42.68/31    | 202.11.42.68.69 | $2^1 = 2$                   | Például egy egyetlen Point-to-Point összeköttetést tartalmazó hálózat két interface-e |
| 255.255.255.255/32 | 255.255.255.255 | $2^0 = 1$                   | IPv4 limited broadcast address  |



# IPv4 cím szabályok

Az IP címeknél is vannak unicast, multicast és broadcast címek:

**Unicast címek:** 0.0.0.0 és 223.255.255.255 között

**Multicast címek:** 224.0.0.0 és 239.255.255.255 között

**Broadcast címek:** az (al)hálózat legmagasabb címe

Történelmi okokból a 240.0.0.0 és 255.255.255.254 közötti címeket nem használjuk.

Általában a címtartomány legkisebb címe nem kiosztható; az csak az alhálózatot jelöli.



# IPv4 cím szabályok

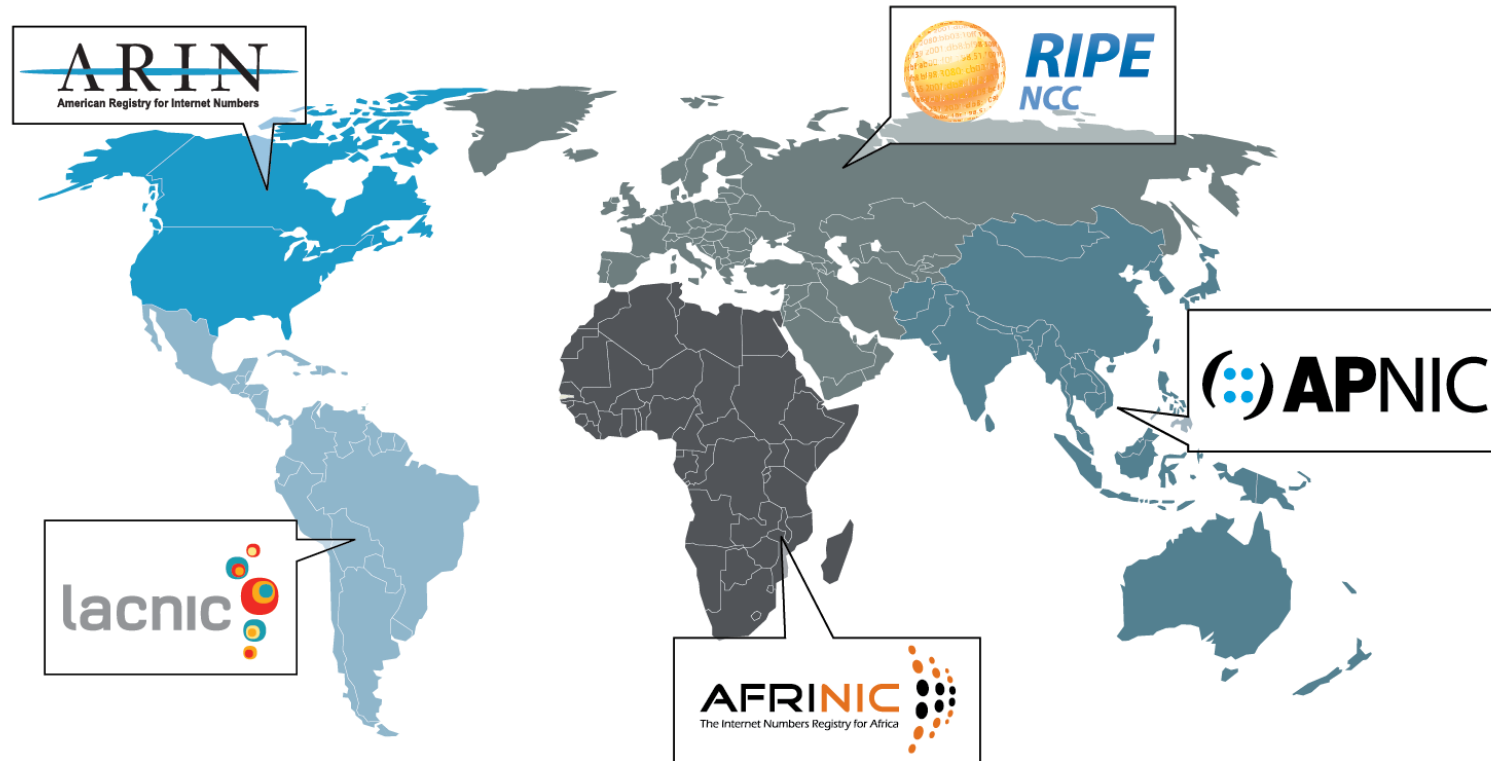
Vannak olyan címek és címtartományok, melyek speciális célokra vannak fenntartva, és csak az adott környezeten (scope) belül használhatók.

A legfontosabb ilyen tartományok:

| Alhálózat      | Legutolsó cím   | Scope           | Megjegyzés   |
|----------------|-----------------|-----------------|--|
| 0.0.0.0/8      | 0.255.255.255   | szoftver        | csak mint a jelen hálózatban lévő ismeretlen IP cím szerepelhet                                |
| 10.0.0.0/8     | 10.255.255.255  | privát hálózat  | csak a privát hálózaton belül jelenhetnek meg, az internetre nem kerülhetnek ki ilyen csomagok |
| 127.0.0.0/8    | 127.255.255.255 | hálózati eszköz | csak az adott hálózati eszközön belül használható (ún. loopback interface-en)                  |
| 172.16.0.0/12  | 172.31.255.255  | privát hálózat  | csak a privát hálózaton belül jelenhetnek meg, az internetre nem kerülhetnek ki ilyen csomagok |
| 192.168.0.0/16 | 192.168.255.255 | privát hálózat  | csak a privát hálózaton belül jelenhetnek meg, az internetre nem kerülhetnek ki ilyen csomagok |

# Globális IPv4 címek kiosztása

A Föld régiókra van osztva. Az egyes régiók IP tartományokat kapnak. Az egyes régiókban belül a különböző szervezetek osztják tovább a címeket.



# #03/3 – Összefoglalás

## **Elvek**

IPv4 cím felépítése

CIDR jelölés

Globális és privát címtartományok

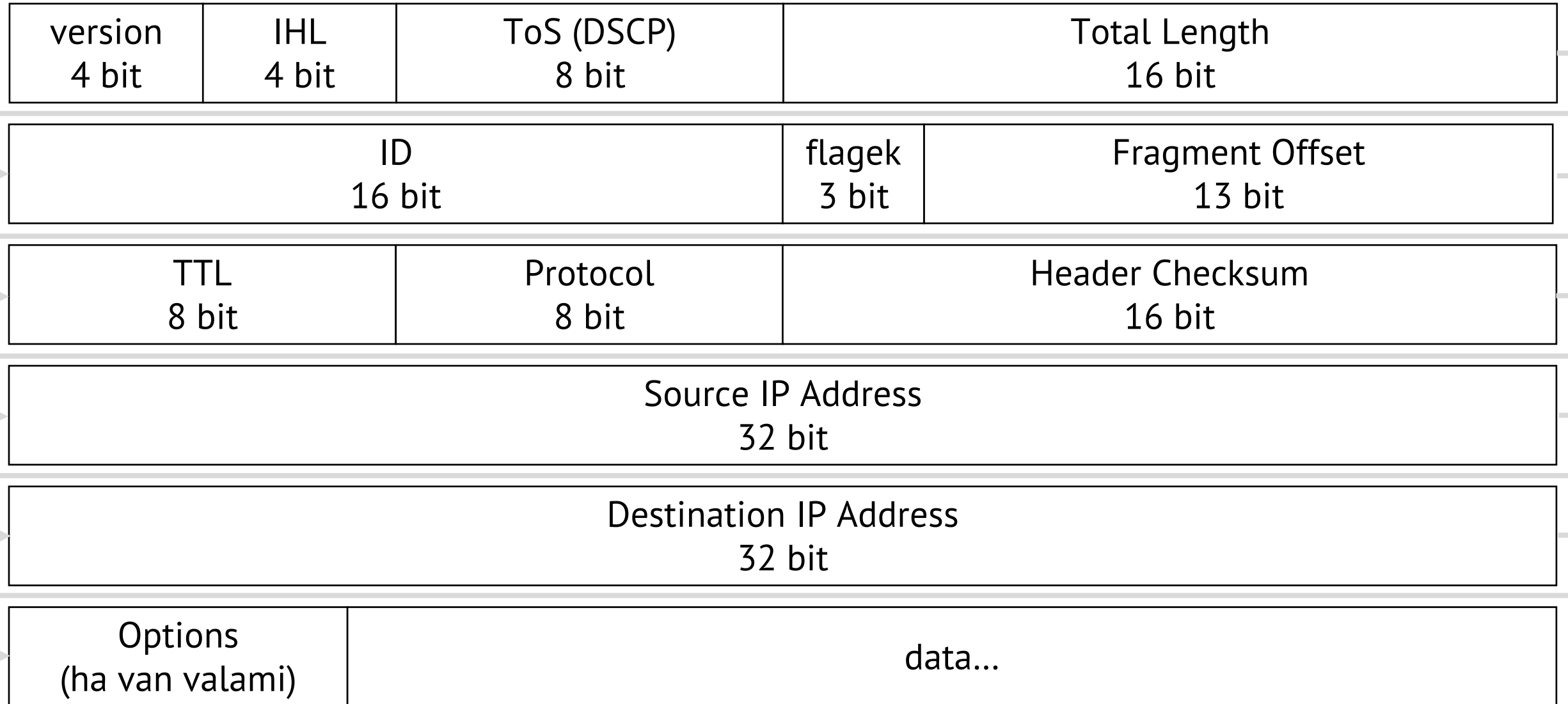
## **Képesség**

IPv4-es cím CIDR felírásából megmondani, hogy mely címek tartoznak még ebbe az alhálózatba



# **#03/4 – Az Internet Protocol csomagformátum**

# IPv4 packet szerkezete



# IPv4 packet szerkezete

## version [4 bit]

- Az IP protokoll verziója (sokáig a 4-es verzió volt, de manapság lehet a 6-os is)

## IHL (IP Header Length) [4 bit]

- Az IP fejrész hossza

## ToS (Type Of Service) [8 bit]

- Az IP csomagba csomagolt adat (szolgáltatás) típusát jelzi. Célja, hogy a csomagok prioritása kiderüljön belőle. Rengetegszer átdolgozott szabvány. A hálózati eszközök figyelmen kívül hagyhatják vagy módosíthatják is.
- Másik megnevezése DSCP (Differentiated Services Code Point)

# IPv4 packet szerkezete

## Total Length [16 bit]

- Az IP csomag teljes hossza byteban. Maximum 65535 lehet az értéke.

## ID [16 bit]

- A csomag azonosítója. Kezdetben inkrementális, később random (biztonság).

## flag-ek [3 bit]

- A túlméretes IP csomagok darabolásánál (fragmentálás) van szerepük.

## Fragment Offset [13 bit]

- Ha fragmentált a csomag, akkor azt mutatja, hogy ez a darab hova illik az eredetiben. Mértékegysége 8 byte.



# IPv4 packet szerkezete

## TTL (Time To Live) [8 bit]

- Az időt ugrásokban méri, felső korlátot ad a max. ugrások számára.
- Minden továbbításnál csökken az értéke, ami ha nulla lesz, megsemmisül.
- Ha nem lenne, a csomagok végtelen ideig keringenének.

## Protocol [8 bit]

- Az IP feletti protokollt azonosítja.

## Fejrész checksum [16 bit]

- Nem CRC hanem sima bit-összeadás; ha jó a csomag, 0-t ad.
- Csak a fejrészt ellenőrzi.

# IPv4 packet szerkezete

## Source IP Address [32 bit]

- A küldő eszköz IP címe.

## Destination IP address [32 bit]

- A címzett eszköz IP címe. E címek alapján történik a csomag továbbítása.

## Opciók

- Egy csomaghoz több opció is fűzhető (hogyan legyen továbbítva).
- A mindennapi gyakorlatban igen ritka, nem használjuk.

# #03/4 – Összefoglalás

|              |                   |
|--------------|-------------------|
| <b>Mezők</b> | Type of Service   |
|              | Time to Live      |
|              | Forrás és cél cím |

# **#03/5 – Forgalomirányítás IP szinten**



Az IP szintű forgalomirányítás feladata  
egy dinamikusan változó,  
irányított, súlyozott gráfban  
minimális költségű utat keresni.

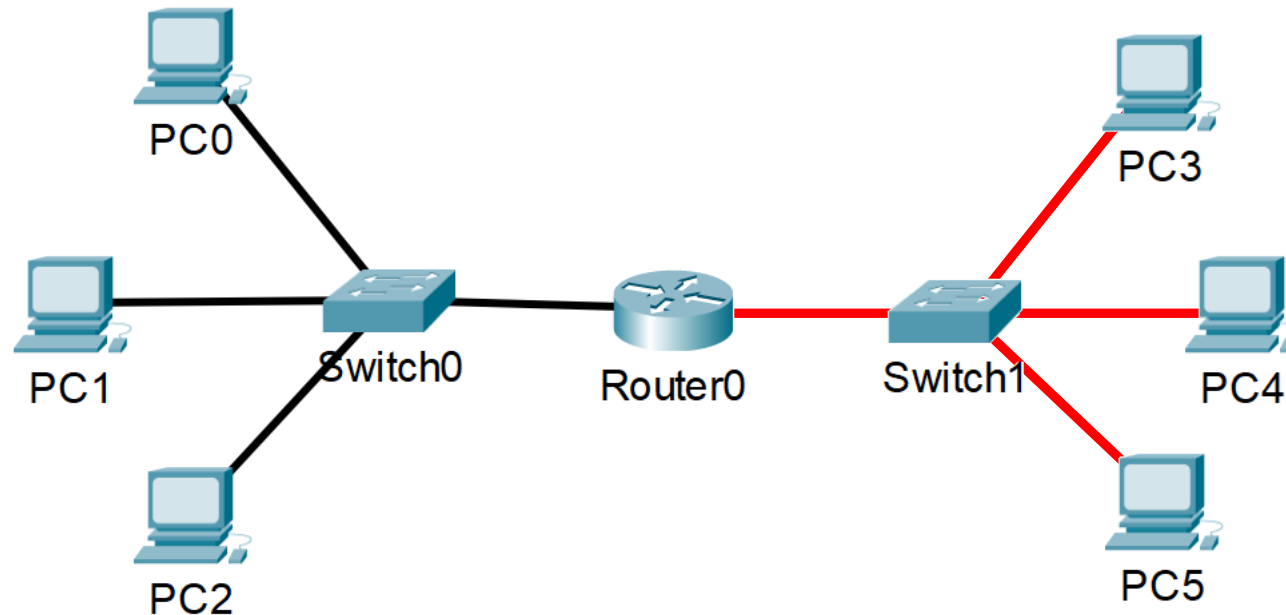
A feladatnak rengetegféle megközelítése és kiterjedt irodalma van.



# Network layer szinten működő eszközök

## Router (gateway, átjáró, útválasztó)

- Legalább két IP címmel rendelkezik
- A beérkező IP csomagokat megpróbálja a címzett alhálózatába továbbítani
- Figyeli a címezést, esetleg egyes IP címtartományokból jövő vagy oda címzett üzeneteket nem visz át.



# Routing tábla



Az eszközöknek tudnia kell,  
hogy melyik CIRD blokkba szánt csomagot hova kell küldeni.

Erre szolgál a ***routing tábla***:

- Minden, alhálózatok közt forgalmazó eszközben jelen van
  - Nem csak a routerben, hanem pl. egy PC-ben, okostelefonban is van.
  - A switch, bridge csak L2 szinten, alhálózaton belül forgalmaz, ezekben nincs routing tábla.
- Megadja, hogy melyik CIDR blokkot melyik interfészen melyik IP-re továbbítsuk
  - Egyszerűbb alhálózatban a feladat általában megoldható egy darab default route-tal.
  - Bonyolultabb hálózatban több irányba is küldhető a csomag, itt az út hossza is számíthat.

