Számítógép-hálózatok
(2. rész)

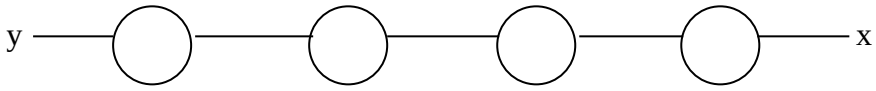
(Dr. Almási Béla előadása alapján 2000/2001 1.félév)

Előadás fóliái: http://it.math.klte.hu\user\almasi\cn

Hálózati réteg

Szolgáltatási típusok:
1. Kapcsolatorientált (összeköttetés alapú)
2. Kapcsolatmentes

A hálózati rétegben valósul meg az alhálózatok közötti összeköttetés.



(Datagram átvitel) .

	Kapcsolatorientált	Kapcsolat nélküli
Alhálózatok felé mutatott igény:	Kapcsolatorientált	Bármely
Címzés:	VC címekkel	Minden információban ott kell legyen a célazonosító címe.
Routing (forgalomirányítás):	VC kialakításakor történik	Bármely csomag külön forgalomirányításra kerül.
Routing hiba hogyan befolyásolja a rendszer működését?	Leáll	Automatikusan helyreáll a kommunikáció (keres másik utat)

Az Internet protokoll – hálózati címzés

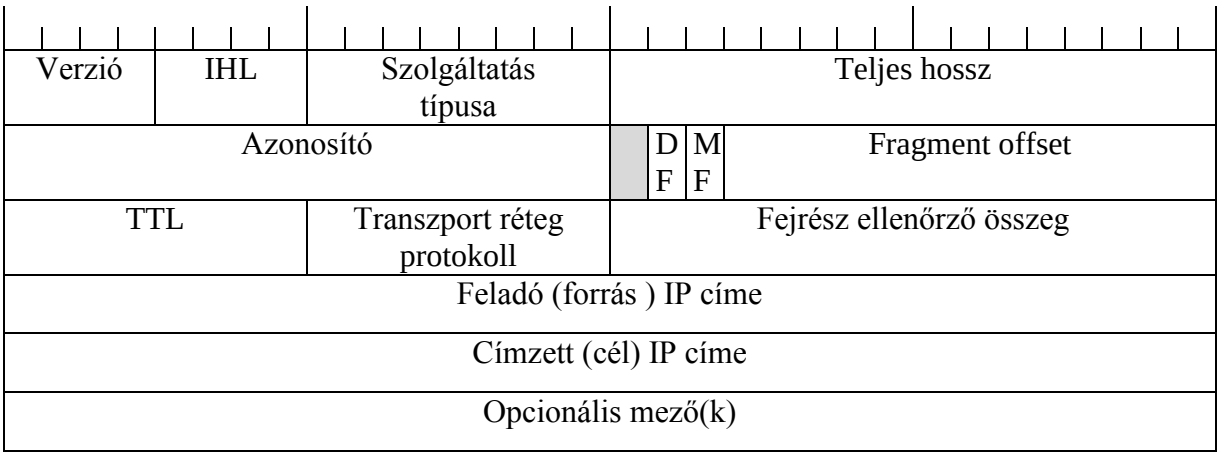
Fizikai címek, hálózati címek.

Miért van szükség hálózati címekre?
Miért nem elegendőek a fizikai címek használata?
- A fizikai címek elhelyezkedése strukturálatlan. (Ethernet kártyánál a sorszám első része a gyártó azonosítója, atöbbi azon belüli sorszám. – A forgalomirányítás szempontjából strukturálatlan.)
- Útvonalválasztást strukturálatlan címrendszerrel lehetetlen megoldani.
- A fizikai cím csak egy alhálózatba kapcsolt csomópontok kommunikációjához megfelelő.
- Szükség van egy másik, strukturált címrendszerre: a hálózati címekre.

Az IP hálózati protokoll

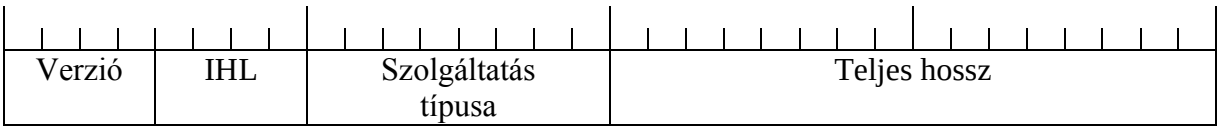
IP (Internet Protocol) RFC 791
- A TCP/IP referenciamodell hálózati réteg protokollja.
- Széles körben használt, az Internet alapeleme.
- Legfontosabb jellemzői:
  - IP fejrész szerkezete.
    » 32 bites szavakból áll
    » Minimum 5, maximum 15 szó hosszú.
  - IP címzés, címosztályok.
  - Darabolás (fragment) támogatás.

Internet csomag fejrész szerkezete



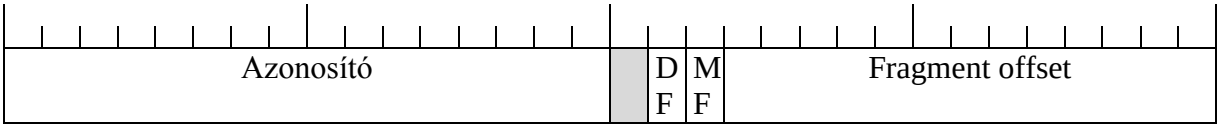
Az első öt szó kötelező mező. (Még lehetnek opcionálisak is.)

1. Az első szó tartalma:



Az első szó tartalma – általános információk:
- 4 bit: Verziószám (Ipv4).
- 4 bit: IP fejrész hossza (szavakban mérve).
- 8 bit: Szolgáltatás típusa (pl. hang vagy fájl átvitel).
- 16 bit: Teljes csomaghossz (bájtokban mérve).

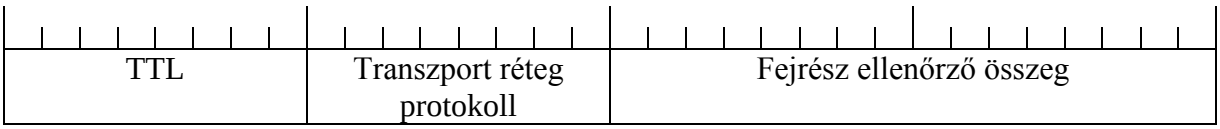
2. A második szó tartalma



A második szó tartalma – darabolás (fragment) adatai (a feldarabolt sorozaté):
- 16bit: Azonosító, a fragment sorozat azonosítója.
- 1 bit: Nem használt.

