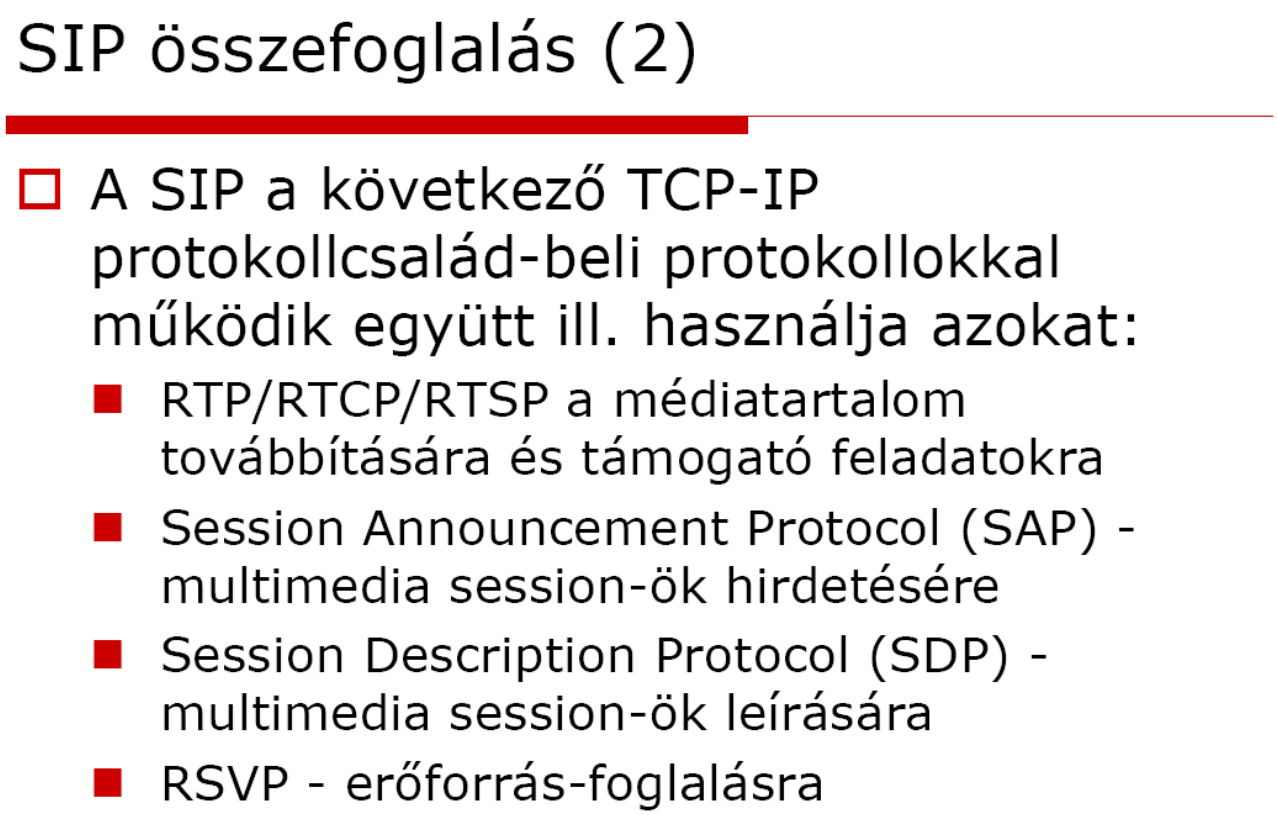
SIP (Session Initiation Protocol):

Alkalmazásrétegbeli protokoll a TCP-IP architektúra szerint, amit azért hoztak létre, hogy szabványosítsák a sessionök létrehozását, módosítását, befejezését egy vagy több partnerrel. Tehát a jelentősége a kapcsolatok felépítsében és menedzselésében van, miután az ezzel kapcsolatos feladatokat elvégezte a program, akkor már egy másik protokoll segítségével folyik a kommunikáció a felépített csatornán.
Session leírókat (descriptors) visz át különböző média-típusokra vonatkozóan, így egyezteti a felekkel, hogy kinek milyen képessége van az átvitelt illetően, és ennek megfelelően szabályoz. Hívásátadást is meg lehet vele valósítani.