# Routing tábla

A routing tábla legalább 3 információt tartalmaz:

- **network identifier**

Az elérni kívánt alhálózat IP címmel és netmaskkal vagy CIDR módon megadva

- **next hop**

Annak az interface-nek az IP címe, ahova a csomagot továbbítani kell

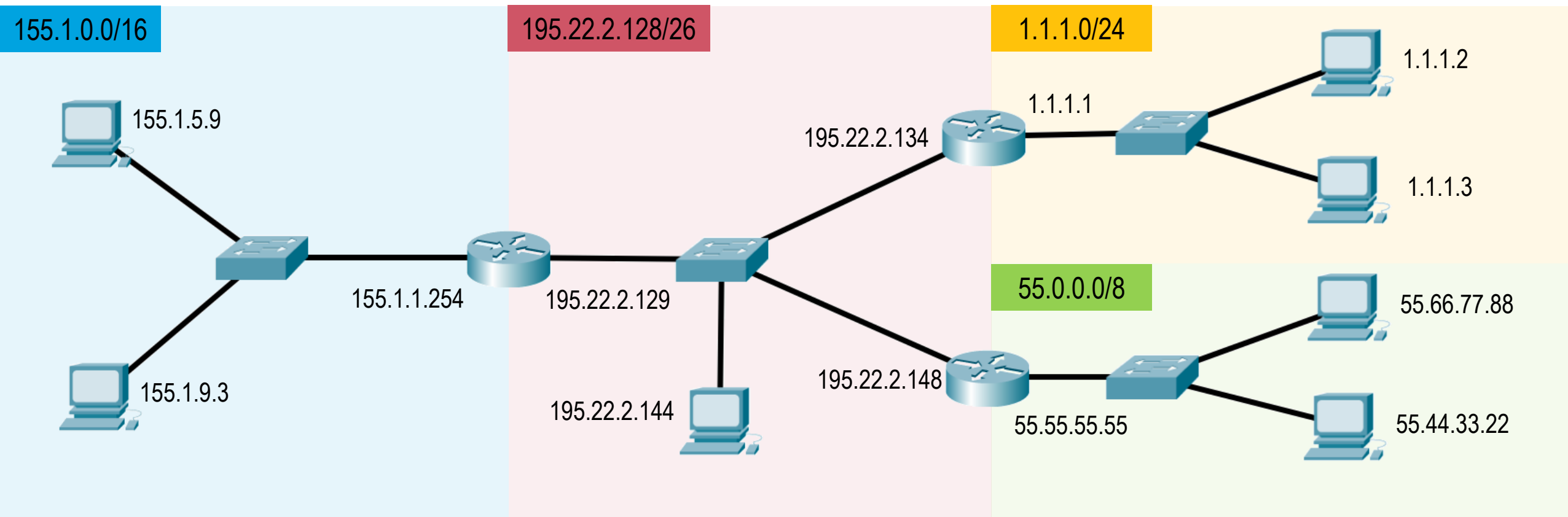
- **metric**

Valamilyen metrika vagy prioritás, ami az útvonal költségét jelzi

| network         | next hop     | metric |
|-----------------|--------------|--------|
| 155.1.0.0/16    | 195.22.2.129 | 1      |
| 195.22.2.128/26 | -            | 0      |
| 1.1.1.0/24      | 195.22.2.134 | 1      |
| 55.0.0.0/8      | 195.22.2.144 | 1      |

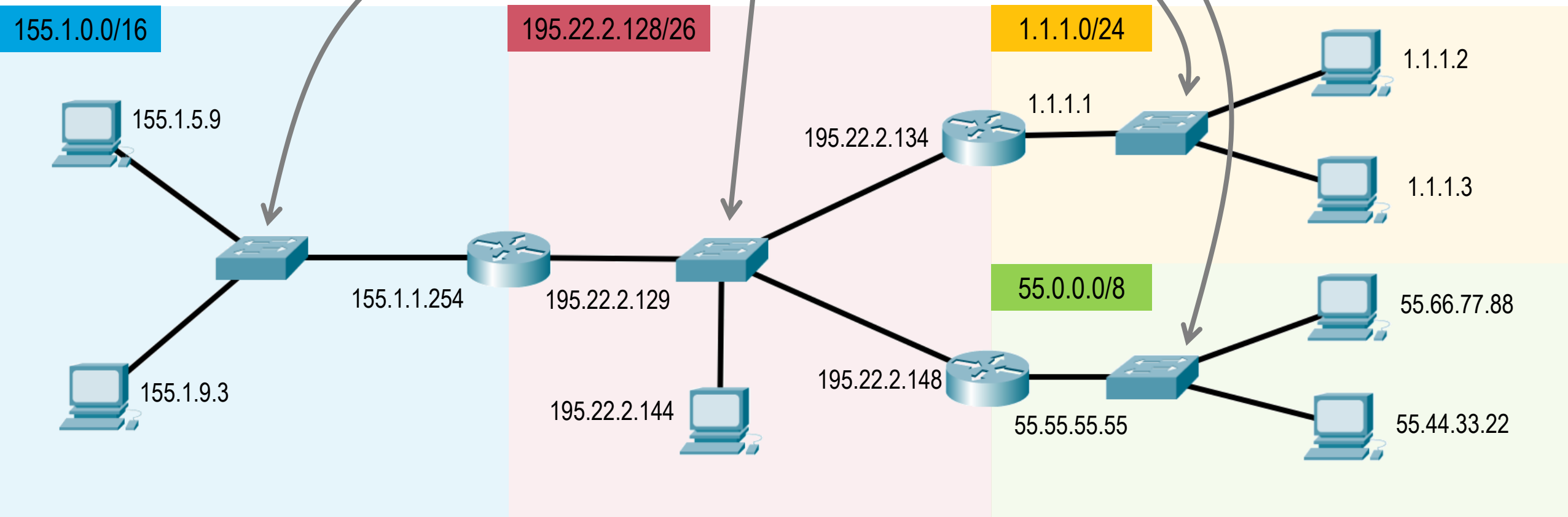
# Routing tábla példák

Tekintsük az alábbi, négy alhálózathból (subnet) álló hálózatot.  
Néhány példa az eszközökben lévő routing táblára:



# Routing tábla példák

A switchek nem végeznek L3 szintű forgalomirányítást, így bennük nincs routing tábla.

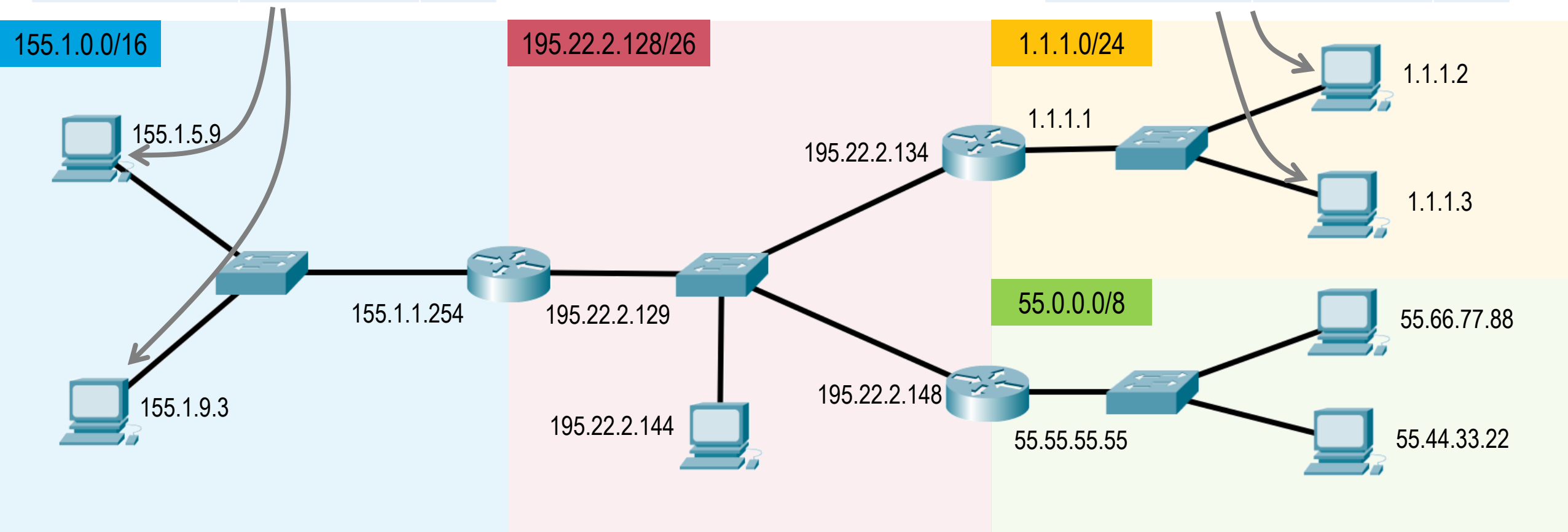


# Routing tábla példák

Egyes eszközökben csak egy „*default route*” található meg.  
Minden, nem a saját alhálózatba címzett csomagot erre a címre továbbít.

| network      | next hop    | metric |
|--------------|-------------|--------|
| 0.0.0.0/0    | 195.1.1.254 | ?      |
| 155.1.0.0/16 | -           | 0      |

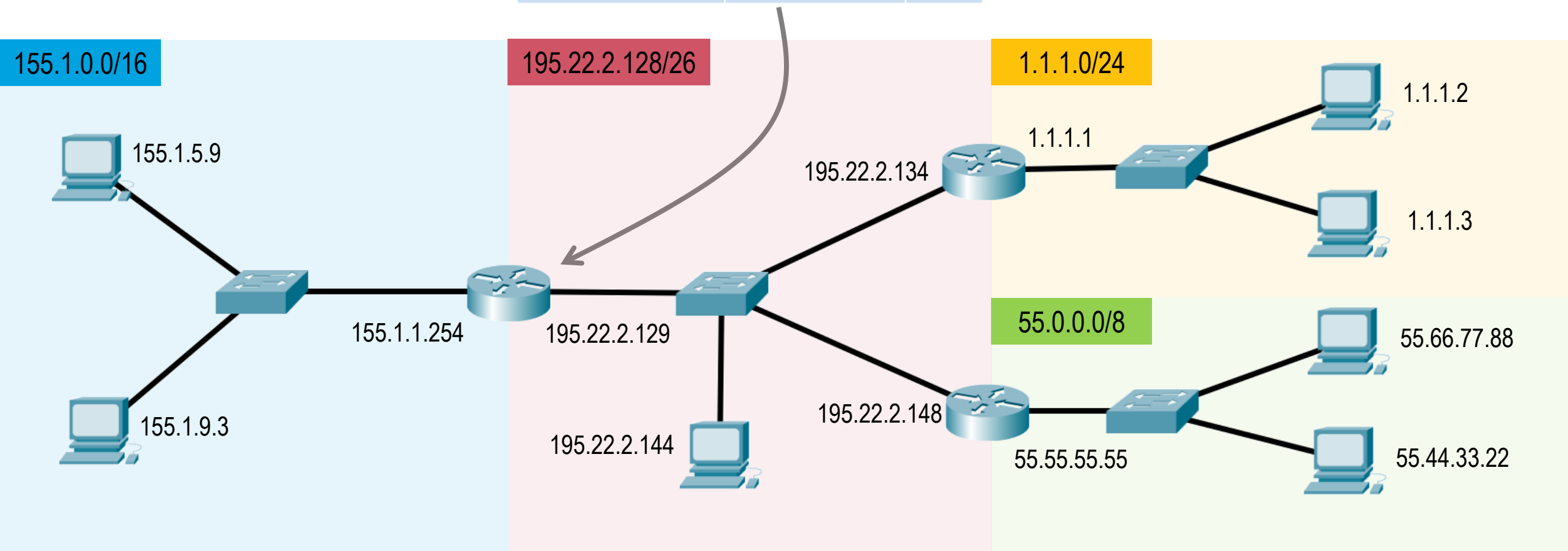
| network    | next hop | metric |
|------------|----------|--------|
| 0.0.0.0/0  | 1.1.1.1  | ?      |
| 1.1.1.0/24 | -        | 0      |



# Routing tábla példák

A default route megoldás akár egyes routerekben is előfordulhat.

| network         | next hop     | metric |
|-----------------|--------------|--------|
| 0.0.0.0/0       | 195.22.2.134 | ?      |
| 155.1.0.0/16    | -            | 0      |
| 195.22.2.128/26 | -            | 0      |



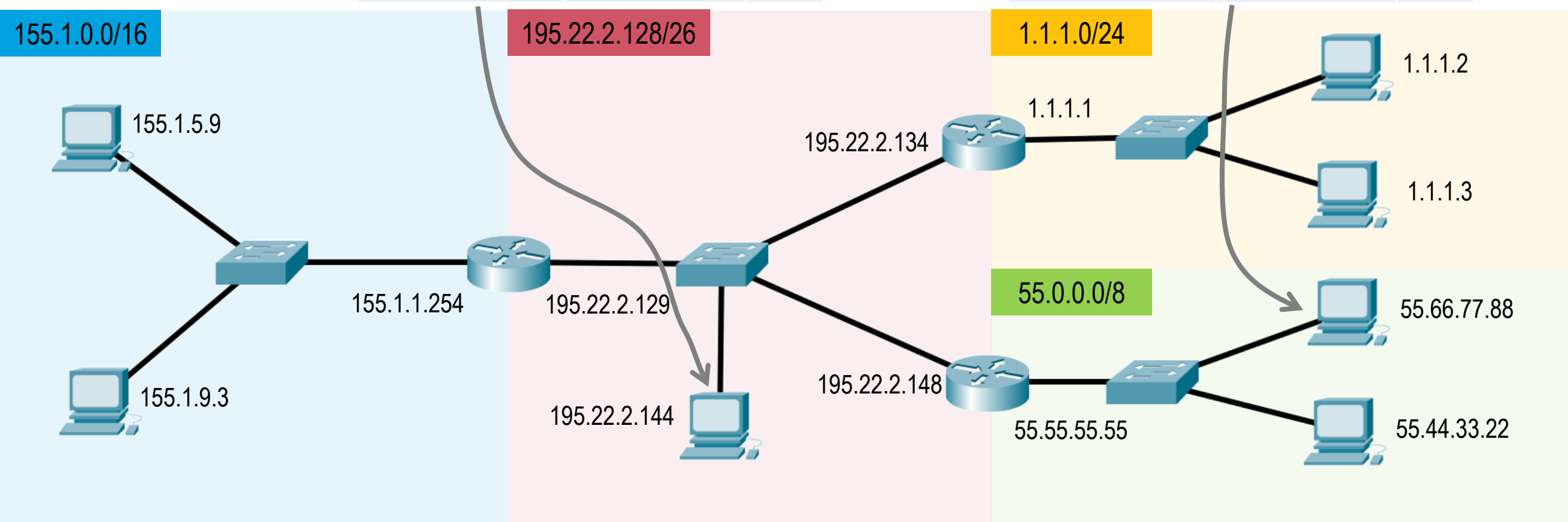


# Routing tábla példák

És természetesen az is jó megoldás, ha a hálózat pontos leírása van a táblában:

| network         | next hop     | metric |
|-----------------|--------------|--------|
| 195.22.2.128/26 | -            | 0      |
| 55.0.0.0/8      | 195.22.2.144 | 1      |
| 1.1.1.0/24      | 195.22.2.134 | 1      |
| 155.1.0.0/16    | 195.22.2.129 | 1      |