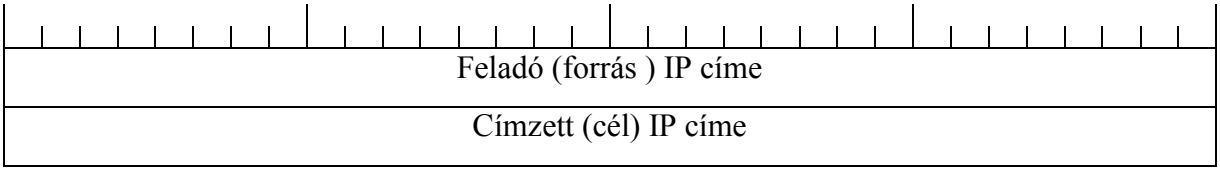
- 1 bit: DF – nem darabolható (pl. boot program).
- 1 bit: MF – további fragmentek léteznek (ebből tudja, hogy fel van darabolva).
- 13 bit: Fragment offset (a fragment helye a sorozatban – hányadik darab a sorozatban).

3. A harmadik szó adatai



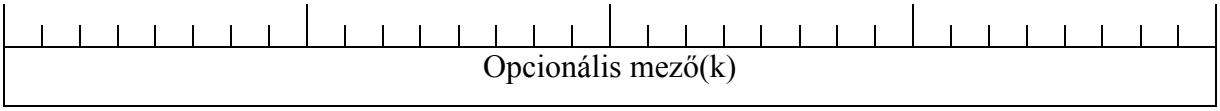
A harmadik szó adatai – általános információk:
- 8 bit: TTL a csomag „hátralevő életidejének” jelzése. (Forgalomirányítási hiba esetén körbe-körbe mehet a csomag. Azért, hogy ez ne terhelje a hálót, specifikálják az élettartamát. Értéke 255-ről indul, s minden egyes forgalomirányító ezt eggyel csökkenti.)
- 8 bit: Transzport rétegbeli protokoll azonosítója. (Innen tudja a vételi oldalon, hogy kinek kell továbbküldeni.)
- 16 bit: A fejrész ellenőrző összege.

4, 5. A negyedik, ötödik szó adatai



A negyedik, ötödik szó adatai – címzések:
- 32 bit: A „forrás” IP címe.
- 32 bit: A „cél” IP címe.

6. A hatodik szótól



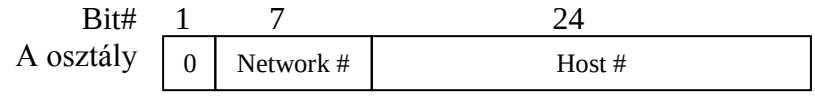
A hatodik szótól – 32 bites opcionális információk, pl.:
- Security – Védelmi opció. (pl. a Pentagonból származó csomag ne menjen keresztül Irakon.)
- Record route – A továbbítás útvonalának naplózása.
- Timestamp – A késleltetési idők naplózása.

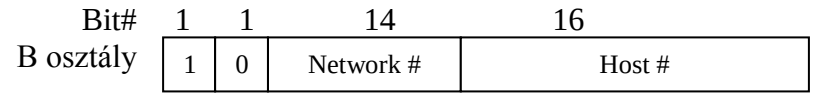
IP címek (csomópont azonosítók)

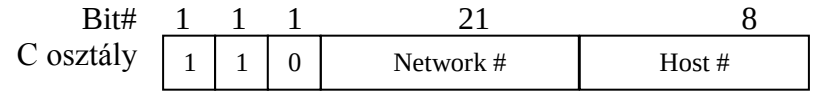
- A csomópont hálózatbeli azonosítója.
- Pontozott decimális megjelenítés. Pl.157.45.190.57
- Az azonosítók kezelése – InterNIC. (NIC – Network International Center.) Ez a szervezet specifikálja a hálózat címtartományának elejét (a hálózatazonosítót), a többit (a hálózaton belülit) az intézmény.

IP címosztályok

A osztály – nagy intézmények számára.







Első bájt szabály

Kezdőbit(ek)	1. Bájt értéke	Osztály
0	0 - 127	A
10	128 - 191	B
110	192 - 223	C

Speciális IP címek

00000000.00000000.00000000.00000000		Az aktuális gép.
000000….00000	Host	Az aktuális hálózat megadott gépe.
11111111.11111111.11111111.11111111		Broadcast az aktuális hálózaton.
Network	00000000….00000000	A megadott hálózat azonosítója.
Network	11111111….11111111	Broadcast a megadott hálózaton.
01111111	Bármi	Loopback.

IP alhálózatok

Miért van szükség alhálózatok létrehozására?
- Az intézmény logikai működése, felépítése, térbeli elhelyezkedése indokolja.
- Egy IP hálózaton több üzenetszórási (broadcast) tartományt kell létrehozni.

Hogyan hozunk létre alhálózatokat?
- Az IP cím host részének legmagasabb helyiértékű bitjeiből néhányat az alhálózat (subnet) azonosítására használunk.
- Az új hálózat-csomópont határt egy ún. hálózati maszk (netmask) értékkel jelöljük.

Hálózati maszk

Egy olyan 32 bites maszk, mely 1-es bit értékeket tartalmaz a hálózat és alhálózat azonosításában résztvevő bithelyeken és 0-s bit értékeket tartalmaz a csomópont azonosítására szolgáló bithelyeken.
A hálózati maszk segítségével az eredetileg az osztályba sorolás által (statikusan) meghatározott hálózat-gép határ dinamikusan módosítható.

Hálózati maszk – példa:

- Hálózat IP címe: 197.45.112.0

C osztályú intézmény azonosítója

- Alapértelmezett hálózati maszk: 255.255.255.0
- Használjunk 3 bitet alhálózat azonosítására. 001|00000 - 32

alhálózat azonosító gép azonosító

- Hálózati maszk: 255.255.255.224
- Összesen 8 alhálózat elkülönítésére van lehetőség.
- Általában a csupa 0 és a csupa 1 bit értékekből felépülő alhálózat azonosítókat nem használják (6 alhálózat építhető).