Fontos, hogy síp üzeneteket igen könnyű generálni, emiatt könnyen implementálható. Az üzenetek hasonlítanak a HTTP kérésekhez. Mind UDP, mind TCP felett működhet, az e-mail URL-ekhez hasonló URL-eket alkalmaz.
Pl. szabo@hit.bme.hu címre küldhetünk kapcsolódási kérelmet.

A SIP architektúrája: építőelemek

- *User agent - UA*
  - „Request”-eket kezdeményeznek és azok címzettjei
  - IP-telefon, PC, konferencia-szerver
- *Proxy server*
  - A SIP request-eket és response-okat route-olják
  - A kliensek (UA-k) megbízásából tevékenykednek
- *Registrars*
  - Nyilvántartják a felhasználókat egy domain-en belül
  - Név-cím-összerendeléseket kezelnek
- *Redirect servers*
  - Request-re megadják a felhasználó címét
  - De nem kezelnek hívásvezérlést és nem továbbítanak SIP request-eket

UA (User Agent):

Gyk. ezek a végpontok a felhasználók számára. Lehet hardver és szoftver alapú, tehát akár egy IP-telefon, akár egy program. Két összetevőből állnak: a User Agent Client-ből (UAC), ez küldi a SIP-requesteket, és fogadja a SIP-responseokat, illetve a User Agent Server-ből (UAS), ami a responseokat küldi és a requesteket fogadja. Tehát kb. adó-vevőként lehet elképzelni a dolgot, szerintem ez többnyire csak logikai elválasztást jelent.

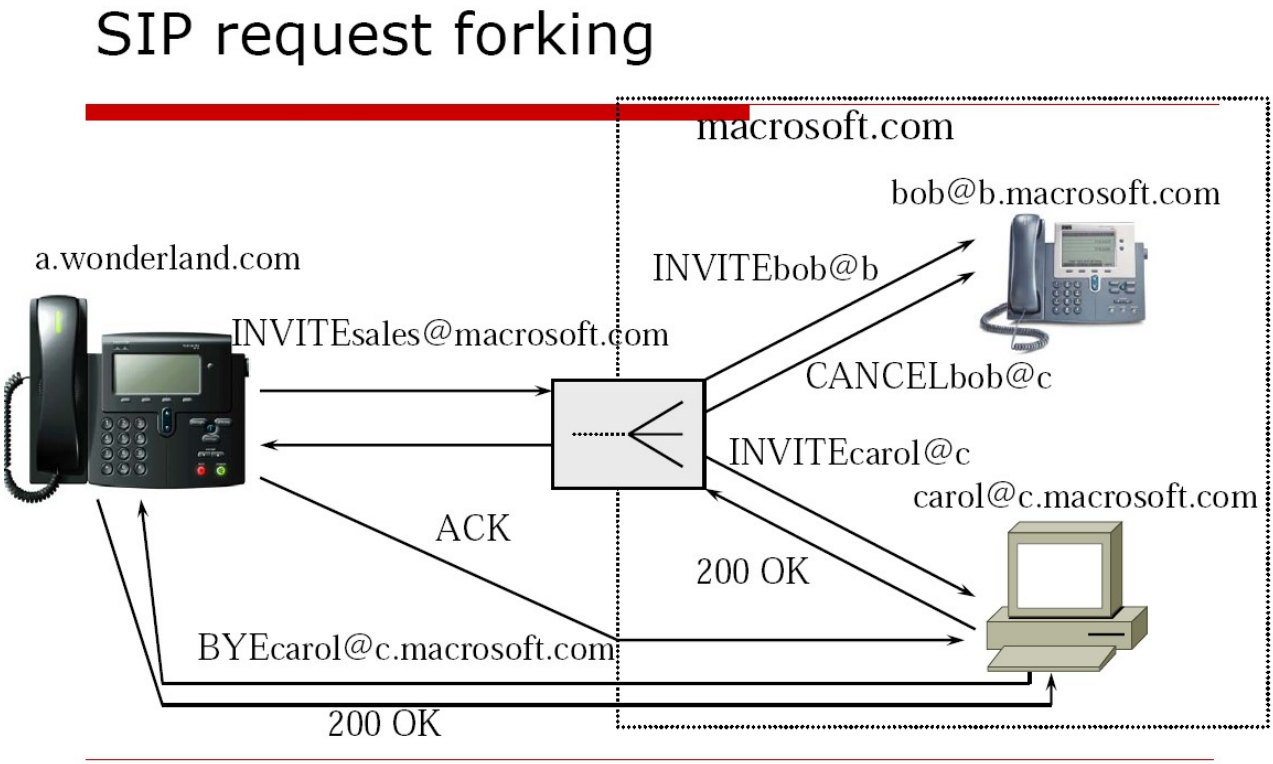
Proxy Server:

A session-invitation-okat (tehát azokat a SIP-üzeneteket, amivel kapcsolat felépítsére szólíthatunk fel más usereket) továbbítja a hívott fél közelébe.
Két típusa létezik:

Állapotmentes:
Ezek egyszerű és gyors működésű rendszerek, a transactionokról nem tudnak, többnyire terhelés-elosztásra használják őket.

Állapot-alapú:
Forking-működés: egy üzenet vételekor 2, vagy több üzenetet küldhet el.

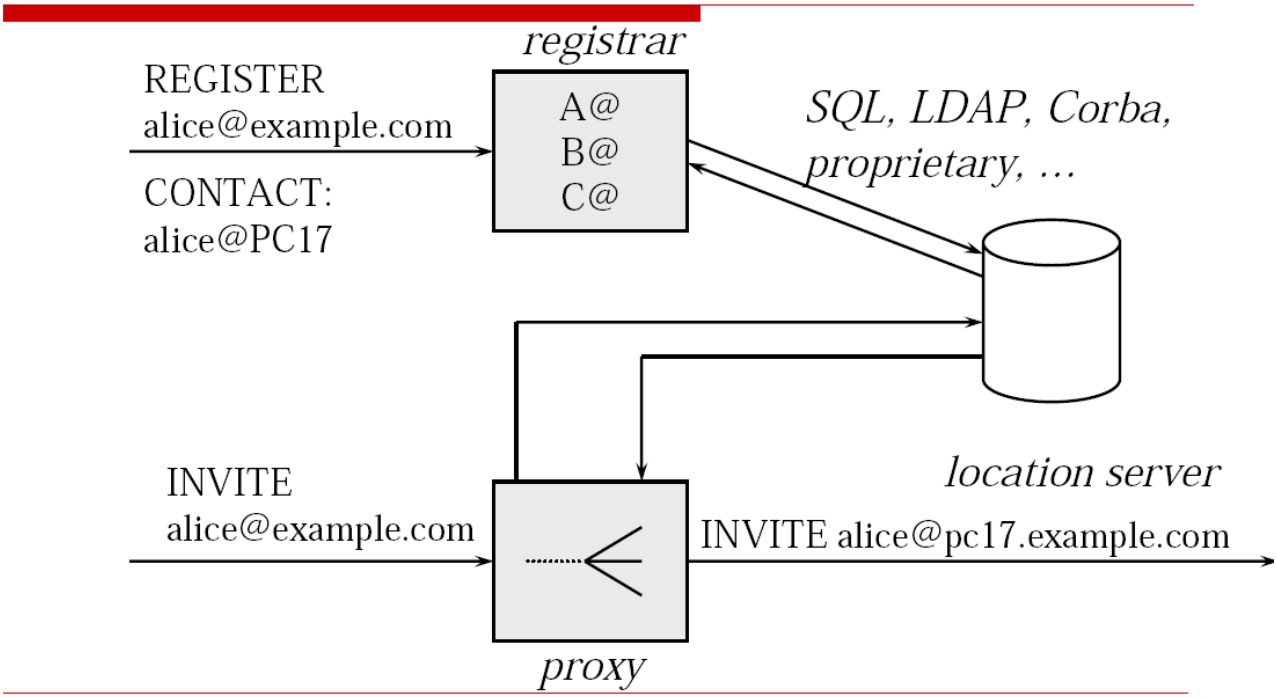
Ekkor tehát 2 vagy több requestet küld a server egy beérkező kérés hatására, amit megtehet egyidejűleg ill. akár egymás után is. Ezzel meg lehet valósítani pl. a hívástovábbítást egy hangpostára, vagy automatikus hívás-szétosztást.