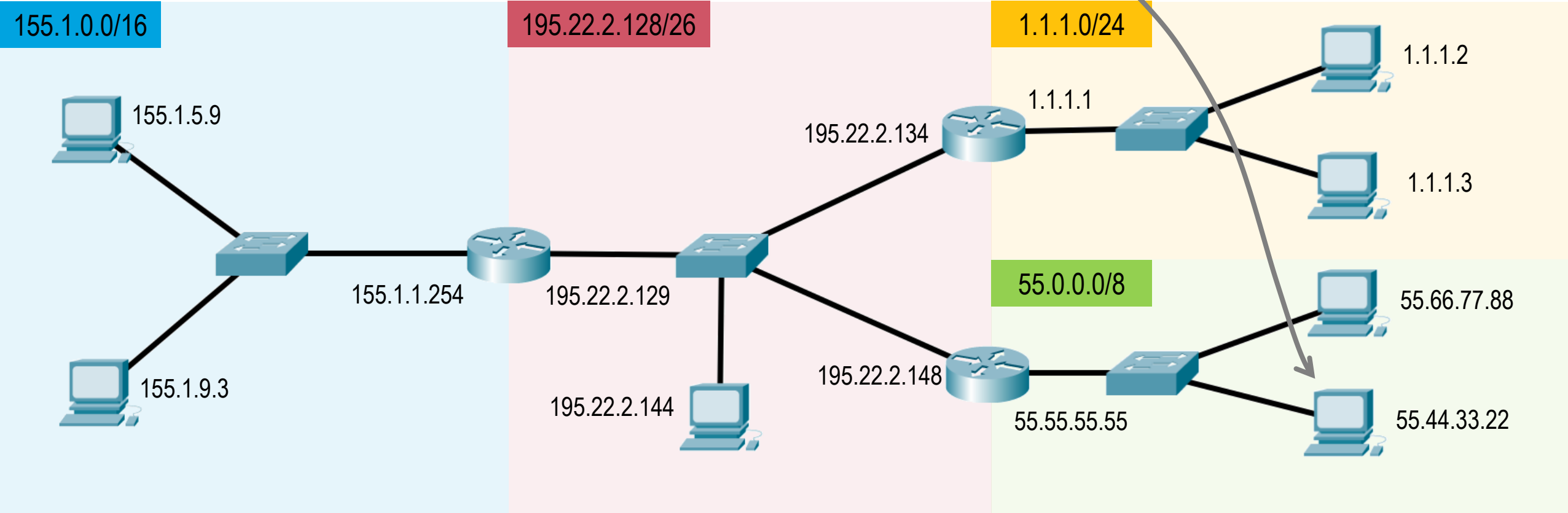
| network         | next hop    | metric |
|-----------------|-------------|--------|
| 55.0.0.0/8      | -           | 0      |
| 195.22.2.128/26 | 55.55.55.55 | 1      |
| 1.1.1.0/24      | 55.55.55.55 | 2      |
| 155.1.0.0/16    | 55.55.55.55 | 2      |



# Routing tábla példák

Végül az is előfordulhat, hogy a routing táblában nincs default route, vagy a megadott next hop lehetetlen helyre mutat. A csomag továbbítása ilyenkor nem lehetséges.

| network    | next hop    | metric |
|------------|-------------|--------|
| 0.0.0.0/0  | 155.1.1.254 | ?      |
| 55.0.0.0/8 | -           | 0      |



# Ismétlés

## Ethernet cím (MAC cím)

00:00:0C:49:F4:31

Az interface azonosítója L2 szinten.

*Használata:*

Ethernet frame-ek átvitelére két node között egy lokális hálózatban.

## IP cím (IPv4 cím)

192.168.1.4

Az interface azonosítója L3 szinten.

*Használata:*

Az IP csomagok átvitele két eszköz között, sok link és node közbeiktatásával az interneten.

# Layerek viselkedése az adattovábbításban

## L1 – Physical layer viselkedése

- A beérkező fizikai jelenséget adattá alakítja. Ha ez rendben van (pl. checksum OK), akkor az adatot a felettes rétegnek adja tovább.
- A felettes rétegtől kapott adatot fizikai jelenséggé alakítja, és kiküldi az interface-en.

## L2 – Data link layer viselkedése

- A beérkező frame-et megvizsgálja, ha ő volt a címzett, akkor a frame-be csomagolt adatot feladja a felettes réteg számára.
- Ha nem ő a címzett, akkor vagy eldobja (végponti eszköz) vagy megpróbálja másik interface-en továbbítani (switch) a frame-et.
- A felettes rétegtől kapott adatot frame-be csomagolja, és azt átadja az alatta álló réteg számára.

# Layerek viselkedése az adattovábbításban

## L3 – Network layer viselkedése

- A beérkező packetet megvizsgálja, ha ő volt a címzett, akkor a packet-be csomagolt adatot feladja a felettes réteg számára.
- Ha nem ő a címzett, akkor vagy eldobja (végponti eszköz) vagy megpróbálja másik interface-en továbbítani (router) a packetet.
- A felettes rétegtől kapott adatot packetbe csomagolja, és azt átadja az alatta álló réteg számára.

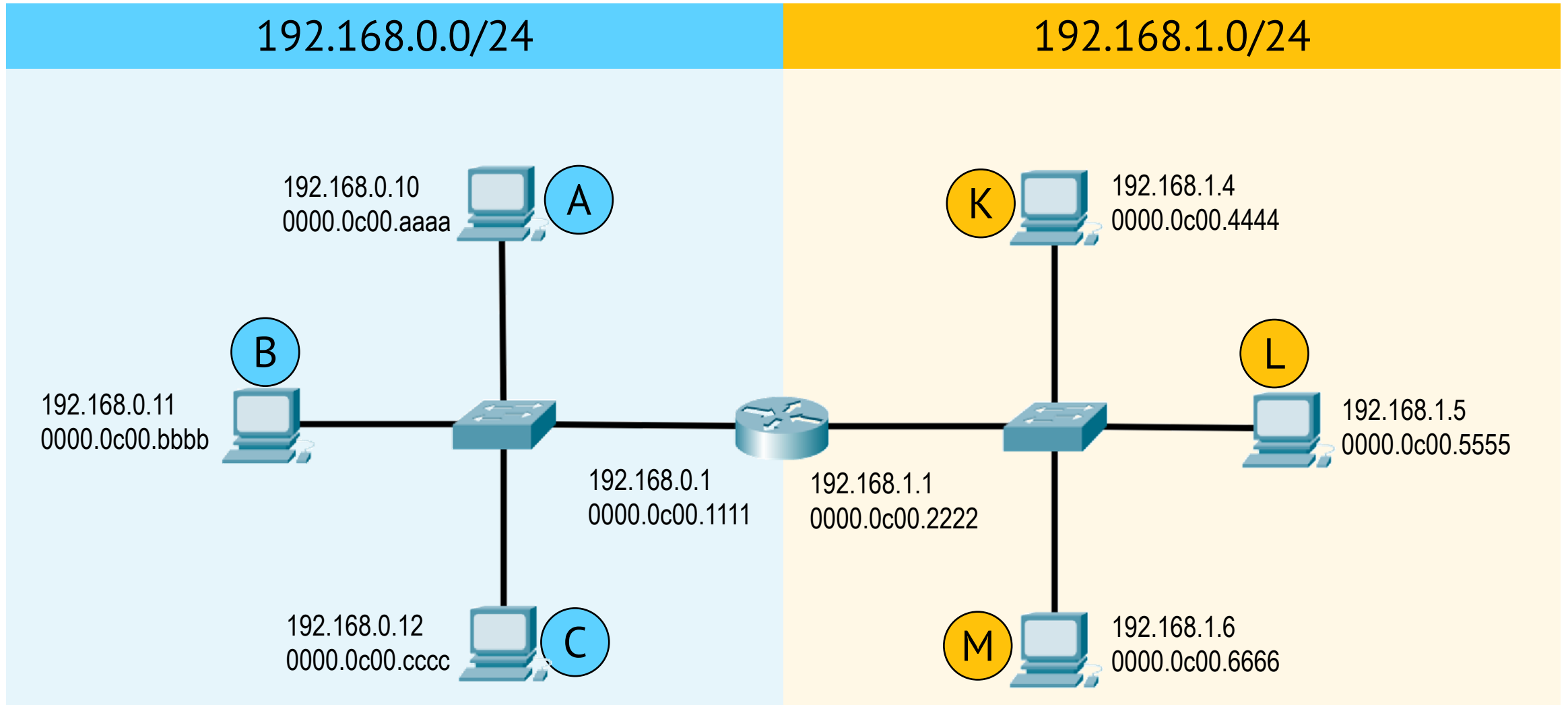
# Csomag útjának végigkövetése

**Nézzünk most két példát!**

1. IP csomagküldés alhálózaton belüli eszközök között
2. IP csomagküldés különböző alhálózatba tartozó eszközök között

# Alhálózaton belüli csomagküldés


Legyen adott az alábbi, két alhálózatból álló hálózat.



# 1. Példa – Csomagküldés alhálózaton belül

Az  eszköz adatot szeretne küldeni a  eszköznek.

Hogyan zajlik a folyamat?

1. Az  eszközben a *transport layer* átadja az adatot a *network layer*-nek.
2. A *network layer* előállítja az IP csomagot, ehhez megadja a feladó és a címzett IPv4 címét (és természetesen a többi paramétert is – hossz, verzió stb.)
3. A *data link layer* megkapja a *network layer* által előállított IP csomagot, és azt egy Ethernet frame-be teszi. Ehhez megadja a feladó és a következő állomás fizikai címét (MAC címét).
4. A *physical layer* ténylegesen mozgatja az adatot a két pont között.



# 1. Példa – Csomagküldés alhálózaton belül

5. A fogadó állomáson belüli *physical layer* visszaállítja az Ethernet frame-et, és feladja azt a *data link layer* réteg számára.
6. A *data link layer* megvizsgálja, hogy ez az L2 szintű interface volt-e a címzett (ennek a MAC címe szerepel-e a frame-ben), és ha igen, akkor kicsomagolja belőle az IPv4 csomagot, és feladja azt a *network layer* számára.  
Ha nem ez az interface volt a címzett, akkor figyelmen kívül hagyja a frame-et.
7. A *network layer* megvizsgálja, hogy ez az L3 szintű interface volt-e a címzett (ennek IPv4 címe szerepel a csomagban), és ha igen, akkor kicsomagolja belőle az adatot, és azt feladja a *transport layer* számára.  
Ha nem ez az interface volt a címzett, akkor megpróbálja a címzett felé továbbküldeni az IPv4 csomagot.

# 1. Példa – Csomagküldés alhálózaton belül

Az  eszköz adatot szeretne küldeni a  eszköznek.

Az adatátvitelben érintett összes node csak a saját tudására és a csomagban előforduló jelzésekre támaszkodik. Nincs „globális tudás” vagy előre egyeztetés az átvitelről.

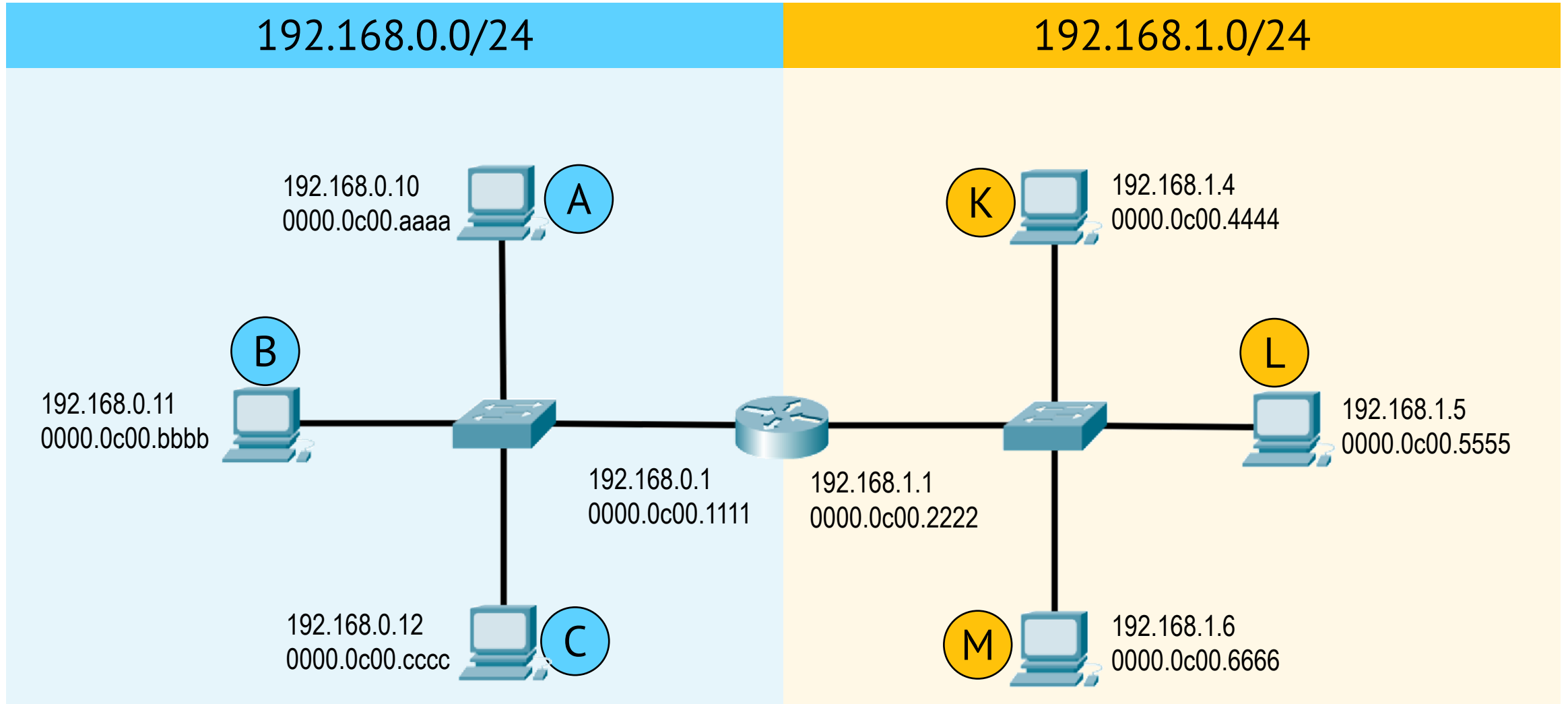
Milyen ismeretek szükségesek ehhez?

- A *transport layer*-től meg kell kapni az adatot.
- A *network layer*-nek tudnia kell a feladó és a címzett IPv4 címét.
- A *data link layer*-nek tudnia kell a feladó és a következő node MAC címét.

# 1. Példa – Csomagküldés alhálózaton belül

Az **A** eszköz adatot szeretne küldeni a **C** eszköznek.

A hálózati struktúra:

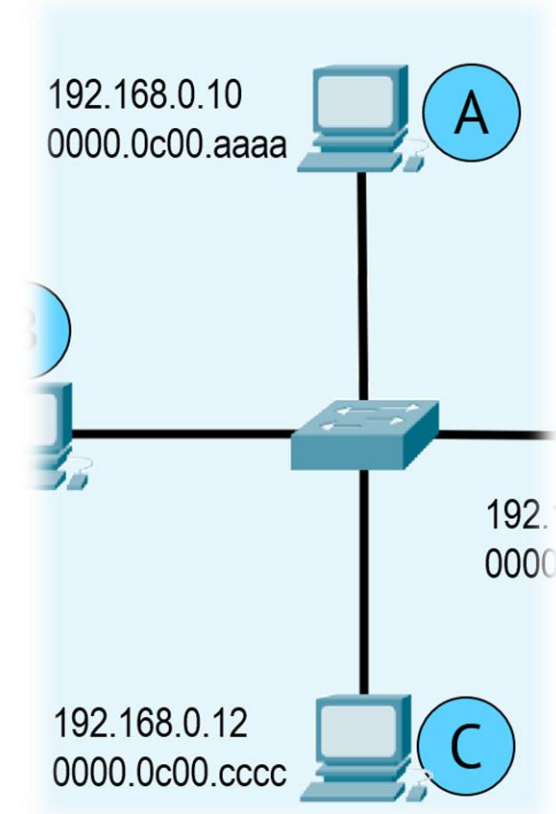


# 1. Példa – Csomagküldés alhálózaton belül

Az **A** eszköz adatot szeretne küldeni a **C** eszköznek.

- *Ki lesz a feladó L3 szinten?*  
**192.168.0.10**
- *Ki lesz a címzett L3 szinten?*  
**192.168.0.12**
- *Ki lesz a feladó L2 szinten?*  
**0000.0c00.aaaa**
- *Ki lesz a címzett L2 szinten?*


A feladó kiszámolja, hogy a címzett vele egy alhálózaton van-e → IGEN  
A címzett közvetlenül elérhető; az ő MAC címe kerül feltüntetésre.  
**0000.0c00.cccc**

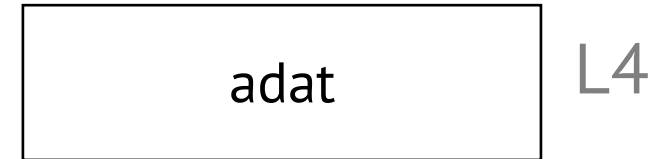


# 1. Példa – Csomagküldés alhálózaton belül

Az  eszköz adatot szeretne küldeni a  eszköznek.