Az alhálózatok címei:

Sorszám	Alhálózat címe	Alhálózati gépcímek
1.	197.45.112.32	197.45.112.33-62
2.	197.45.112.64	197.45.112.65-94
3.	197.45.112.96	197.45.112.97-126
4.	197.45.112.128	197.45.112.129-158
5.	197.45.112.160	197.45.112.161-190
6.	197.45.112.192	197.45.112.193-222

Hálózati maszk szerepe

Továbbra is .35 marad, de a router tudja, hogy a .32 felé kell irányítani.

(+ 19. oldal)

A kettős címrendszer problémái

- Az adatkapcsolati réteg enkapszulációjához meg kell határozni a hálózati címhez tartozó fizikai címet.
- Bizonyos helyzetekben (pl. Hálózati boot esetén) szükség lehet arra, hogy a fizikai címhez meghatározzák a hálózati címet.

Hálózati cím → fizikai cím (ARP)

ARP (Address Resolution Protocol) RFC 826
- Minden node egy táblázatban (ARP táblázat) tartja nyilván a hálózati címekhez tartozó fizikai címeket.
- Hogyan kerül be egy új adat (címpár) a táblázatba?
  1. ARP kérdés: Ki tudja az X hálózati cím fizikai címét?

2. A kérdés keretét üzenetszórásos küldéssel az alhálózat valamennyi csomópontja megkapja és feldolgozza.
3. Ha valamely csomópont „magára ismer” az X hálózati címben, akkor a saját fizikai címével megválaszolja az ARP kérdést.

Fizikai cím → Hálózati cím (RARP)

RARP (Reverse Address Resolution Protocol) RFC 903
- Csak speciális esetekben szükséges (pl. hálózati boot).
- Egy (vagy több) RARP szerver táblázatban (RARP táblázat) tartja nyilván a fizikai címekhez tartozó hálózati címeket.
- A táblázatot a rendszeradminisztrátor tartja karban.
- A fizikai cím - hálózati cím összerendelés statikus.
- Több RARP szerver esetén egy fizikai címhez minden RARP szerveren ugyanazt a hálózati címet kell rendelni (nem függhet a szervertől az összerendelés).

Működési vázlata:
1. RARP kérdés: Ki tudja az X fizikai cím hálózati címét?
2. A kérdés keretét üzenetszórásos küldéssel az alhálózat valamennyi csomópontja megkapja.
3. A RARP szerverek feldolgozzák a kérdést: Ha megtalálják a táblázatukban az X fizikai címet, akkor a táblázatban található hálózati címmel megválaszolják a RARP kérdést.

Fizikai cím → Hálózati cím (DHCP)

DHCP (Dynamic Host Configuration Protocol)
Működési vázlata:
1. DHCP kérdés: Ki tud adni egy IP címet?
2. A kérdés keretét üzenetszórásos küldéssel az alhálózat valamennyi csomópontja megkapja.
3. A DHCP szerverek feldolgozzák a kérdést: Ha a kezelt címtartományukban van még szabad IP cím, akkor azzal megválaszolják a DHCP kérdést.
4. A kliens a hozzá érkező DHCP válaszokból választ egyet, s visszajelzi a választását a megfelelő DHCP szervernek.
5. A DHCP szerver „könyveli” a címválasztást (foglalt lett a cím), s a könyvelésről megerősítést küld a kliensnek. Ez a kiosztás általában időszeletekre szól.

Az Internet növekedése

90 Január	927
90 Április	1525
90 Július	1727
90 Október	2063
91 Január	2338
91 Április	2622
91 Július	3086
91 Október	3556
92 Január	4526

IP címosztályok problémái

Az IP címosztályok statikus hálózat-gép határának problémái:
- A kb. ~5000 csomóponttal rendelkező intézmények számára a „B” osztály túl nagy, a „C” osztály túl kicsi.
- Szükség van egy dinamikus határ meghatározására (változó hosszúságú hálózati maszk).
- A 90’-es évek elején az időegység alatt kiosztott új hálózatcímek száma exponenciális növekedést mutatott. (A „C” osztályú címek száma 2^{21}!)
- A router-táblázatok mérete a hálózatok számával arányos.
- Meg kell akadályozni a router-táblák robbanásszerű növekedését.

IP címosztály problémák - megoldás

A megolcás:CIDR (Classless Inter-Domain Routing) RFC 1519.
- Folytonos „C” osztályú címek kiosztása („B” helyett).
- A hálózat-gép határ változó hosszúságú hálózati maszk segítségével tetszőleges bitszámmal balra (supernetting) illetve jobbra (subnetting) tolható.
- Területi elrendeződés szerinti címtartomány –zónák kialakítása.
- Összevont forgalomirányítási információk a hálózati maszkok segítségével.
- A hálózati címek reprezentációja: <Hálózat IP szám, Hálózati maszk>

Kontinensek IP címtartományai

A „C” osztályú IP címtartományokat kontinentális alapon osztják ki (router táblák mérete jelentősen csökkenthető) RFC 1366, 1466:

Kontinens	Címtartomány
Európa	194.0.0.0 – 195.255.255.255
Észak-Amerika	198.0.0.0 – 199. 255.255.255
Közép- Dél-Amerika	200.0.0.0 – 201. 255.255.255
Ázsia, Ausztrália	202.0.0.0 – 203. 255.255.255

CIDR példa

Egy Internet-szolgáltató 2048 db „C” osztályú IP cím kiosztásáról rendelkezik:
104.24.0.0 – 194.31.255.255
A szolgáltatót (kívülről) specifikáló információ: <194.24.0.0, 255.248.0.0>
A szolgáltatóhoz 3 intézménytől érkezik Internet csatlakozási igény:
AI 2000 csomópont
BI 4000 csomópont
CI 1000 csomópont
Az intézményeknek kiosztott címek:
AI 194.24.0.0 – 194.24.7.255; <194.24.0.0, 255.255.248.0> (2048 cím)
BI 194.24.16.0 – 194.24.31.255;<194.24.16.0, 255.255.240.0> (4096 cím)
CI 194.24.8.0 – 194.24.11.255; <194.24.8.0, 255.255.252.0> (1024 cím)

A példa működtetéséhez szükséges forgalomirányítási információk:
- Az európai (aggregált) forgalomirányításhoz:
  <194.24.0.0, 255.248.0.0>
  Egy bejegyzéssel 2048 db. „C” osztályú cím kezelhető.