Registrar és Location server:

A lényege, hogy ha egy user éppen nem a megszokott helyén tartózkodik, akkor ezt a tényt registrálhatja a Registrar serveren, így amikor a megszokott helyére hívást kap, akkor a Location server kibányássza onnan, hogy éppen most hol van beregisztrálva, és a megfelelő helyre küldheti a hívást. A registrar azonban mindezt csak egy domainen belül teszi lehetővé.

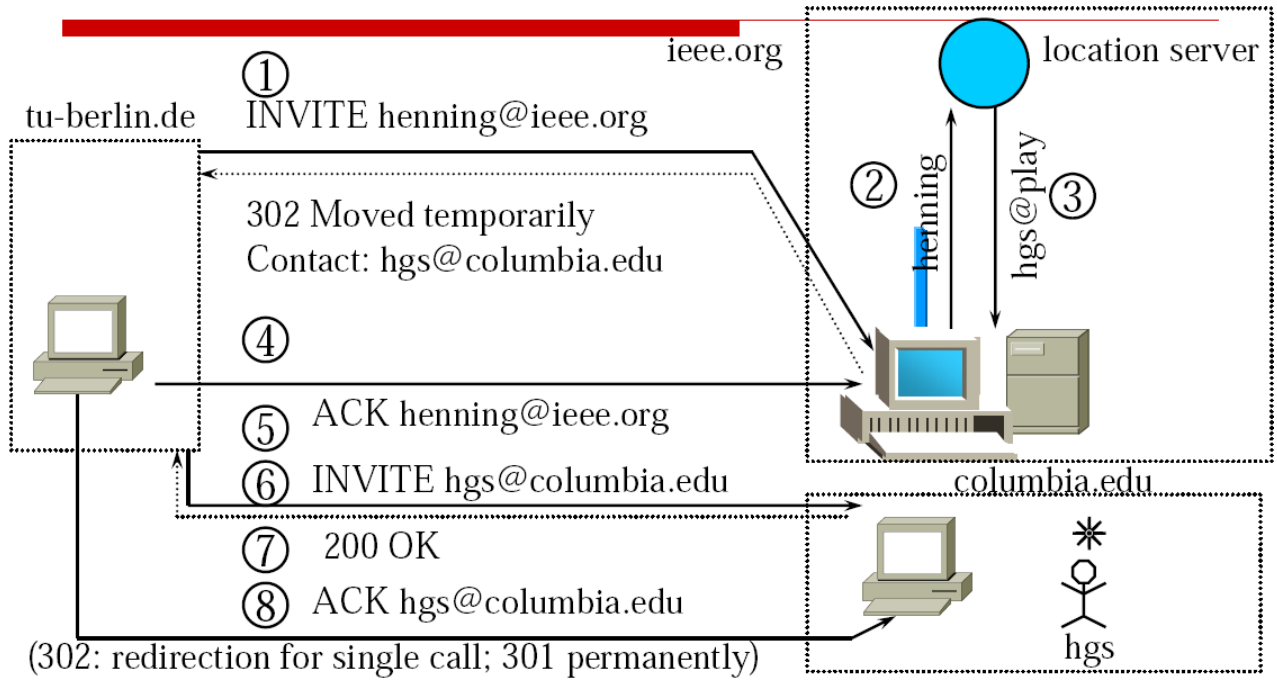
Registrar és Location Server



Redirect Server:

Ezek a serverek csak megadják a hívott user aktuális címét, amennyiben az nem a szokásos helyén tartózkodik. Viszont ennél többet nem tesznek, tehát nem kezdeményeznek automatikusan hívást a megfelelő helyre pl., vagy nem továbbítanak üzeneteket.