Hogyan zajlik a folyamat?

1. Az  eszközben a *transport layer* átadja az adatot a *network layer*-nek.

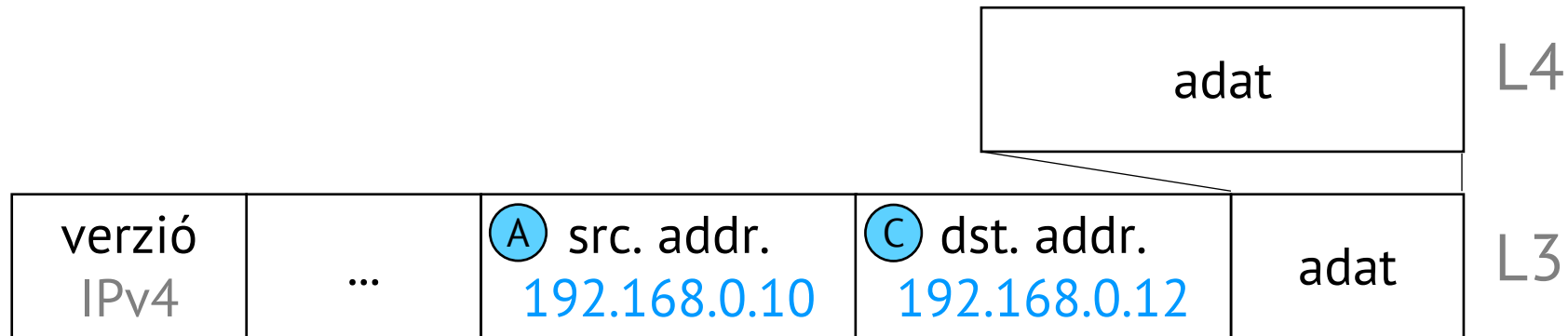


# 1. Példa – Csomagküldés alhálózaton belül

Az **A** eszköz adatot szeretne küldeni a **C** eszköznek.

Hogyan zajlik a folyamat?

2. A *network layer* előállítja az IP csomagot, ehhez megadja a feladó és a címzett IPv4 címét, és a többi szükséges paramétert.

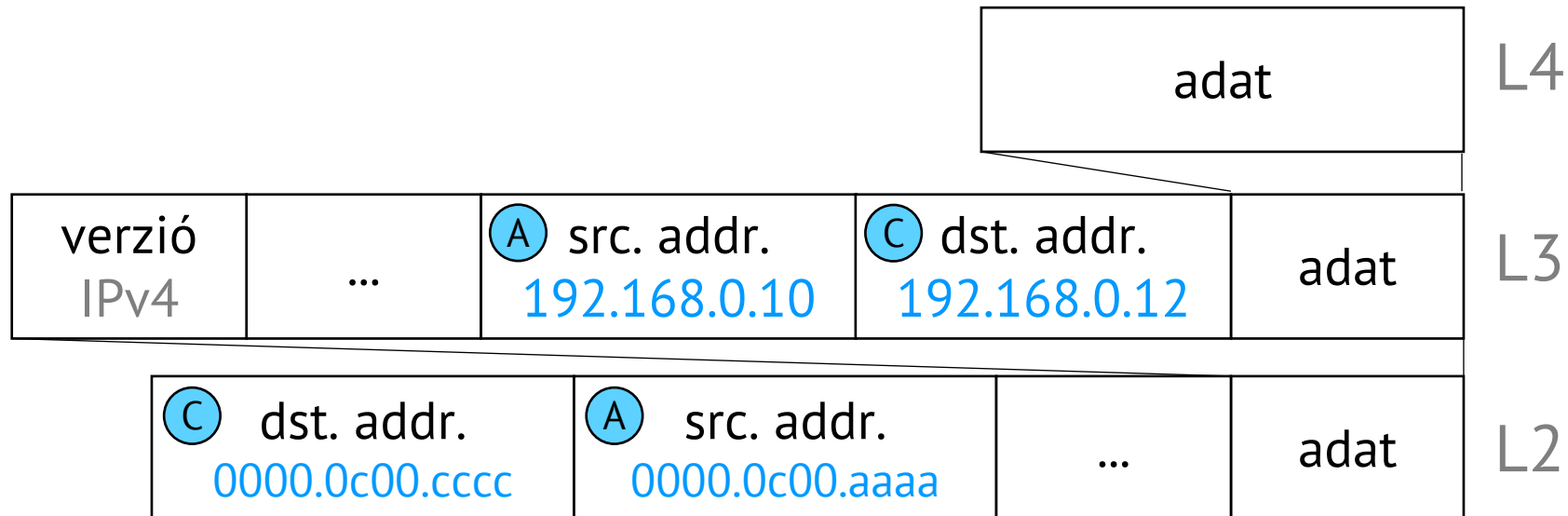


# 1. Példa – Csomagküldés alhálózaton belül

Az **A** eszköz adatot szeretne küldeni a **C** eszköznek.

Hogyan zajlik a folyamat?

3. A *data link layer* megkapja a *network layer* által előállított IP csomagot, és azt egy Ethernet frame-be teszi. Ehhez megadja a következő célállomás és a forrás MAC címét.

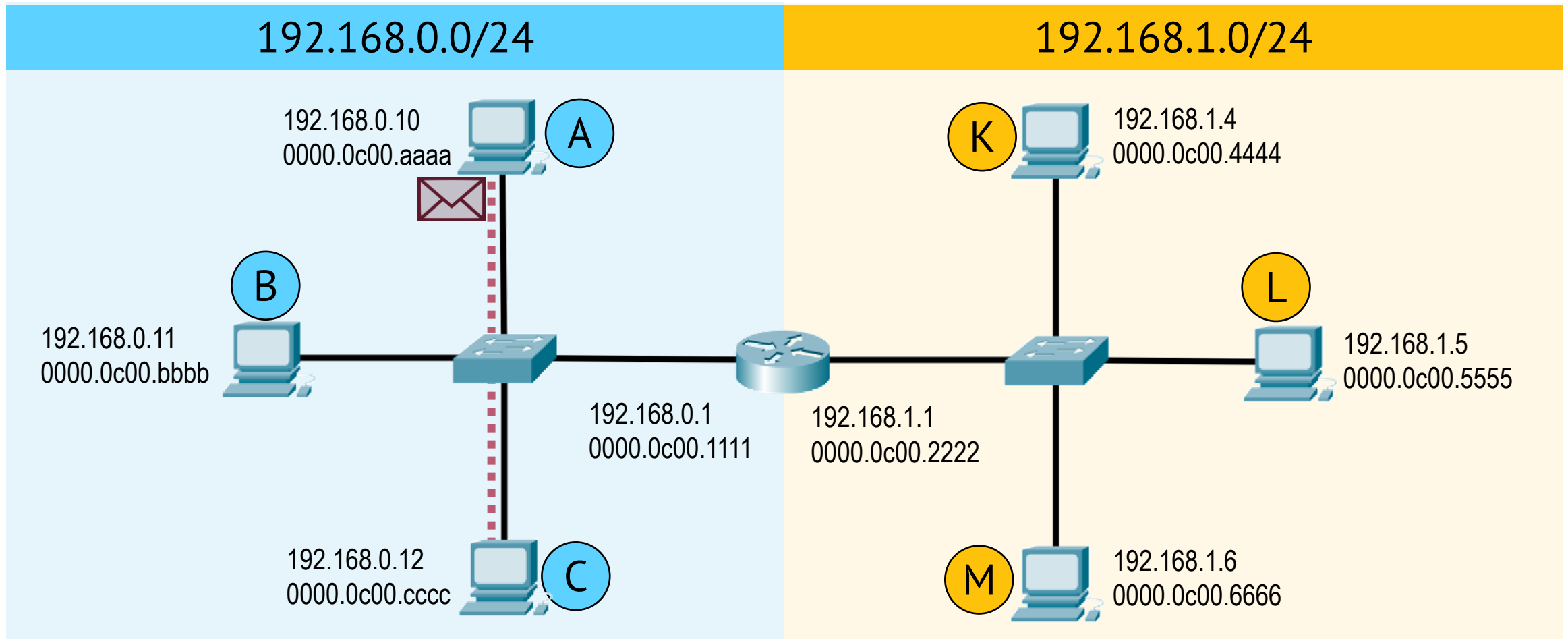


# 1. Példa – Csomagküldés alhálózaton belül

Az **A** eszköz adatot szeretne küldeni a **C** eszköznek.

Hogyan zajlik a folyamat?

4. A *physical layer* elvégzi az adat mozgását a hálózatban.



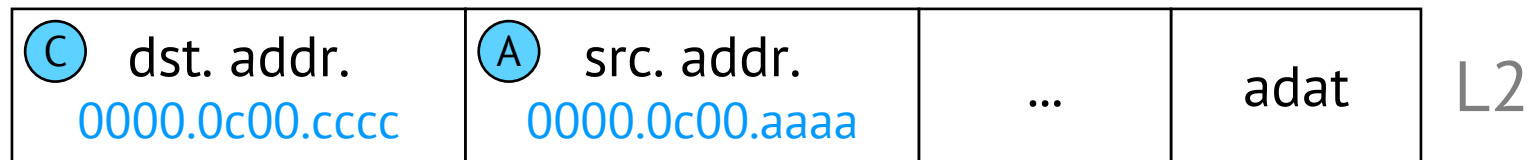


# 1. Példa – Csomagküldés alhálózaton belül

Az  eszköz adatot szeretne küldeni a  eszköznek.

Hogyan zajlik a folyamat?

5. A fogadó node-ban lévő *physical layer* visszaállítja az Ethernet frame-et, és feladja azt a *data link layer* réteg számára.



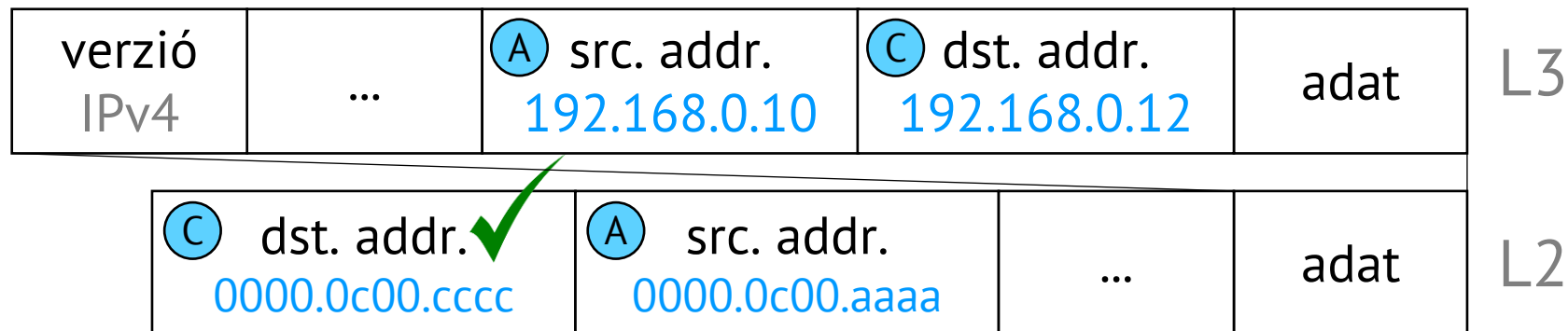
# 1. Példa – Csomagküldés alhálózaton belül

Az **A** eszköz adatot szeretne küldeni a **C** eszköznek.

Hogyan zajlik a folyamat?

**6.** A *data link layer* megvizsgálja, hogy ez az L2 interface volt-e a címzett (ennek a MAC címe szerepel-e a frame-ben).

Ha igen, akkor kicsomagolja belőle az IPv4 csomagot, és feladja azt a *network layer* számára. Ha nem, akkor figyelmen kívül hagyja a frame-et.



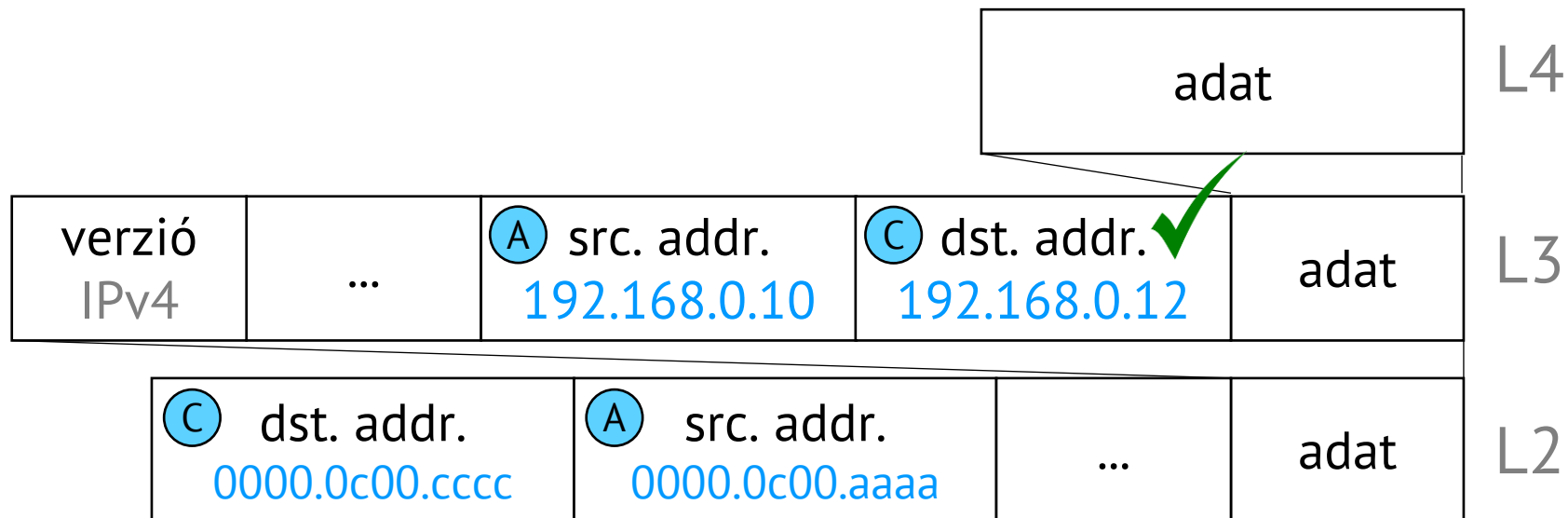
# 1. Példa – Csomagküldés alhálózaton belül

Az **A** eszköz adatot szeretne küldeni a **C** eszköznek.

Hogyan zajlik a folyamat?

**7.** A *network layer* megvizsgálja, hogy ez az L3 interface volt-e a címzett (ennek IPv4 címe szerepel a csomagban).

Ha igen, akkor kicsomagolja belőle az adatot, és azt feladja a *transport layer* számára.  
Ha nem, akkor megpróbálja a címzett felé továbbküldeni a csomagot.