- Az Internet-szolgáltató belső forgalomirányításához:
  <194.24.0.0, 255.255.248.0>
  <194.24.16.0, 255.255.240.0>
  <194.24.8.0, 255.255.252.0>
  Három bejegyzéssel 28db. „C” osztályú cím kezkelhető.

{
Nem egyedi IP cím, hanem tartományról van szó.
Amiatt, hogy a BI-nek 1 címtartománya legyen, nem a .8.0-val folytatják, hanem a .16.0-tól.
Ügyelni kell, hogy mindig az első szabad tartományból kerüljön a cím kiosztásra.
}

CIDR példa - routing

(25. oldaltól) Pl. érkezik egy 194.24.9.35-ös csomag. … ÉS művelettel meghatározza a célhálózat címet. Ezt minden interfészre el kell végezni.

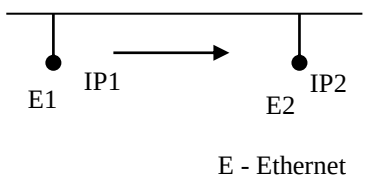
Forgalomirányítási alapfogalmak

- IP forgalomirányítás (routing):
  Csomagok (IP datagramok) továbbítási irányának meghatározásával kapcsolatos döntések meghozatala.
- Forgalomirányítási táblázat (routing table):
  A forgalomirányításhoz szükséges információkat tartalmazó táblázat.
- Forgalomirányítási protokoll (routing protocol):
  A forgalomirányítási táblázat(ok) felépítéséhez, karbantartásához szükséges információk továbbítását (routerek közötti cseréjét) leíró protokoll (pl. RIP, OSPF, BGP).
- Forgalomirányított protokoll (routed protocol):
  Olyan, hálózati réteghez kötődő (hálózati adattovábbítási) protokoll, melyet a forgalomirányító (router) irányítani képes (pl. IP, IPX).
- Autonom rendszer (AS – autonom system):
  Hálózatok forgalomirányítási adminisztrációs egysége, amelyben egy közös forgalomirányítási stratégia (routing protocol) érvényesül (azonos forgalomirányítási alapelvek).
- Metrika:
  Egy adott forgalomirányítás eredményeként előálló útvonal hosszának (költségének, jóságának) mérési módja.

Forgalomirányítási konfigurációk osztályozása

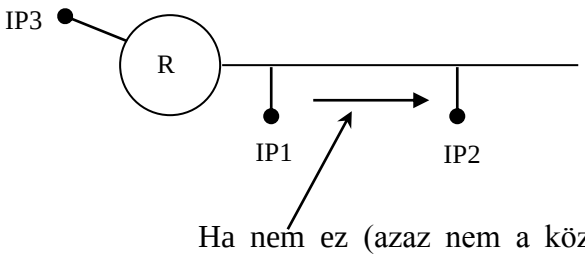
- Minimális routing:
  Teljesen izolált (router nélküli) hálózati konfiguráció.

  A feladó az IP számból és a netmaszkból meghatározza, hogy a cél vele vele egy hálózatban van-e.
  (Az ÉS ugyanazt a címet adja a sajátja és a cél esetén.)



E - Ethernet

- Statikus routing:
  A forgalomirányítási táblázatot a rendszeradminisztrátor tartja karban. Amit egyszer beírunk, az marad a cím. A világ felé egy, vagy csak néhány kapcsolódási pont van.



Ha nem ez (azaz nem a közvetlen elérésű hálózatban van), akkor a statikusan beírt címre küldi a csomagot. A bitenkénti ÉS-sel megállapítja, hogy nem a közvetlen hálóban van. Előny: nincs hálózati kommunikáció, a rendszeradminisztrátor tartja karban a táblát.

- Dinamikus routing:
  Ha több kapcsolódási pont van. Hogyan tartják karban a táblát?
  A forgalomirányítási táblázat(ok) valamilyen routing protocol segítségével kerülnek karbantartásra.
  – Belső forgalomirányítási protokollok (IGP – belső forgalomirányítási osztály, pl. RIP, OSPF).
  » Legfőbb alapelv a „legjobb útvonal” meghatározása.
  – Külső forgalomirányítási protokollok (EGP – Extension Gateway Protocol – külső forgalomirányítási osztály, pl. EGP – ugyanolyan nevű protokoll, BGP).
  » Nem feltétlenül a legjobb útvonal meghatározása a cél (politikai alapú forgalomirányítás – BGP).

Távolságvektor alapú forgalomirányítás

Működési alapelv:
- A routerek minden elérhető célra (gép vagy hálózat) nyilvántartják, hogy a legjobb úton milyen irányban milyen távolsággal érhető el az adott cél (távolságvektor).
- A forgalomirányítók ezen információkat meghatározott időközönként kicserélik egymással.
- Az új információk birtokában a routerek ellenőrzik, hogy szükséges-e változás valamelyik eddig ismert legjobb úttal kapcsolatban

Routing tábla felépítés (Bellman-Ford)

Kiindulási helyzet (most kapcsoljuk be a routert):
- Legyen: $D(i,j)=\begin{cases}0, \text{ ha } i=j,\\ \infty, \text{ egyébként.}\end{cases}$
- Minden *i* entitás ismeri a *d(i,k)* távolságot minden *k* szomszédjára vonatkozóan.

Működési algoritmus (tetszőleges $i \rightarrow j$ útra vonatkoztatva):
1. Minden *i* entitás minden *k* szomszédjától megkapja a *D(k,j)* értéket.
2. Az *i* entitás minden *k* szomszédjára vonatkoztatva kiszámítja az (1) formulában szereplő minimum értékrt az 1. pontban kapott információ segítségével.
   Ha az új minimum érték kisebb, mint az eddigi *D(i,j)*, akkor a *j* entitás *i*-ből aktuálisan az

új minimumot szolgáltató *k* entitás felé érhető el a számított minimumértéket használva *D(i,j)*-ként.

3. Folytassuk az 1. pontnál.

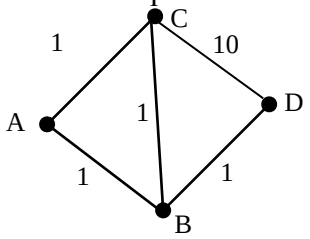
Az eljárás véges sok lépés után az optimális utat szolgáltatja.