Redirect Server



A SIP működése Redirect-módban

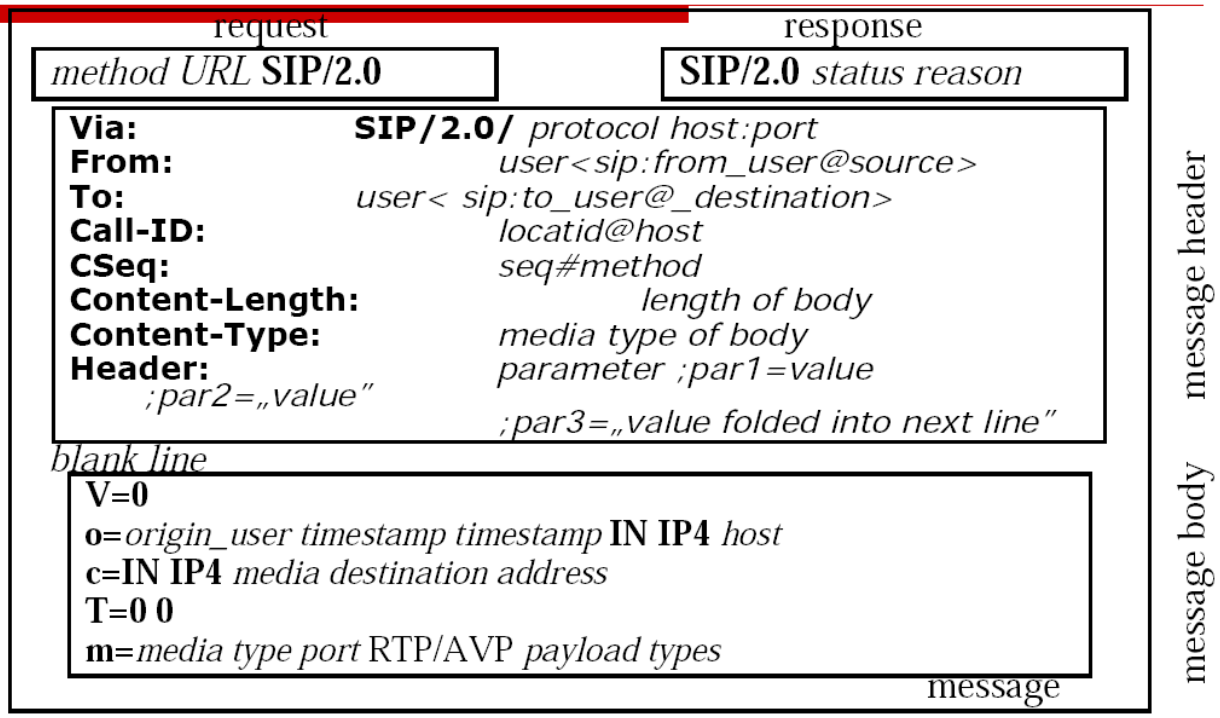
LÉPÉSEK:

- A Redirect server fogadja az INVITE-ot és kapcsolatba lép a location service-szel
- Ha megvan a felhasználó, a redirect server visszaküldi a címet a hívó félnek (nem generál INVITE-ot!)
- A hívó UA ACK-t küld a redirect server-nek
- Az UA INVITE-ot küld közvetlenül a kapott címre
- A hívott UA jelzi a sikert (200 OK) és a hívó UA ACK-t küld

SIP üzenetek:

Igen hasonlítanak a HTTP-üzenetekhez, két kategóriába sorolhatók: Requestek és Responseok. Ezek formátuma megegyezik, az első sort kivéve, üzenettörzset is tartalmazhatnak:
-Session leírás
-HTTP vagy ASCII

SIP üzenetek



SIP Request-ek

- INVITE – hívás kezdeményezése
- ACK – válasz nyugtázása
- BYE – hívás befejezése (és továbbítása)
- CANCEL – keresés és „csengetés" törlése
- OPTIONS – a másik fél által támogatott tulajdonságok
- REGISTER - regisztráció a location service-szel

A SIP az SDP-t (Session Description Protocol) használja a multimédia sessionok leírására. Az SDP-t a SIP-csomagok mint üzenettözset viszik át. Az SDP inkább egy szabványosított leíró formátum, mint protokoll.

SIP-tranzakció: egy request és egy vagy több response.