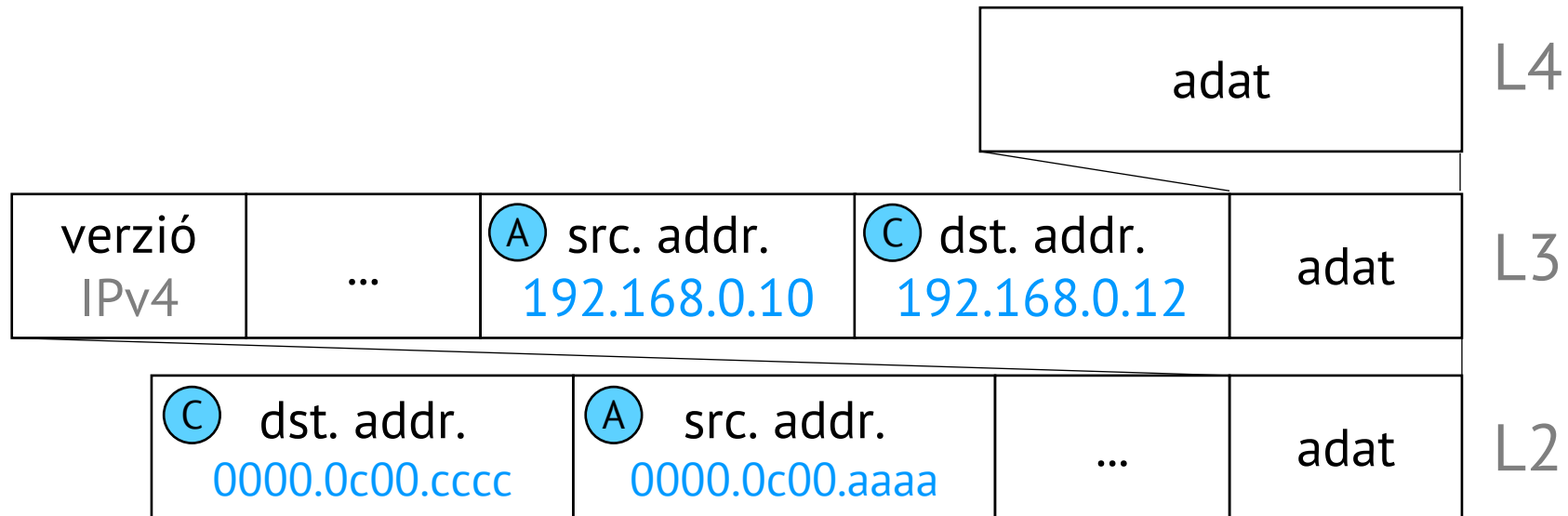


# 1. Példa – Csomagküldés alhálózaton belül

Az **A** eszköz adatot szeretne küldeni a **C** eszköznek.

Hogyan zajlik a folyamat?

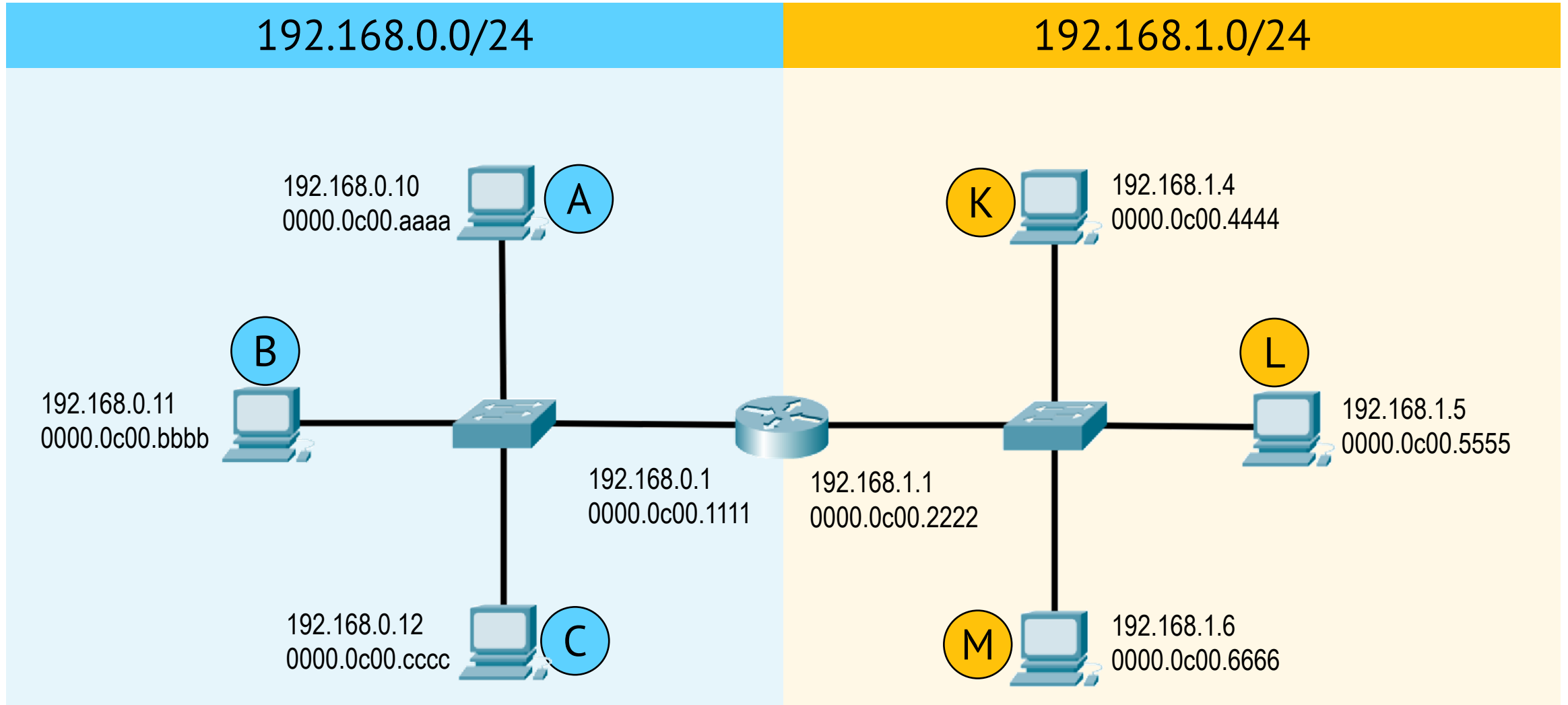
8. Végül a címzett node-ban a *transport layer* megkapja az adatot, és azt csinál vele, amit akar.



## 2. Példa – Csomagküldés alhálózatok között

Az **A** eszköz adatot szeretne küldeni a **K** eszköznek.

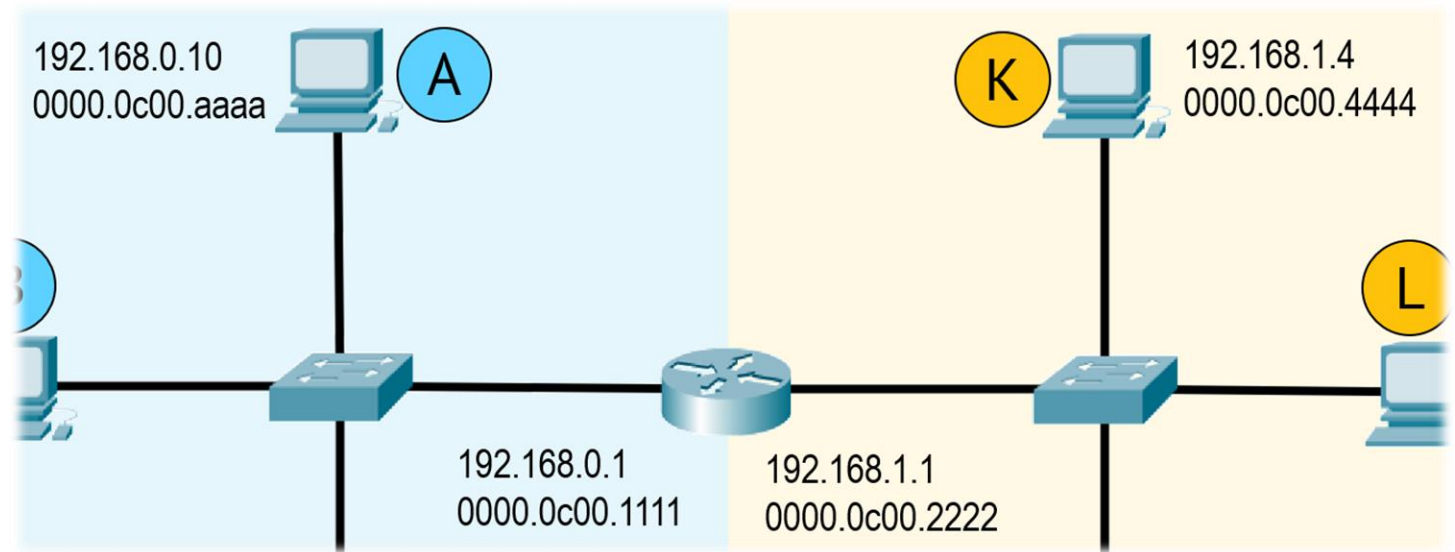
A hálózati struktúra:



## 2. Példa – Csomagküldés alhálózatok között

Az **A** eszköz adatot szeretne küldeni a **K** eszköznek.

- *Ki lesz a feladó L3 szinten?*  
**192.168.0.10**
- *Ki lesz a címzett L3 szinten?*  
**192.168.1.4**
- *Ki lesz a feladó L2 szinten?*  
**0000.0c00.aaaa**
- *Ki lesz a címzett L2 szinten?*




A feladó kiszámolja, hogy a címzett vele egy alhálózaton van-e → NEM  
A címzett eléréséhez ebben az alhálózatban lévő alkalmas node-ot kell keresni.  
A feladó megvizsgálja a saját routing tábláját. A routing tábla alapján a router elvileg alkalmas erre, tehát ennek az ebbéli alhálózatbéli L2 címe lesz a címzett:  
**0000.0c00.1111**

## 2. Példa – Csomagküldés alhálózatok között

Az  eszköz adatot szeretne küldeni a  eszköznek.

Hogyan zajlik a folyamat?

1. Az  eszközben a *transport layer* átadja az adatot a *network layer*-nek.

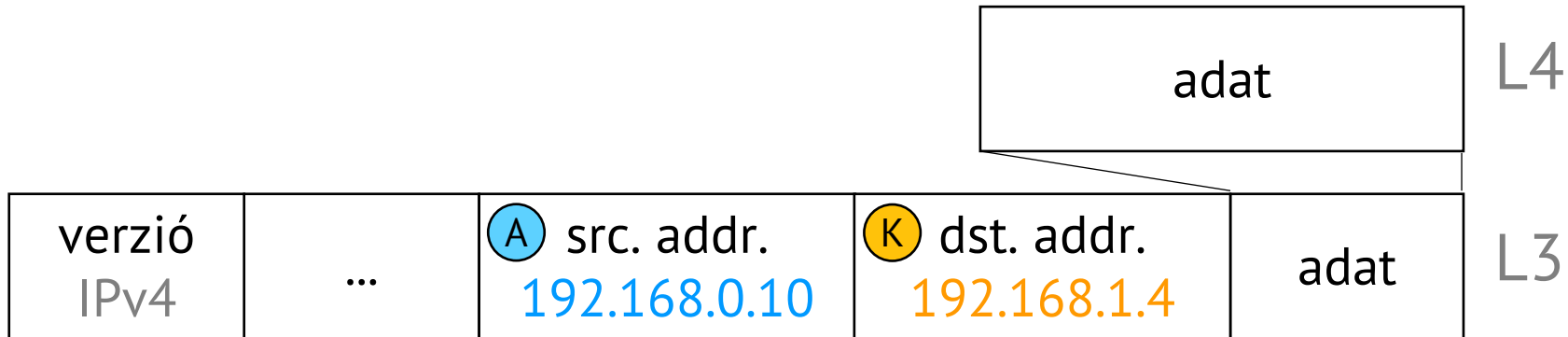


## 2. Példa – Csomagküldés alhálózatok között

Az **A** eszköz adatot szeretne küldeni a **K** eszköznek.

Hogyan zajlik a folyamat?

2. A *network layer* előállítja az IP csomagot, ehhez megadja a feladó és a címzett IPv4 címét, és a többi szükséges paramétert.



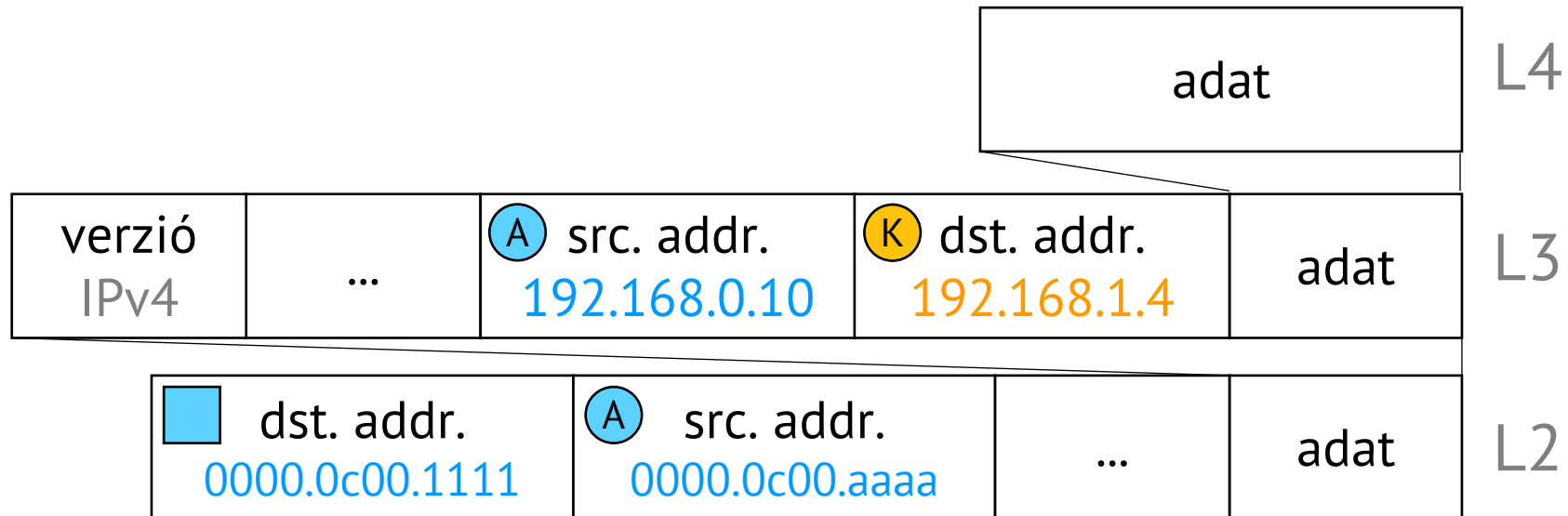


## 2. Példa – Csomagküldés alhálózatok között

Az  eszköz adatot szeretne küldeni a  eszköznek.

Hogyan zajlik a folyamat?

3. A *data link layer* megkapja a *network layer* által előállított IP csomagot, és azt egy Ethernet frame-be teszi. Ehhez megadja a következő célállomás és a forrás MAC címét.

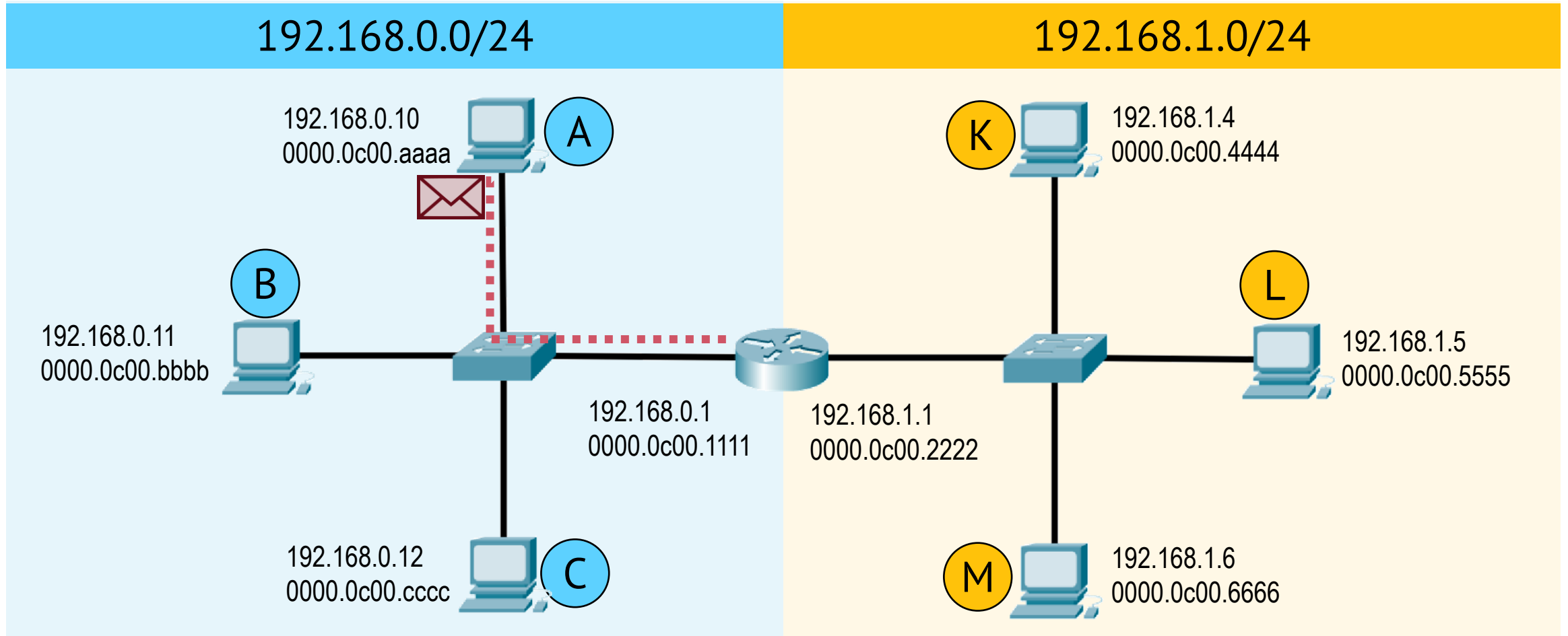


## 2. Példa – Csomagküldés alhálózatok között

Az **A** eszköz adatot szeretne küldeni a **K** eszköznek.