Távolságvektor – routing tábla problémák

Túl kicsi kezdőérték probléma:
- Ha az optimális út „megsérül”, nagyobb költségű (hosszabb) út nem léphet helyébe.
- Megoldás: Az optimális út irányából érkező nagyobb költség felülírja a (kisebb) költséget.

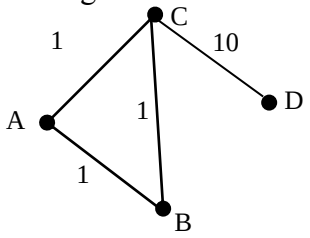
Végtelenig számlálás (Count to infinity) pobléma:
- Az eljárás bizonyos esetekben igen lassan reagál a topológia változására.

Végtelenig számlálás – példa:



Tekintsük a „D”-be irányuló forgalomirányítást.
Kiinduló forgalomirányítási bejegyzések (optimális irányok D-be):
- A: B,2
- B: D,1
- C: B,2

Tekintsük a routing táblák alakulását a B-D link sérülése esetén:



A	B,2	C,3	C,4	C,5	…	C,10	C,11	C,11
B	---	C,3	C,4	C,5	…	C,10	C,11	C,11
C	B,2	A,3	A,4	A,5	…	A,10	D,10	D,10

(Lassú a konvergencia, amíg megtalálja a jó utat. Ez leterheli a hálózatot.)

Routing Information Protocol – RFC 1058

A Routing Information Protocol (RIP) jellemzői:
- Távolságvektor alapú IGP protokoll.
- Régi, de folyamatosan fejlesztik, javítják.
- Metrika: Hop-ok száma (16=végtelen távolság).
- Max. 15 router hosszúságú optimális útvonalak esetén használható.

- - 30 másodpercenkénti routing információ küldés.
- „Triggerelt update” a végtelenig számlálás idejének csökkentésére.

(A két routeres ciklust kiküszöbölhetjük, ha a legjobb utat nem küldjük vissza)

RIP Forgalomirányítási Táblázat

A RIP routing táblázatának legfontosabb elemei:
- A cél (gép vagy hálózat) IP száma.
- A célhoz vezető optimális út hossza.
- Az optimális út szerint következő router IP száma.

} Ezek kötelezők (a működéshez elengedhetetlen).

- A következő routerhez vezető interfész azonosítója. (Hogy ne kelljen bitenkénti ÉS-t végezni.)
- Időzítéssel kapcsolatos információk.
- Különböző jelzőbeállítások (Flag-ek).

OSPF – a forgalomirányítók a teljes routertopológiát nyilvántartják. – Egy optimalizálási eljárás fog lefutni.

Link állapot alapú forgalomirányítás

Link State Routing működési vázlat:
1. Szomszédok felfedezése.
2. A szomszédok felé vezető út költségének (hosszának) mérése.
3. Csomag készítés a mérési eredményekről.
4. A készített csomag küldése a hálózati egység összes forgalomirányítójának.
5. Minden router ismeri a hálózat topológiáját, s ki tudja számítani (pl. Dijkstra algoritmussal) a többi routerhez vezető optimális utat (feszítőfa, spanning tree).

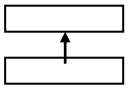
(Inkonzisztens térkép alakulhat ki a routerekben, ha a hiba és annak javulása gyorsan követi egymást.)

Transzport (szállítási) réteg

Feladata: bármely két csomópont között egy pont-pont jellegű megbízható adatátvitelt nyújtson.
Két protokoll terjedt el: TCP, UDP. A TCP a felsőbb réteg számára (applikációs réteg számára) megbízható összeköttetést szolgáltat, az UDP nem.
A legelterjedtebb hálózati szolgáltatások a TCP-re támaszkodnak. Kicsi szolgáltatásnál az UDP hatékonyabb lehet (pl. DNS megvasósítása).

A protokoll fejlécek felépítése:

Applikációs r.
Transzport r.
Hálózati r.



Szükség van egy azonosítóra, hogy tovább tudja adni az információt (a Tr. – Appl. r. átmenetnél). Ezeket portszámoknak nevezzük, a TR. réteg fejrészében helyezkedik el, innen tudja, hogy milyen szoftvernek kell továbbítani az információt.

Szolgáltatást igénylő szoftver — Szerver szoftver: valamilyen hálózati szolgáltatást nyújt

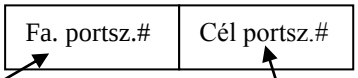
A kommunikáció úgy indul, hogy a kliens a szerverhez fordul szolgáltatásigénnyel. A kliensnek ismernie kell a szerver portszámot, hogy meg tudja szólítani. Ezért a szerver szolgáltatások applikációs rétegbeli szoftverek azonosítóját rögzítik. A portszámokat méretük alapján 3 osztályba rögzítik.

- 1000 alatt rögzített portszámok: a szerver által nyújtott hálózati szolgáltatások szoftverének portszáma.
- 1000-2000: átmeneti.
- 2000 fölött: szabadon felhasználható

Pl. 80 – www azonosítása
25 – levelező szerver azonosítási portszáma
7 – echo

Némely azonosítót csak a TCP-re adják meg, de van amit TCP-re és UDP-re is.

UDP csomag szerkezete: A feladó és a cél portszáma szerepel benne. (+ ellenőrző összeg).
(Nem összeköttetés alapú, megbízhatatlan. Itt az applikációs rétegre marad a megbízhatóság lekezelése.)

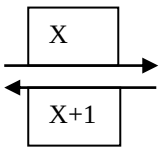


Ez rögzített

Ennek a specifikációja: nem rögzített, az operációs rendszer által a kliens entitáshoz rendelt szám. Elképzelhető, hogy ugyanazon hardverhez többen fordulnak egyszerre valahová, ezt el kell tudni különíteni. Ez csak úgy történhet, ha külön számot kapunk.

TCP: összeköttetés alapú, megbízható.

- A csomagok sorszámokat kapnak. Ennek segítségével összerakhatók. A hiba felderítésének lehetőségét adja.
- Nyugtázás – érkezési visszajelzés, hogy mely sorszámú csomagok érkeztek meg, illetve hogy milyet vár következőként. (Ez megállapodás kérdése.) Előre mutató nyugtázás: a soron következő sorszámot küldik vissza.



- Ablakozás: egy csomag sorozat (pl. 500 db.) kerül továbbításra, s ezután csak egy nyugtát vár vissza, ez hatékonyabb.