A SIP előnyei

- Programozhatóság
  - language-based APIs: Parlay, Jain
  - CPL (Call Processing Language)
    generally installed in proxy servers, determine the handling INVITE transactions
  - server-side approaches: cgi and Java servlets
  - including Java applets in SIP requests
- Integration with other Internet services
  - SIP addresses similar to regular URLs
    - Can be embedded in a Web-page, e-mail etc.
  - Calls can be forwarded to a Web-page, e-mail, IRC chat etc.
  - SIP messages can carry any binary or text object

H.323 – SIP összefoglalás

- H.323
  - ITU
  - „távközlési” hívásvezérlő protokollon alapul
  - nem jól skálázható (Megaco/H.248-cal kombinálás segít)
  - állapot-alapú, bináris protokoll
  - web-alapú szolgáltatásokkal nem könnyen kombinálható (gw)
- SIP
  - IETF (újabban 3GPP és ITU is, az NGN szolgáltatásnyújtó platformja, az IMS ezt használja!)
  - IP-központú
  - állapotmentes, szöveges protokoll
  - kombinálhatóság mindenféle web-alapú szolgáltatással

MPLS

Összeköttetés-alapú hálózatok

- A csomópontok csomagtovábbítási tevékenysége (packet forwarding) egyszerű, ezáltal
- gyors működést tesznek lehetővé, és
- a csomagtovábbítási tevékenység teljesen független az útvonalirányítási (vezérlési) tevékenységtől.
- Az összeköttetés felépítése során sokféle szempont figyelembe vételével lehet az útvonalat kialakítani (csak egyszer kell megtenni, így viszonylag körültekintőbben járhatunk el, mintha minden csomagra meg kellene ismételni), így
- könnyen lehet forgalom- és hálózat-menedzsment szempontokat is tekintetbe venni.

Összeköttetésmentes hálózatok

- nem kell várni az információtovábbításnál az összeköttetés kiépítésére
- robosztusabb a meghibásodások kezelésben (útvonalváltozás)
- rendkívül jó „túlélési esélyek”

Hogyan lehetne egyesíteni a a két módszer jó tulajdonságait? Lehet-e csinálni egy olyan hálózatot, amely összeköttetés-alapú is meg nem is?

Erre tesz kísérletet az MPLS!

Vagyis az MPLS köztes módszer az összeköttetés-alapú és összeköttetésmentes megoldások közt. A

legnagyobb probléma az, hogy az IP összeköttetésmentes, és erre kellene alkalmazni az MPLS-t, ami az összeköttetés-alapú verzió felé hajlik. MPLS-nél is, hasonlóan pl. az ATM-hez címkéket kapnak az egyes csomagok, viszont itt lokális éretelmezése lesz ezeknek a címkéknek, nem pedig globális. Címkéket könnyebben és gyorsabban lehet routolni, és általában szinte minden előnyét megkapja így a folyam az összeköttetés alapú hálózatoknak.

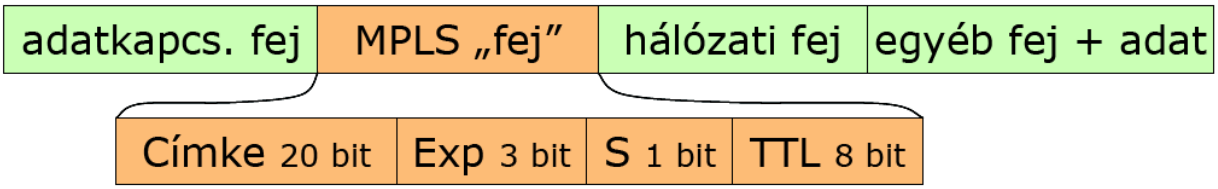
Hogyan működik a két módszer?

- Az összeköttetés-alapú hálózati működés esetén:
  - a felhasználó jelzési információt küld, amelyben benne van a partner globális azonosítója (címe),
  - a csomópont az útvonaltábla segítségével továbbítja a jelzéscsomagot a következő lépést képviselő csomópontnak,
  - feljegyzi saját magának a kiépülő csatornával kapcsolatos lokális azonosítókat (címke),
  - tájékoztatja a következő csomópontot is arról, hogy milyen címkével fogja a vonatkozó adatcsomagokat küldeni
  - A kommunikáció végén el kell bontani a csatornát.
- Mi a helyzet az összeköttetésmentes hálózatban?
  - Jön az adatcsomag (nem kell jelzés), továbbítjuk, de nem jegyzünk fel semmit, mert
  - **nem akarunk állapotnyilvántartást** a csomópontokon (biztosítva ezzel a robosztusságot).