Hogyan zajlik a folyamat?

4. A *physical layer* elvégzi az adat mozgását a hálózatban.

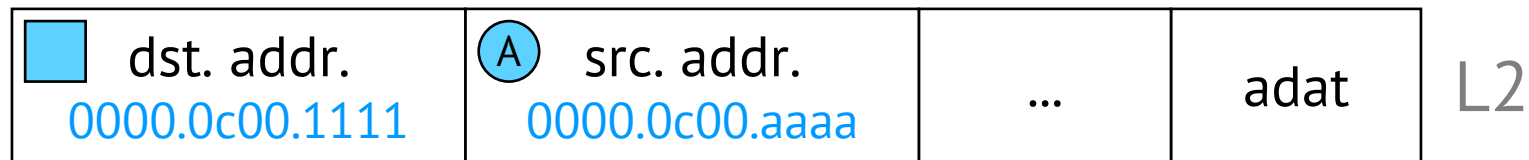


## 2. Példa – Csomagküldés alhálózatok között

Az  eszköz adatot szeretne küldeni a  eszköznek.

Hogyan zajlik a folyamat?

5. A fogadó node-ban lévő *physical layer* visszaállítja az Ethernet frame-et, és feladja azt a *data link layer* réteg számára.



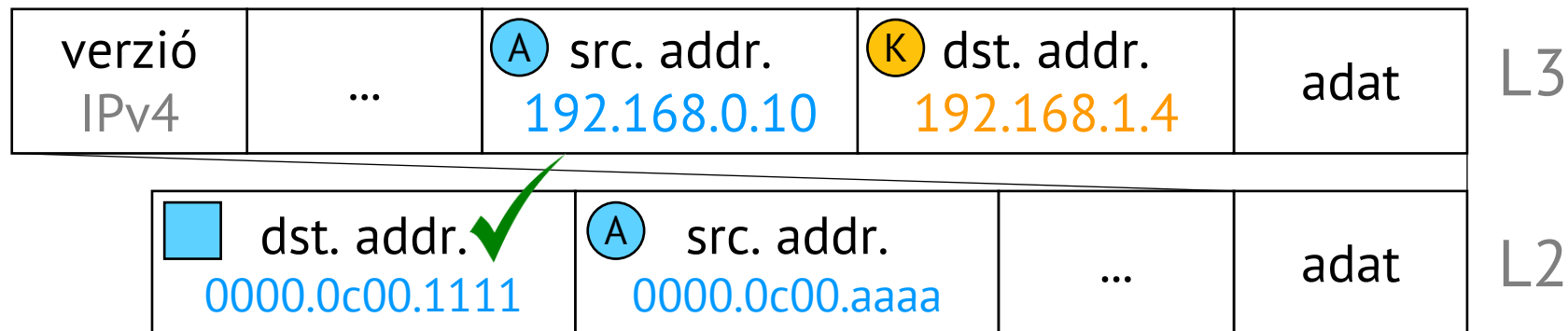
## 2. Példa – Csomagküldés alhálózatok között

Az  eszköz adatot szeretne küldeni a  eszköznek.

Hogyan zajlik a folyamat?

**6.** A *data link layer* megvizsgálja, hogy ez az L2 interface volt-e a címzett (ennek a MAC címe szerepel-e a frame-ben).

Ha igen, akkor kicsomagolja belőle az IPv4 csomagot, és feladja azt a *network layer* számára. Ha nem, akkor figyelmen kívül hagyja a frame-et.



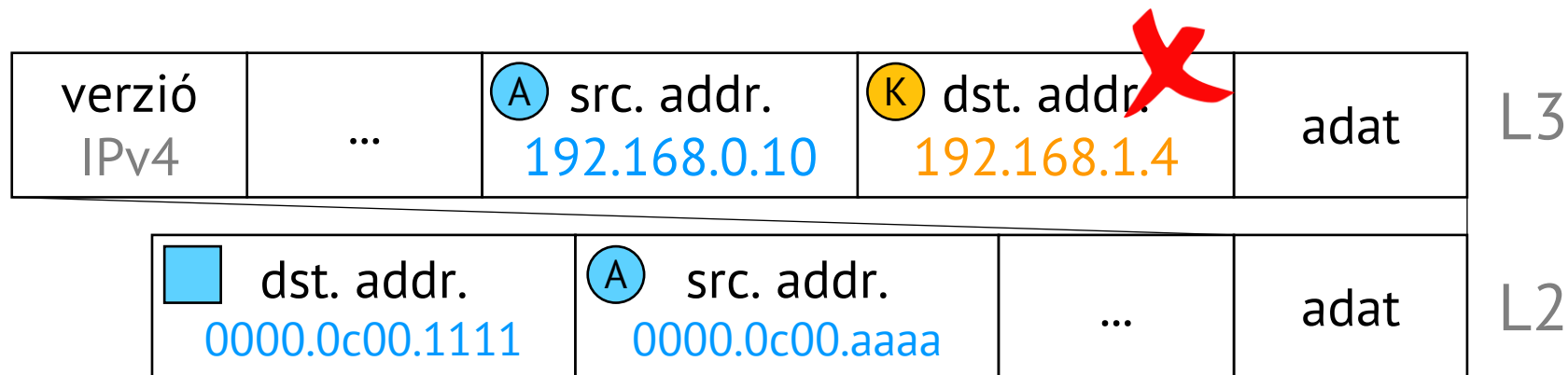
## 2. Példa – Csomagküldés alhálózatok között

Az  eszköz adatot szeretne küldeni a  eszköznek.

Hogyan zajlik a folyamat?

7. A *network layer* megvizsgálja, hogy ez az L3 interface volt-e a címzett (ennek IPv4 címe szerepel a csomagban).

Ha igen, akkor kicsomagolja belőle az adatot, és azt feladja a *transport layer* számára.  
Ha nem, akkor megpróbálja a címzett felé továbbküldeni a csomagot.





## 2. Példa – Csomagküldés alhálózatok között

Az  eszköz adatot szeretne küldeni a  eszköznek.

Hogyan zajlik a folyamat?

8. Mivel a cél címben szereplő 192.168.1.4 nem azonos a fogadó interfész címével (ami 192.168.0.1), ezért még nem értünk célba. A router megvizsgálja a routing táblában, hogy van-e a cél cím hálózatához tartozó next hop-ja, vagy ismer-e valakit, akinek van.

|                |     |   |   |      |    |
|----------------|-----|---|---|------|----|
| verzió<br>IPv4 | ... |  src. addr.<br>192.168.0.10 |  dst. addr.<br>192.168.1.4 | adat | L3 |
|----------------|-----|---|---|------|----|

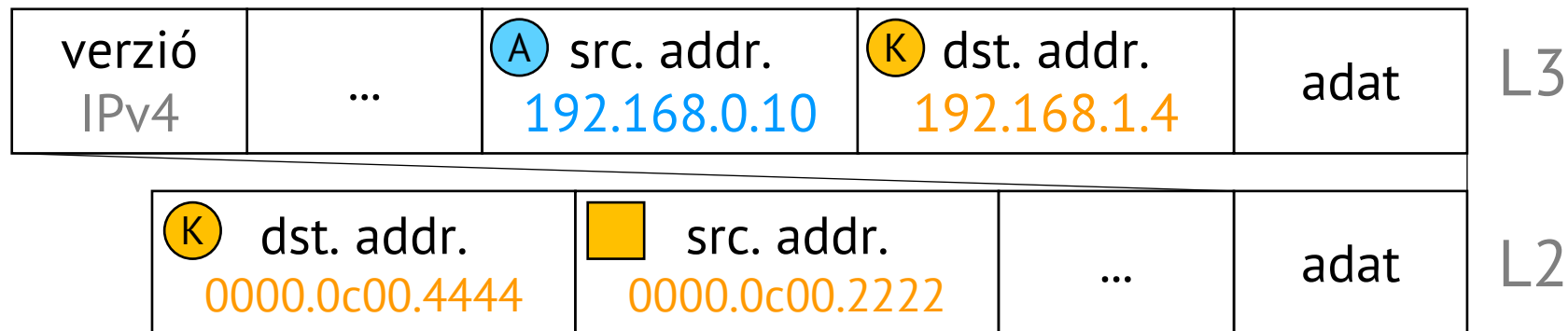
## 2. Példa – Csomagküldés alhálózatok között

Az  eszköz adatot szeretne küldeni a  eszköznek.

Hogyan zajlik a folyamat?

9. A router megtalálta, hogy a 192.168.1.1 IPv4 című interface-e a címzett alhálózatán van, tehát ezen ki tudja küldeni a csomagot, és akkor az célba jut.

A csomagban apró módosításokat végez (pl. csökkenti a TTL értéket) de az IPv4 címek változatlanok maradnak. A csomagot egy új Ethernet frame-be teszi, amit a ságra interface-en fog kiküldeni.

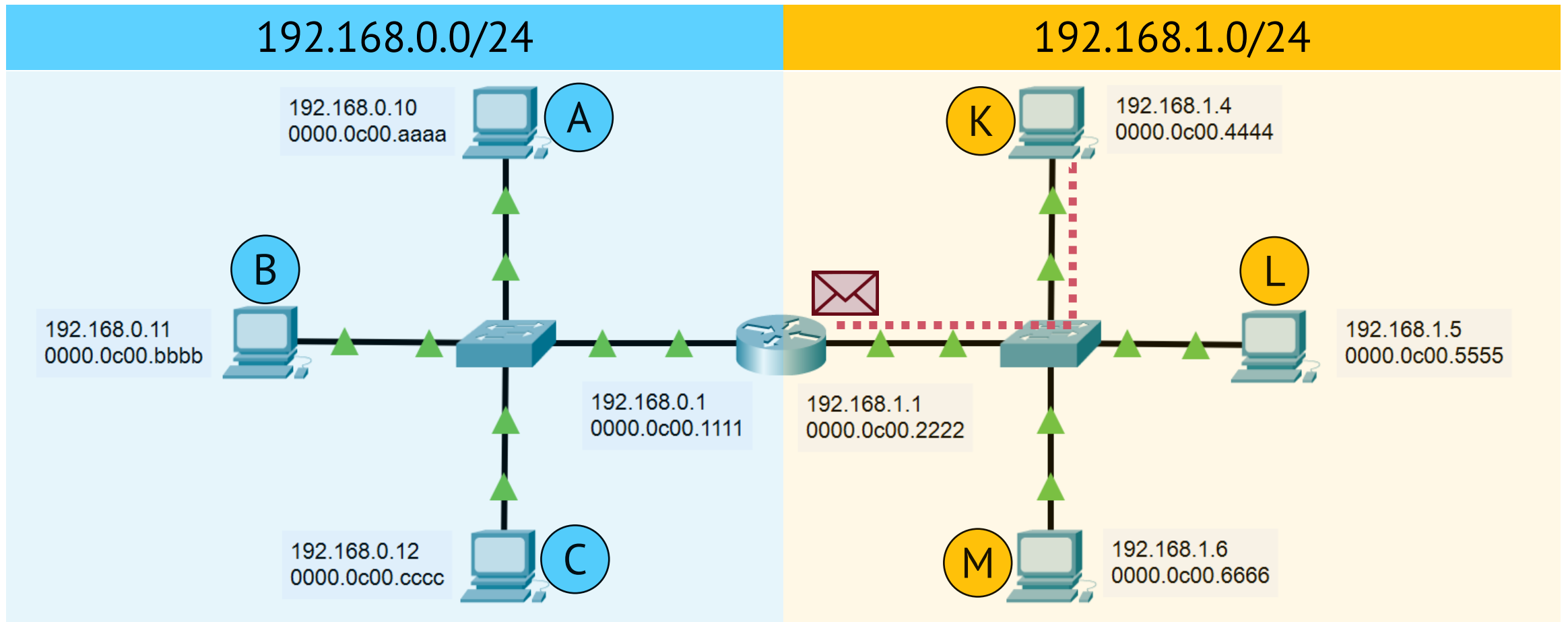


## 2. Példa – Csomagküldés alhálózatok között

Az **A** eszköz adatot szeretne küldeni a **K** eszköznek.

Hogyan zajlik a folyamat?

**10.** A *physical layer* elvégzi az adat mozgatóását a hálózatban.



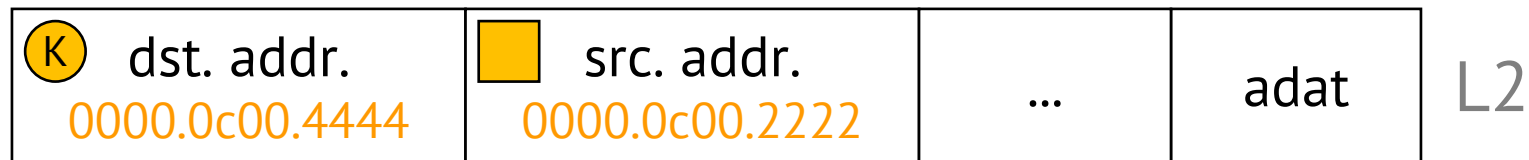


## 2. Példa – Csomagküldés alhálózatok között

Az  eszköz adatot szeretne küldeni a  eszköznek.

Hogyan zajlik a folyamat?

**11.** A fogadó node-ban lévő *physical layer* visszaállítja az Ethernet frame-et, és feladja azt a *data link layer* réteg számára.



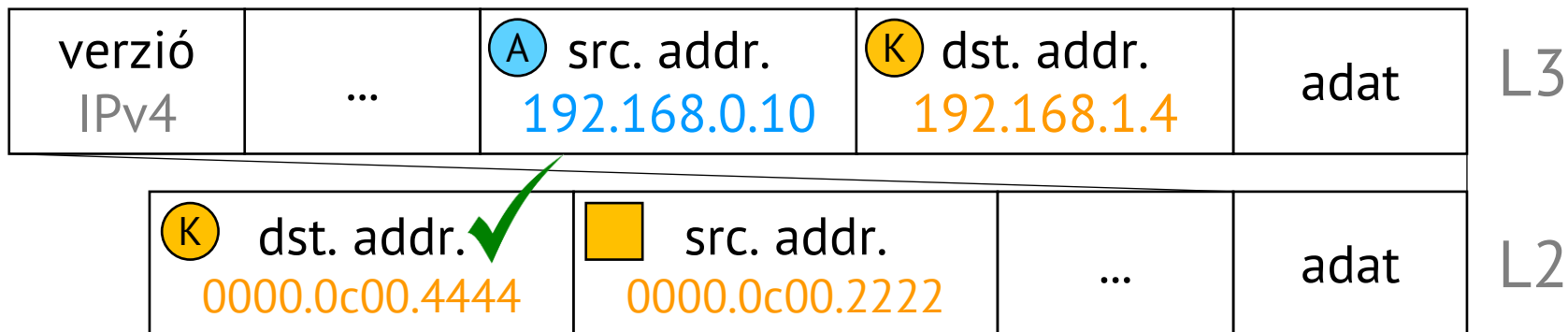
## 2. Példa – Csomagküldés alhálózatok között

Az  eszköz adatot szeretne küldeni a  eszköznek.