Emellett a TCP biztosít egy „túl gyors adó – lassú vevő” beállítási, átviteli szabályozást is. (Ezt nem részletezzük.)

A TCP fejléc felépítése:

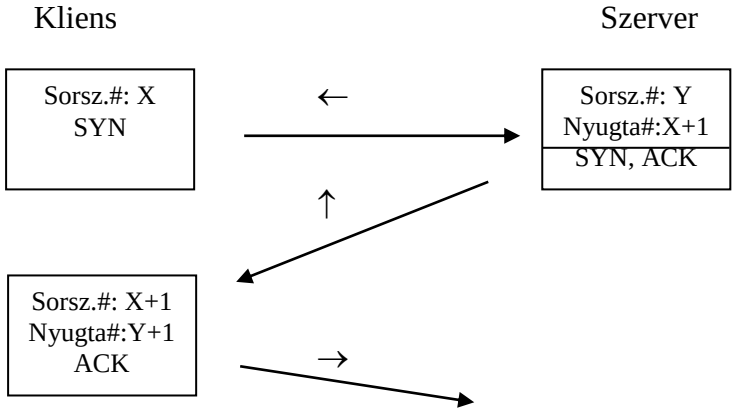
A sorszámozás – nyugtázás felépítése:
- Portszámok
- Sorszámok
- Nyugtázás

- Ablakméret
- Jelzőbitek
    SYN – a kommunikáció indulásával és lezárásával kapcsolatos (szinkron).
    ACK – a nyugtasorszám érvényességét jelzi (Jelzi, hogy valós nyugtasorszám, vagy nem. Pl. az elsőnél még nincs mit nyugtázni.)
    FIN – a kommunikáció végét jelzi (finish)

Kapcsolat kiépítése → kommunikáció → kapcsolat bontás.

Hogyan történik a kapcsolat kiépítése? Módszer: háromutas kézfogás.
1. Sorszámok egyeztetése, ablakméret beállítás.
- A szerver várja a kliens megszólítását.
- A kliens összeállítja a csomagot, s ennek lesz egy sorszáma (X#).
- Beállítja a szinkronizációs jelzőbitet.
- Ezeket elküldi a szervernek.
- A szerver máshol kezdi a sorszámozást, ez az ő magánügye. (Y#) Beállítja a SYN-t, az ACK-t is, ezeket visszaküldi a kliensnek, és az X+1-et is.

Ezt három út után vágják el:



Ezután már mehet a tényleges kommunikáció.
Menet közben kezdeti ablakméreteket felhasználnak egymásnak (de ez mellékes dolog).

A leállításnál hasonló probléma van. A leállítás jelzése – nyugtázás: hasonló, mint fent, de a szinkronizáció helyett a FIN-nel. A szerver is , a kliens is kezdeményezheti a befejezést.)

Applikációs réteg

DNS – Tartománynév kezelő rendszer

Nevek használata – kezdeti megoldások

Természetes emberi igény IP számok helyett nevek használata.
- Kezdeti megoldás: hosts.txt állomány letölthető a NIC-től.
- Néhány 100 csomópont esetén működtethető.
- Internet növekedése (80'-as évek) – új megoldás szükséges.

DNS – Domain Name System RFC 1034, 1035
- Hierarchikus tartományalapú névkiosztási séma.
- Osztott adatbázisban történő implementáció.

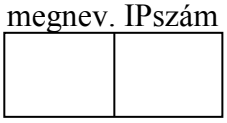
DNS - Tervezési szempontok

Alapvető cél: nevekhez erőforrások rendelése.
Nagyméretű adatbázis elosztott kezelése.
- Átmeneti tárolás (cache) lehetőség biztosítása.

Általános célú megoldásnak kell lennie.
- név → hálózati cím,
- név → postafiók információ,
- Egyéb (előre nem ismert) applikációk támogatási lehetősége.

Tagolás: osztály és típus szerint.
A lekérdezési tranzakció független a kommunikációs eszköztől.
Platformfüggetlen megvalósíthatóság.

1. Helyi host tábla: A rendszer automatikusan kikeresi.
   Jelentős adminisztrációt igényel.
   Lehet N:N kapcsolat is.



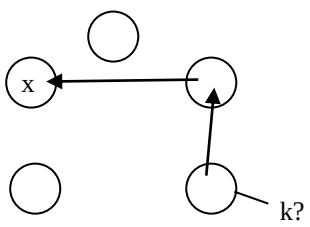
2. Helyi hálózati host tábla: kijelölünk egy csomópontot a hálózatban, ahol ugyanúgy megcsináljuk a táblát, s ezt valamennyi szervernek hozzáférhetővé tesszük. UNIX környezetben ez a NIS (Network Information System) és NIS+ rendszerek. Világméretekben ez nem működőképes, csak helyi hálózatokban.
3. Osztott adatbázis kiépítése (a megoldás) DNS.
   Az adatbázis egyes részét a világ más-más részén tároljuk. Fontosabbnak tekinthető az információ rendelkezésre állása, mint aktuális volta.

DNS – Alkalmazási feltételezések

Adatok többségének lassú változása.
Adminisztratív határok (zónák) kialakítása.
- Általában a zónák intézményeket reprezentálnak.
- Névszerver(eke)t üzemeltetnek.
- Felelősek a tartománynevek egy halmazáért.

Biztosítani kell a kliensek névszerverhez kapcsolódási lehetőségét.
Adathozzáférés kiemelt prioritása (konzisztenciával, naprakészséggel szemben).
Más névszerveren tárolt adatra vonatkozó kérdés megválaszolása:
- Iteratív módszer (kötelező).
- Rekurzív módszer (opcionális).

Osztott adatbázis lekérdezés:
- iteratív
- rekurzív.



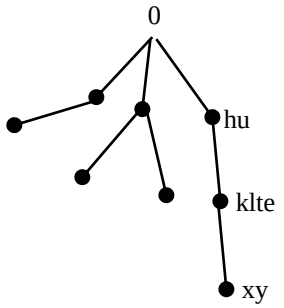
Kérdés merül fel, ami máshová tartozik. Hogyan történik a megválaszolás?