Hogyan zajlik a folyamat?

**12.** A *data link layer* megvizsgálja, hogy ez az L2 interface volt-e a címzett (ennek a MAC címe szerepel-e a frame-ben).

Ha igen, akkor kicsomagolja belőle az IPv4 csomagot, és feladja azt a *network layer* számára. Ha nem, akkor figyelmen kívül hagyja a frame-et.



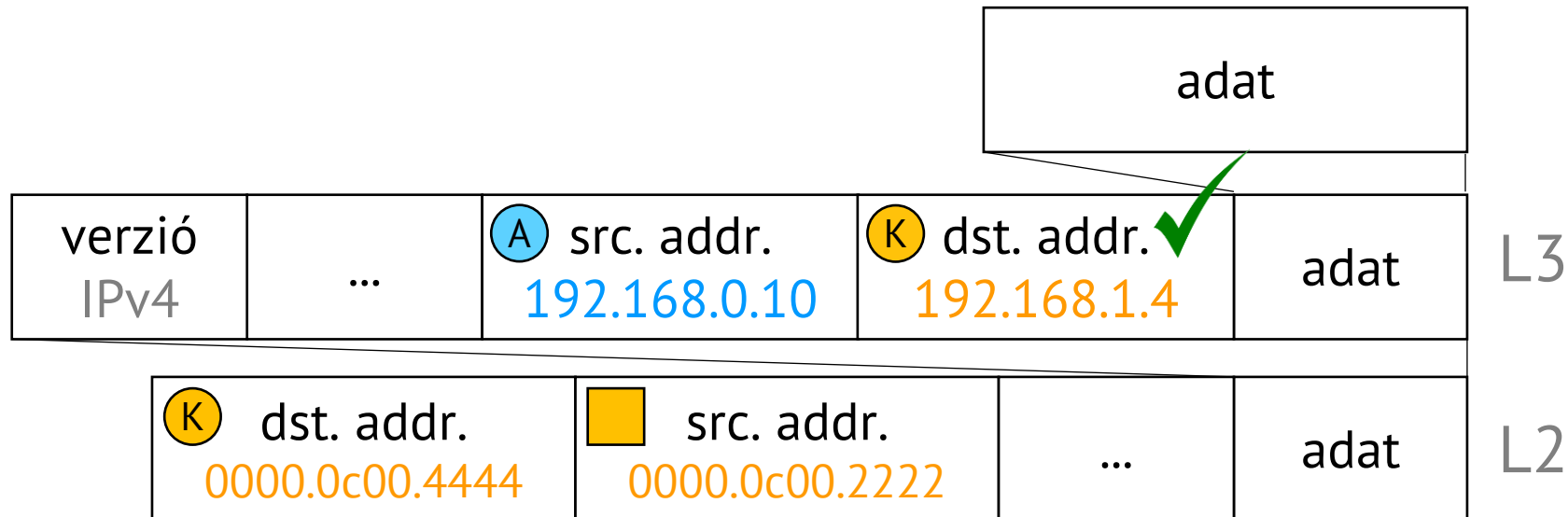
## 2. Példa – Csomagküldés alhálózatok között

Az  eszköz adatot szeretne küldeni a  eszköznek.

Hogyan zajlik a folyamat?

**13.** A *network layer* megvizsgálja, hogy ez az L3 interface volt-e a címzett (ennek IPv4 címe szerepel a csomagban).

Ha igen, akkor kicsomagolja belőle az adatot, és azt feladja a *transport layer* számára.  
Ha nem, akkor megpróbálja a címzett felé továbbküldeni a csomagot.

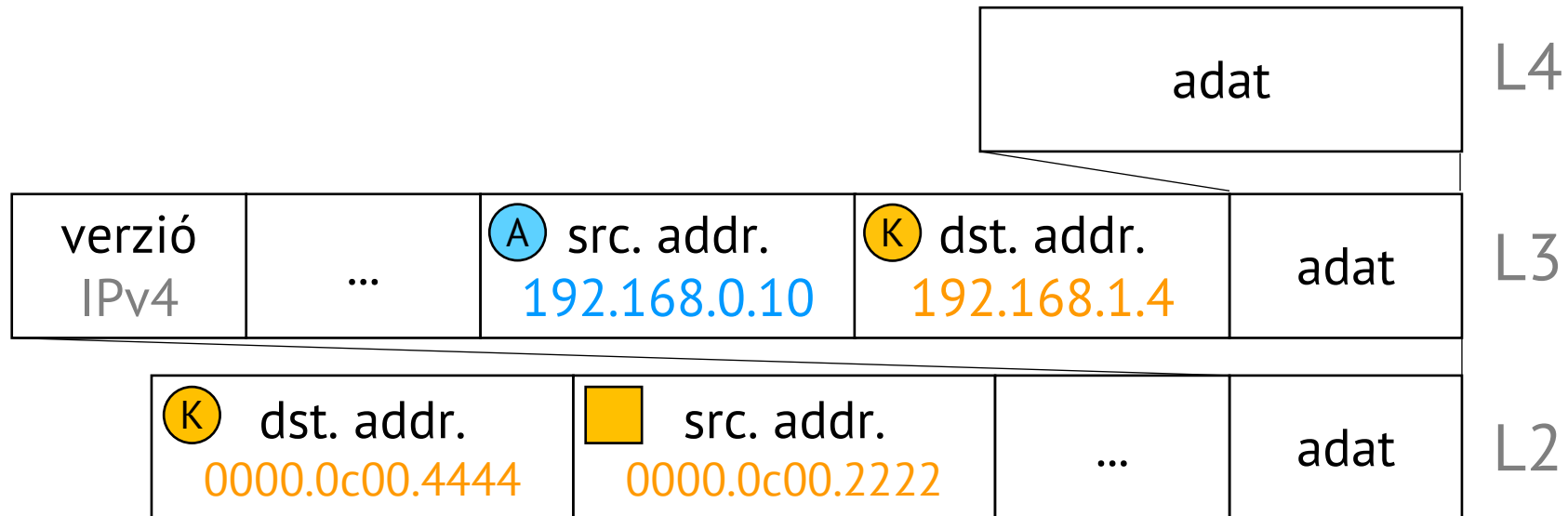


## 2. Példa – Csomagküldés alhálózatok között

Az  eszköz adatot szeretne küldeni a  eszköznek.

Hogyan zajlik a folyamat?

**14.** Végül a címzett node-ban a *transport layer* megkapja az adatot, és azt csinál vele, amit akar.



# A példák tanulsága

Az IP egy csomagkapcsolt átvitel. Vagyis:

- Az adatból egy IP csomagot képzünk,
- Kiválasztjuk a következő fizikai címzettet, és a csomagot egy Ethernet frame-be tesszük, amit elküldünk.
- A fogadott frame-ből kicsomagoljuk az IP csomagot, megnézzük, hogy mi vagyunk-e a címzett. Ha igen, akkor feldolgozzuk a csomagot, ha nem, akkor megpróbáljuk kitalálni, hogy kinek küldjük azt tovább (szintén Ethernet frame-be pakolva).

# A példák tanulsága

Az adatátvitelhez tudnunk kell

- a feladó IPv4 címét,
- a címzett IPv4 címét,
- mi a következő lépés, melyik közvetlen fizikai interfésznek küldjük a frame-et,
- és hogy mi ennek a fizikai interfésznek a MAC címe.

# A valóság

Előfordulhat, hogy

- csak a következő fizikai eszköz IPv4 címét tudjuk, de a MAC címét nem.
- csak a címzett MAC címét tudjuk, az IPv4 címét nem.
- fogalmunk sincs arról, mi a saját IPv4 címünk.
- fogalmunk sincs arról, hogy mi a hálózati struktúra; ki a router.
- fogalmunk sincs arról, hogy mi a hálózati struktúra; mi a CIDR leírás.
- két (vagy több) eszköz is azt hiszi, hogy ugyanaz az IPv4 cím az övé.

# #03/5 – Összefoglalás

**Fogalmak**     Routing tábla

**Eszközök**     Router

**Folyamat**     Milyen IPv4 és MAC címek találhatók abban a csomagban, amit egyik helyről a másikra továbbítunk?

Miként döntjük el, hogy egy alhálózaton van-e velünk a címzett?



# **#03/6 – Address Resolution Protocol**



# Address Resolution Protocol

Feladatok:

- IPv4 címhez MAC címet adni.
- MAC címhez IPv4 címet adni.
- Akkor is adjon „jó” választ, ha a hálózat csak logikailag tartozik egybe.
- Segítsen észrevenni, ha többször van ugyanaz az IP cím a hálózaton.

# ARP frame struktúra

Az ARP frame a következő egységekből áll:

|                 |                 |                |                |                |     |     |     |     |
|-----------------|-----------------|----------------|----------------|----------------|-----|-----|-----|-----|
| HTYPE<br>2 byte | PTYPE<br>2 byte | HLEN<br>1 byte | PLEN<br>1 byte | OPER<br>2 byte | SHA | SPA | THA | TPA |
|-----------------|-----------------|----------------|----------------|----------------|-----|-----|-----|-----|

- HTYPE hardware type – a fizikai hálózat típusa (pl. ethernet)
- PTYPE protocol type – a protokol típusa (pl. IPv4, IPv6, ...)
- HLEN hardware length – a fizikai címek hossza (pl. MAC cím hossza)
- PLEN protocol length – a protokol címek hossza (pl. IPv4 cím hossza)
- OPER operation – az ARP művelet típusa (Request, Reply, ...)
- SHA, SPA sender hw, protocol address – küldő fizikai és protokol címei
- THA, TPA target hw, protocol address – címzett fizikai és protokol címei

# ARP üzenet enkapszulációja

## ARP frame

|                 |                 |                |                |                |     |     |     |     |
|-----------------|-----------------|----------------|----------------|----------------|-----|-----|-----|-----|
| HTYPE<br>2 byte | PTYPE<br>2 byte | HLEN<br>1 byte | PLEN<br>1 byte | OPER<br>2 byte | SHA | SPA | THA | TPA |
|-----------------|-----------------|----------------|----------------|----------------|-----|-----|-----|-----|

## Ethernet frame

|                    |               |              |              |                    |      |               |               |
|--------------------|---------------|--------------|--------------|--------------------|------|---------------|---------------|
| preamble<br>7 byte | SFD<br>1 byte | DA<br>6 byte | SA<br>6 byte | len/type<br>2 byte | data | pad<br>n byte | CRC<br>4 byte |
|--------------------|---------------|--------------|--------------|--------------------|------|---------------|---------------|

Az ethernet frame típus mezője fogja megmondani, hogy a benne lévő adat ARP frame.

Az ARP frame-ben lévő OPER mező mondja meg, hogy az ARP üzenet kérés, válasz, RARP kérés vagy RARP válasz...

# ARP folyamat

**Cél:** IPv4 cím ismeretében megtudni a hozzá tartozó MAC címet.

Kétlépcsős folyamat:

1. ARP request
2. ARP reply



# ARP folyamat

**Cél:** IPv4 cím ismeretében megtudni a hozzá tartozó MAC címet.

Kétlépcsős folyamat:

1. ARP request → Mondja meg az ITK, hogy mi az ő címe!
2. ARP reply ← 1083 Budapest, Práter utca 50/A

A folyamat a kérdező és az összes többi, az alhálózatba tartozó eszköz közt zajlik le.

# ARP folyamat

**1. ARP request** → „Mondja meg az ITK, hogy mi az ő címe!”

A kérdező szeretné megtudni az IPv4 címhez tartozó MAC címet.

Broadcast Ethernet üzenet:

- feladó IP címe
- feladó MAC címe
- címzett IP címe (a keresett IPv4 cím)
- címzett MAC címe (csupa nulla, mert nem tudjuk; pont ez a kérdés)

# ARP folyamat

## 2. ARP reply ← „ 1083 Budapest, Práter utca 50/A”

Az összes hálózati eszköz megkapja a broadcast üzenetet, és megvizsgálja az abban lévő cél IP címet. Ha az nem egyezik a saját címével, akkor nem válaszol.

Ha viszont a saját címét látja, akkor reply üzenetet küld:

Unicast Ethernet üzenet:

- feladó IP címe (a keresett IPv4 cím)
- feladó MAC címe (a feladó megmondja a keresett MAC címet)
- címzett IP címe (a kérdező IPv4 címe)
- címzett MAC címe (a kérdező MAC címe)



# ARP folyamat

Előfordulhat, hogy egy ARP kérésre senkitől sem érkezik válasz.  
Ekkor az IP cím valószínűleg nincs kiosztva  
(vagy aki birtokolja, nincs beszédes kedvében).

Meg lehet próbálni újra,  
vagy bele lehet törődni, hogy ez nem sikerült.

# Gratuitous ARP

A saját IP címemre adok ki ARP kérést.

Ha más is válaszol, akkor baj van: ő is azt hiszi, hogy ez az ő címe, tehát ugyanaz a cím több interfésznek is ki van osztva. Ezt nem szabad!

Sokszor az operációs rendszerek is ezzel kezdik a cím használatát; egy vagy egy pár ilyen kérdést intéznek a többi eszköz felé. Ha választ kapnak, akkor nem kezdik használni a címet.

# ARP cache

Az ARP kérésekre jövő válaszokat egy bizonyos ideig célszerű megjegyezni.

**Vagyis:** nem kell minden kommunikáció előtt ARP kérést intézni, mert egy lokális ARP táblában fel tudjuk jegyezni az IP címhez tartozó MAC címet.

Ekkor, ha kíváncsiak vagyunk egy IP-hez tartozó MAC címre, akkor először az ARP táblában nézünk utána, és csak akkor indítunk ARP broadcast üzenetet, ha nem találtuk meg a saját táblánkban.

| Internet Address | Physical Address  | Type    |
|------------------|-------------------|---------|
| 192.168.0.1      | 00:00:0C:11:11:11 | dynamic |
| 192.168.0.10     | 00:00:0C:AA:AA:AA | dynamic |
| 192.168.0.255    | FF:FF:FF:FF:FF:FF | static  |



# ARP cache

## Dinamikus ARP bejegyzés

- ARP kérések során kerül bele a táblába
- Lejárati ideje van (néhány tíz másodperctől több percig)
- „Bárhonnan” belekerülhet

## Statikus ARP bejegyzés

- Kézzel írjuk bele (vagy a hálózat konfigurálását végző folyamat írja be)
- Örökre benne marad a táblában (addig van benne, amíg kézzel ki nem vesszük)
- Megbízhatóbb forrásból került be

| Internet Address | Physical Address  | Type    |
|------------------|-------------------|---------|
| 192.168.0.1      | 00:00:0C:11:11:11 | dynamic |
| 192.168.0.10     | 00:00:0C:AA:AA:AA | dynamic |
| 192.168.0.255    | FF:FF:FF:FF:FF:FF | static  |

# ARP folyamat – kiegészítés

**2. ARP reply** ← „1083 Budapest, Práter utca 50/A”

Az összes hálózati eszköz megkapja a broadcast üzenetet, és megvizsgálja az abban lévő cél IP címet. Ha az nem egyezik a saját címével, akkor nem válaszol.

**Ez viszont nem jelenti azt, hogy ne csinálna semmit sem!**

Ha nem is ő volt a címzett, azért hasznára válhat az üzenet: a requestben szereplő kérdező IP címet és kérdező MAC címet elteheti a saját ARP cache táblájába.

Ezzel az ügyes hallgatózással megúszott egy jövőbeni ARP kérést → spórolás.

# ARP cache poisoning

Egy rosszindulatú támadó hamis ARP adatokat szórhat szét; ezek a hamis adatok bekerülhetnek az ARP cache táblába, „megmérgezzhetik azt”.