1. Iteratív: megkapja a célszerű keresési helyeket, a kérdésből kapott választ mindaddig csinálja, amíg vagy talál, vagy nincs ilyen adat. Ezt kötelezően implementálják a DNS-ben. (Minden szerver ismeri.)
Az iteratív módszer a szerver oldalon egyszerűbb megvalósítású. A kérdezőnek kell intelligensnek lenni, mert ki kell értékelni a választ.
2. Rekurzív: nem adja vissza a választ, hanem tovább küldi, ahová szerinte tartozik. Stb.
Rekurzívnál kliens oldalon sokkal egyszerűbb a feladat, szerver oldalon bonyolultabb.

DNS - Komponensek

A tartománynevek rendszerének három fő komponense:
- Tartománynevek (körzetnevek)tere és erőforrás rekordok.
- Névszerverek.
- Címfeloldó (resolver) programok.

Tartománynevek tere



Fa típusú (összefüggő, körívmentes) gráf, melyben minden csúcs egy erőforráshalmazt reprezentál.
A csúcsokhoz egy (max. 63 bájt hosszúságú) címkét rendelünk.
- Két testvér csúcs címkéje nem lehet azonos.
- A zéró hosszúságú címke („null címke") a gyökér számára kizárólagosan foglalt.
- Címke belső reprezentációja:
  - A címke hossza egy bájton
  - A megfelelő karaktersorozat (bájt-string).
- A kis- és nagybetűk között nem teszünk különbséget, de célszerű megtartani a forrás írásmódját.

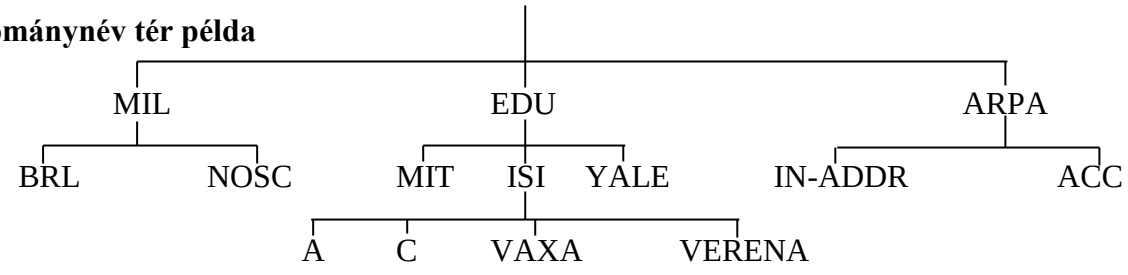
Abszolút tartománynevek: gyökértől a csúcsig a címkesorozat.

Gráfelméleti alapok DNS alkalmazása:
- A tartománynevek terében bármely csúcs egyértelműen reprezentálható a csúcstól a gyökérig vezető utat leíró címkesorozattal (abszolút tartománynév).

Abszolút tartománynév belső reprezentációja:
- Maximum 255 bájt hosszúságú.
- A címkék sorrendhelyesen konkatenáljuk.
- Szükségképpen NULL karakterrel (0 bájttal) végződik.

Tartománynevek reprezentációja felhasználói interfészeknél:
- Címke-sztring sorozat, elválasztó karakter a pont (.).
- Lehet abszolút és relatív.

Tartománynév tér példa



Abszolút név felhasználói specifikációja pl.:
- vaxa.isi.edu.
Relatív név felhasználói specifikációja pl.:
- vaxa (relatív az isi.edu.-hoz képest)
- vaxa.isi (relatív az edu.-hoz képest).

vaxa.isi.edu. belső reprezentációja (hexadecimális forma):

04	76	61	78	61	03	69	73	69	03	65	64	75	00

Erőforrás rekordok (információs erőforrások)

A tartománynevek egy csomópontot specifikálnak.
A csomópontokhoz egy erőforrás-halmaz társítható.
Az információs erőforrások ún. erőforrás rekordokban (Resource Record, RR) tárolódnak.
Az erőforrás rekordok sorrendje lényegtelen.
Az erőforrás rekordok mezői:
- tulajdonos: melyik csúcspontnak a tulajdonában lévő erőforrásról van szó.
- osztály: vagy protokoll architektura pl.: Internet.
- típus: milyen jellegű információt tárol le a rekord.
- élettartam: az adott információ milyen hosszú ideig tárolható le az átmeneti tárolóban.
- adat.

Erőforrás rekordok szerkezete

Tulajdonos:
- Az a tartománynév, amelyhez a RR tartozik.

Osztály:
- 16 bites érték, mely egy protokollcsaládot, vagy egy protokollt azonosít.
- IN – az Internet protokollcsalád.
- CH – A Chaos protokollcsalád.

Élettartam (TTL):
- 32 bites érték: A RR max. felhasználhatósági ideje (sec).

Érték (RDATA):
- A típustól függően értelmezendő bitsorozat (adat):

Típus	**Adat**
A	32 bites IP cím (IN osztály esetén).
CNAME	Tartománynév.
HINFO	Tetszőleges sztring.
MX	16 bites prioritás érték és egy tartománynév.
NS	Egy host tartományneve.
PTR	Egy tartománynév.
SOA	Több mezőből álló rekord.

A – address – a tulajdonos hálózati címét specifikálja (32 bites IP szám – Internet esetén)
CNAME – egy IP-számhoz több nevet akarunk társítani. Ez egy másodlagos név lesz.
MX – postafiók információk.
- prioritás jelzőszám
- tényleges érték v. tartománynév, ide kerül továbbításra.

PTR – pointer – egy mutatót vezetünk be a tartomány egy másik elemére. Fordított IP szám – Név használatra. INADDR – ARPA – címek alapján történő keresést tudjuk elvégezni.

xy.klte.hu

IP szám: hálózat tartománynév, host

ezért ezt fordítva írják fel

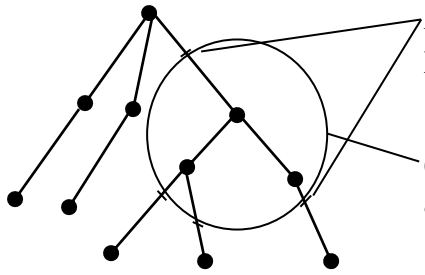
pl.: 193.6.135.55 ⇒ 55.135.6.193. in-addr.arpa.

NS – névszerver információ – az elérés adatait tárolják le.
HINFO – Hardver információ – a rendszer működésében megjegyzésként implementálódik.
SOA – biztonsági információ – hitelességi adatok (source of authoring ?)