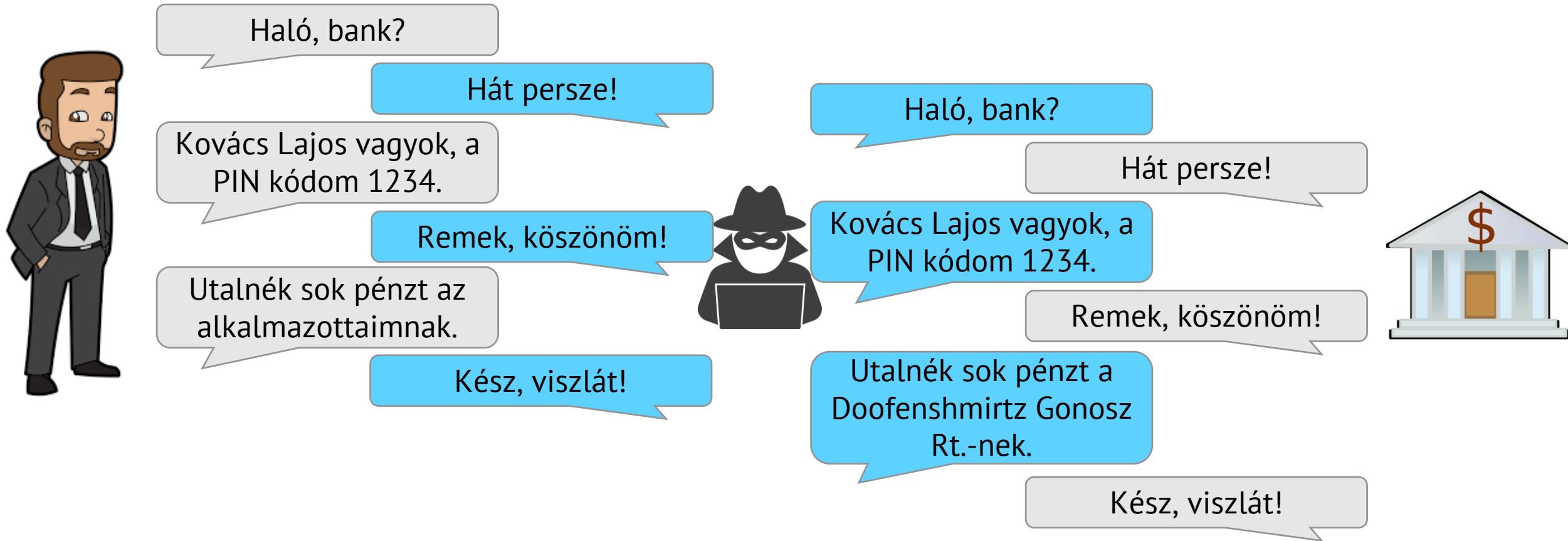
- eltérítheti a forgalmat
- túltöltheti az ARP táblát (DoS – Denial of Service)

**Védekezés módja, hogy a kritikus kapcsolatokat bevasaljuk az ARP táblába.  
(statikus bejegyzés)**

# Man-in-the-Middle Attack



A Man-in-the-middle támadás lényege, hogy észrevétlenül befúrjuk magunkat a két kommunikáló fél közé:



# Man-in-the-Middle Attack

Az ARP cache poisoning lehetőséget ad arra, hogy Man-in-the-middle támadást hajtsunk végre. Ehhez úgy kell mérgezni az **A** és **C** eszközök gyorsítótárát, hogy azt higgyék, hogy egymással beszélnek.

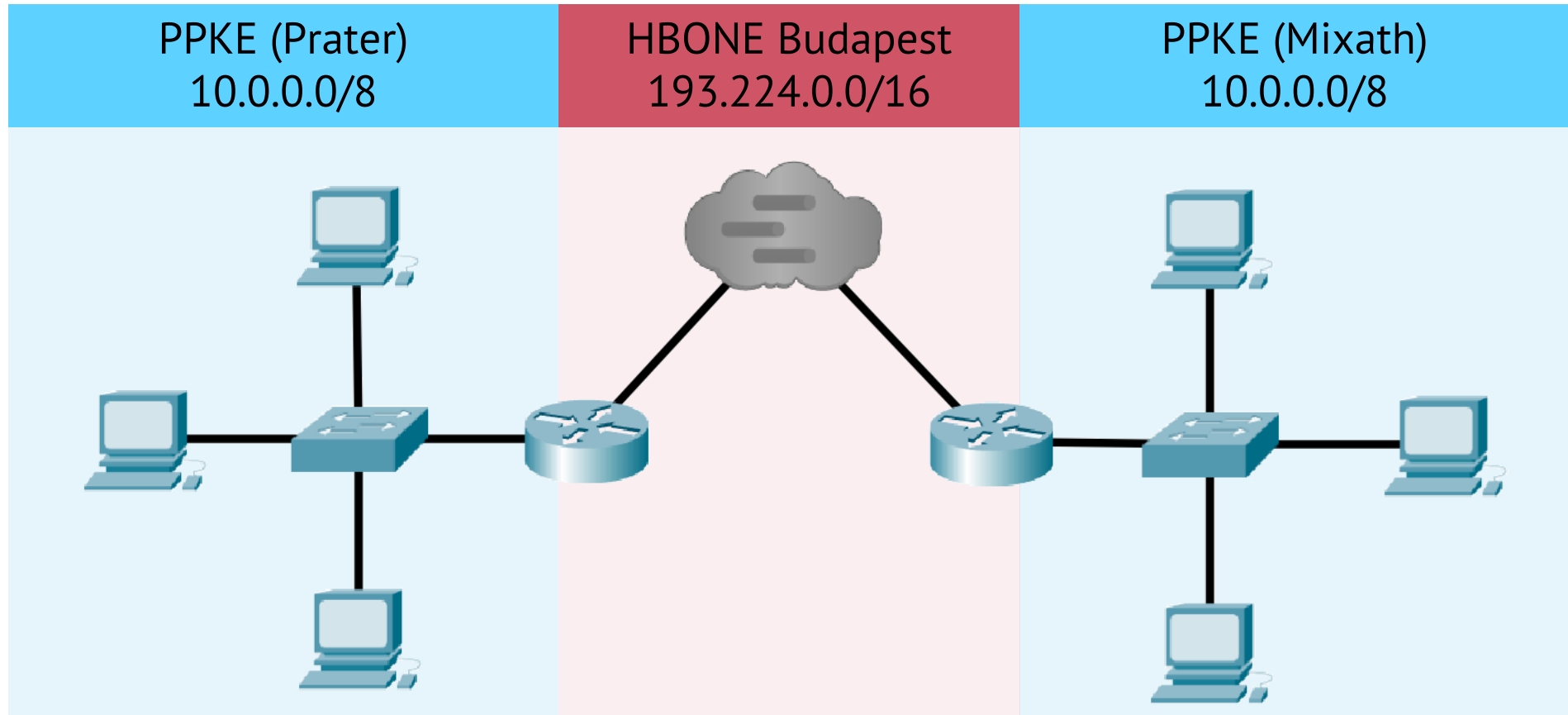


Mivel viszonylag egyszerű ilyen tenni, nagyon fontos, hogy a felsőbb rétegeken megfelelő titkosítást, kódolást használjunk az adatok védelme érdekében!



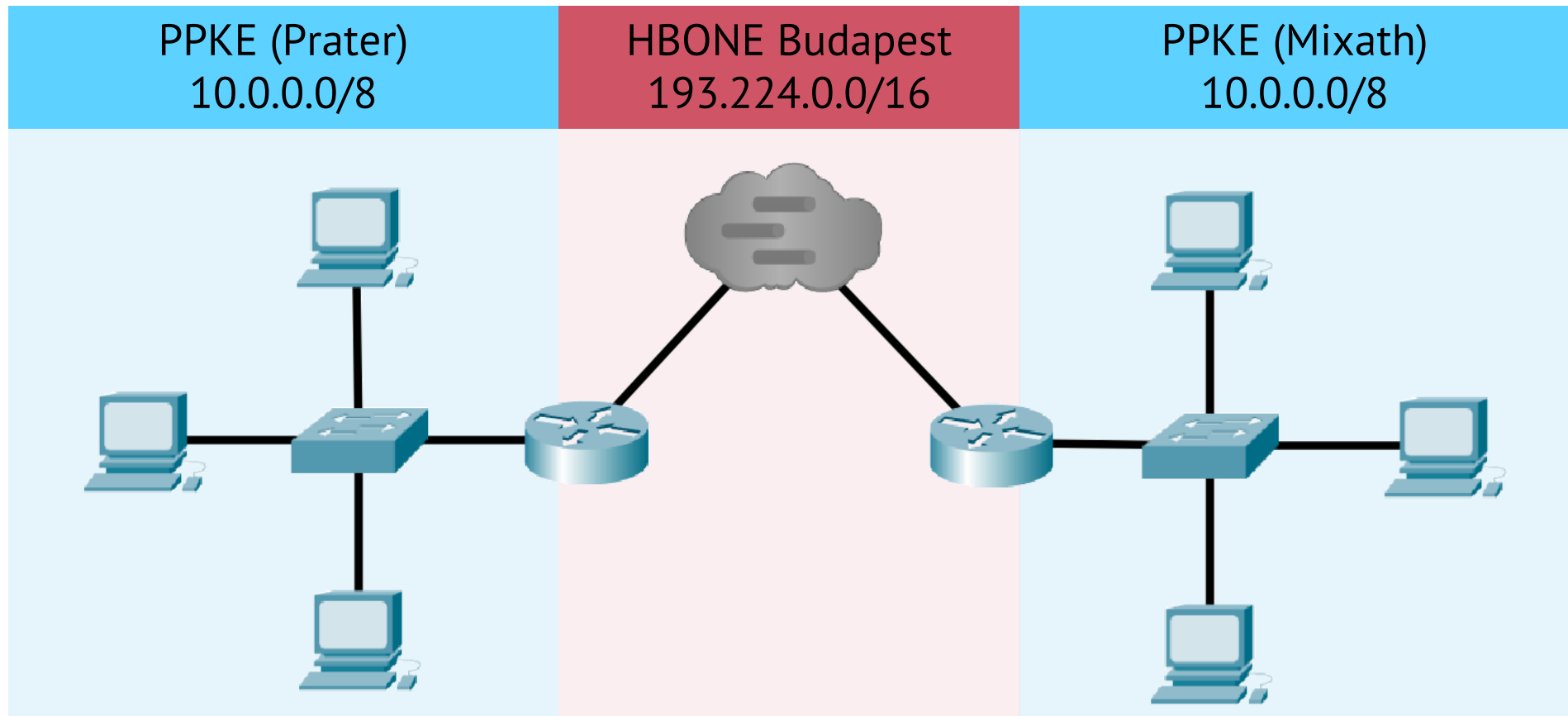
# Proxy ARP

Előfordulhat, hogy egy távoli eszköz is logikailag a helyi alhálózathoz tartozik.



# Proxy ARP

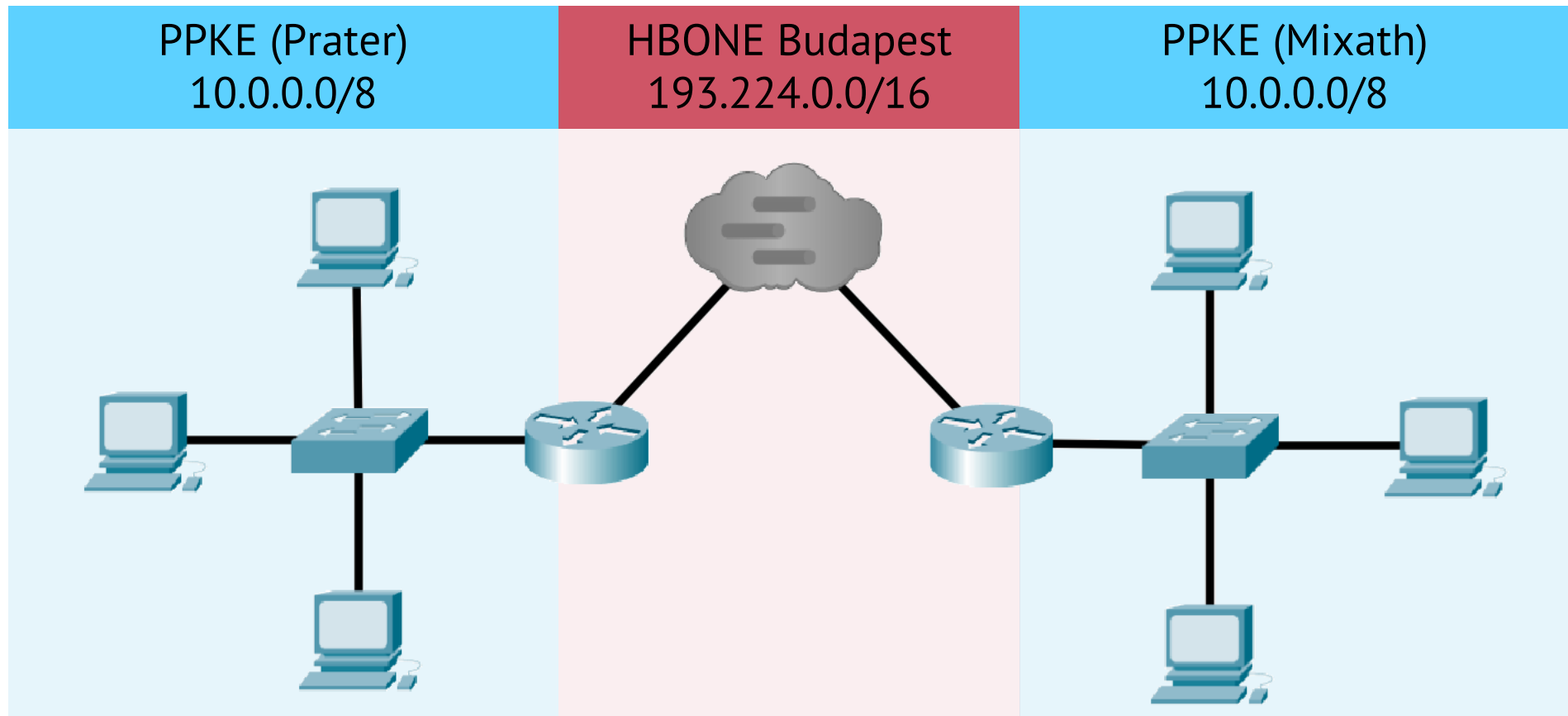
Előfordulhat, hogy egy távoli eszköz is logikailag a helyi alhálózathoz tartozik. A broadcast üzenet nem megy át a routeren, tehát a „*who has X?*” ARP request üzenetek elvesznének; nem látjuk az ITK-ról a Mikszáth téri eszközöket.



# Proxy ARP

Előfordulhat, hogy egy távoli eszköz is logikailag a helyi alhálózathoz tartozik.

**Megoldás:** A *‘kék-piros’* router tudja, hogy rajta keresztül elérhető a Mikszáth hálózat, és úgy tesz, mintha ő lenne az ottani eszköz: a saját MAC címét válaszolja az ARP kérésre. Az IP majd úgy is rendet tesz.



# RARP

Mi van akkor, ha nem tudjuk a saját IP címünket?

**Reverse ARP** – lényegében egy ARP folyamat, csak fordított adattal.

## ***1. RARP request***

- broadcast üzenet
- csak a feladó MAC címe szerepel benne (az IP címek nullák)

## ***2. RARP reply***

- unicast üzenet
- egy „felhatalmazott” eszköztől jön (nem ám akárki osztogatja a címeket)
- benne van a kiosztott IP cím (és a válaszadó IP és MAC címe)
- nincs benne a netmask, default gateway

# #03/6 – Összefoglalás

|                 |                          |
|-----------------|--------------------------|
| <b>Fogalmak</b> | ARP                      |
|                 | Man-in-the-Middle Attack |
|                 | ARP cache poisoning      |
| <b>Folyamat</b> | ARP folyamat             |
|                 | RARP folyamat            |

# VÉGE



## PÁZMÁNY

Pázmány Péter Katolikus Egyetem  
**Információs Technológiai és Bionikai Kar**