A tartománynév tér partícionálása

A tartománynevek tere két (természetes) módon darabolható:
- Az osztály tagozódása alapján.
  - A különböző osztályok paralell név tér faként foghatók fel.
- A tartománynév tér (fa) éleinek átvágásával.
  - Ha a tartománynevek terében bizonyos éleket „átvágunk”, akkor a maximálisan összefüggő részgráfok szintén fa struktúrájúak.
  - Egy ilyen maximálisan összefüggő részgráfot zónának nevezünk.
  - Egy zóna reprezentálható a gyökérhez legközelebbi csúcsának tartománynevével.
  - A zónák közötti „átvágásokat” nyilván kell tartanunk.

Partíciók létrehozása:



Átvágások (Ezt nyilván kell tartani a szomszédos pontokon).

Összefüggő részek, zónák: a DNS szempontjából adminisztratív egységet képez.

Névszerverek

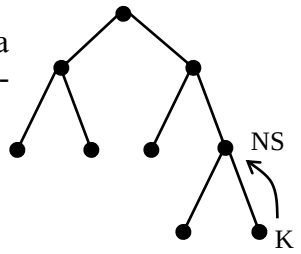
A névszerverek olyan szerver-programok, melyek:
- Információt tárolnak a tartománynevek gráfjáról.
- Tartománynevekhez tartozó erőforrás rekordokat tárolnak.
  - Egy (vagy több) zónához tartozó valamennyi csomópont hiteles (authoritative) erőforrás rekordját.
    » A zóna gyökérhez legközelebbi csúcsát leíró adatokat.
  - Szomszéd (gyermek) zónákhoz (és ezek névszervereihez) vezető információkat.
  - Időlegesen más zónákhoz tartozó RR-t (cache).
- Kérdéseket (lekérdezéseket) válaszolnak meg.
  - Rekurzív módon.
  – Nem rekurzív (iteratív) módon.

{
Elsődleges NS – a rendszeradminisztrátor kézzel beírt adatait tartja.
Másodlagos NS – másolatokat tartalmaz az elsőről.
Hogy ne kelljen minden rekordról kiírni az élettartamot.

SOA-ban van egy sorozatszám, ha az azonos, nincs szükség másolatot csinálni.
Időzítéses információ:
- mennyi időnként kell másolatot csinálni?
- mennyi idő után mondja, hogy nem sikerült másolatot csinálni?

Cache – átmeneti tár.

Lekérdezés:
A kliens jelzi, hogy rekordinformációt kér. A NS vagy meg tudja válaszolni (mert az általa nyilvántartott zónában van, vagy a cache-ben). Ha nem tudja megválaszolni, 2 eset lehet:



1. Itt kellene lennie az adatnak, de nincs → sikertelen, nemleges.
2. Ha más zónára vonatkozik a kérdés, egy másik NS-hez fordul.

Eddig a rekurzív és az iteratív ugyanaz. Innentől:

Iteratív: visszaad egy adatot, hogy hova vonatkozik a kérdés.
1. adott zóna alá → az ott lévő NS-hez fordul a kliens a kérdéssel.
2. a zóna fölé → akkor gyökér (amit minden NS-ben letárolnak). Ez a DNS-ben kötelezően implementált.

A kliens végzi a lekérdezést.

Rekurzív: az NS végzi a lekérdezést. (Ezt vagy implementálták már, vagy nem. Ajánlhatja a kliens felé a rekurzív szolgáltatást.). Előny: a NS-ben tárolt Cache több NS számára elérhető lesz, illetve a zóna számára. (Iteratívnál csak a kliens gép rendelkezik az információval.)
}

DNS kérdések (Applikációs szintű.)

A lekérdezések és válaszok egy standard formátumot követnek:
- Fejrész
  - Egy bitkombináció a különböző kérdések (pl. standard query, status query stb.) elkülönítésére.
- Kérdés
  - A kérdéses név, és a kérdés egyéb paraméterei.
- Válasz
  - A kérdéshez tartozó direkt válasz.
- Hitelesség
  - A hiteles szerverek adatait leíró rekordok. (A válasz jöhet a cache-ből, de NS-ből is.)
- További adatok
  - A kérdéshez kapcsolódó egyéb információk (RR).

DNS kérdés -példa

Fejrész	OPCODE=Standard Query
Kérdés	QNAME=ISI.EDU. CLASS=IN TYPE=MX
Válasz	
Hiteles	
További	

DNS válasz -példa

Fejrész	OPCODE=Standard Query, Response, AA
Kérdés	QNAME=ISI.EDU. CLASS=IN TYPE=MX
Válasz	ISI.EDU 86400 IN MX VAXA.ISI.EDU.
Hiteles	
További	VAXA.ISI.EDU IN A 10.2.0.27 A 128.9.0.33

Rekurzív –Nem rekurzív módszer

Nem rekurzív módszer:
- Szerver oldalon a legegyszerűbb megvalósítás.
- Minden névszerverben implementált.
- A kliensnek lehetősége nyílik az információk értékelésére.

Rekurzív módszer:
- Kliens oldalon a legegyszerűbb megvalósítás.
- Szerveren megvalósítható átmeneti tárolás (cache).
- Opcionális, mind a szerveren, mind a kliensen implementáltnak kell lennie.
  - A szerver minden válaszában egy bit (RA) jelzi az implementációt.
  - A kliens a kérdésben egy bittel (RD) jelzi a rekurzív igényt.

Címfeloldó (resolver) programok

A címfeloldó programok a felhasználói programok és a névszerverek közötti interfészek.
A címfeloldás ideje lehet kicsi (millisec.) pl. helyi adatokból felépített válasz esetén, de lehet nagy (több sec.) névszerverek adatait kérdezve.
A címfeloldás kliens oldala általában platformfüggő.
Általános funkciók:
- Gép név → gép cím meghatározás.
- Gép cím → gép név meghatározás.
- Általános lekérdezési funkció.

Címfeloldási eredmények

A címfeloldók az igényelt tevékenység elvégzése után (általában) a következő eredményekkel térhetnek vissza:
- Egy vagy több RR, a választ tartalmazva.
- Név hiba (Name Error, NE).
  - A kérdezett név nem létezik.
- Adat nem található (Data Not Found).
  - A név létezik, de a kérdezett adat (vagy típus) nem.

- Átmeneti hiba.
  - Pl. valamilyen hálózati hiba (vonalhiba) miatt a kérdezett zóna nem elérhető.
  - Gyakran nem implementálják külön válaszként.

Programozás

Hogyan épülnek fel a programok?