A címkét tartalmazó headert pluszban hozzá kell fűzni minden csomaghoz, ez a plusz header egészen addig utazik a csomaggal, amíg szükség van rá, vagyis mikor kilép az MPLS-es hálózati részből, akkor egyszerűen leválasztják, és továbbroutolódhat „normálisan”. Többnyire a 2. és a 3. osi réteghez tarozó headerek közé szúrják be, 2,5.-ik rétegként.

A címkék elhelyezése

- Közbeiktatott (ún. shim vagy generic) header:

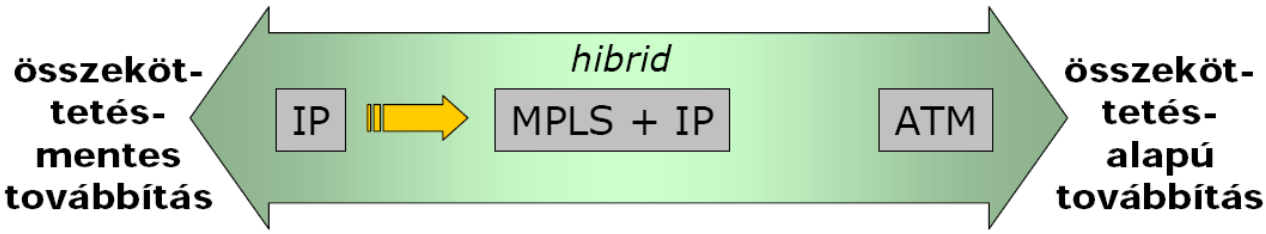


- Ha az adatkapcsolati rétegbeli protokoll összeköttetés-alapú, akkor annak címkéjét lehet használni:
  - ATM cellák esetén:
    - VPI/VCI
  - (Frame Relay keretekben:
    - DLCI)

Azért Multi Protocol, mert többféle 2. és 3. rétegbeli protokollal is képes együttműködni: Ethernettel, ATM-el, Ipv4-el, Ipv6-al, stb. Azért Label Switching, mert itt gyk. a címkék alapján kapcsolják a csomópontok a csomagokat, ami nem igazi routolás.

Az IP, az ATM és az MPLS viszonya

- Az MPLS + IP egyesíti az IP és az összekötetés-alapú csomagkapcsolás legjobb tulajdonságait



- A másik oldalról az ATM (vagy a Frame Relay) nem hozható középre, mint az IP!!
- Javítsunk az IP-továbbítás hatékonyságán!

Milyen az IP-csomagtovábbítás és milyen az MPLS-é?

- **Az IP-CSOMAGTOV. NEM ELÉG HATÉKONY**!
- Az IP-ben az összeköttetésmentes működés miatt mindegyik router függetlenül végez lépésenkénti (hop-to-hop) csomagtovábbítási döntést
  - felhasználva az IP- fej címét (32 bit a v4-nél, 128 bit a v6-nál)
    - a fej viszont sokkal több információt hordoz, amit nem használunk
- Az MPLS felosztja a címteret *forward equivalence class* (FEC)-ekre (a továbbítás szempontjából azonosan kezelendő csomagok osztályaira)
  - és azoknak megfelelő, rövid, helyi érvényű címeket, *címkéket (label)* használ a továbbításra

Az MPLS tehát amellett, hogy a címkékkel gyorsabban is tud dolgozni, össze is fogja csoportokba, FEC-ekbe (Forward Equvalence Classokba) azokat a csomagokat, amiket hasonló módon kell majd továbbítani, ezzel is növeli a hatékonyságot.

Az MPLS alapvető jellemzői

- mechanizmusokat határoz meg a különféle összefogottságú csomagfolyamok kezelésére
- függetlenül működik a 2. és 3. rétegbeli protokolloktól
- módszert biztosít az IP címeknek a leképezésére az egyszerű, fix hosszúságú címkékre
- csatlakozik a használt protokollokhoz (pl. RSVP, OSPF)
- együttműködik az IP-vel, az ATM-mel, és a Frame Relay 2. rétegének protokolljával

Hogyan működik az MPLS?

- A címke-kapcsoló-router (Label Switching Router = LSR) a csomagokat a rájuk „ragasztott” fix hosszúságú címkék alapján továbbítja:
  - Ez „mutat” a kimenő interfészre
- Az LSR átírja a címkét, mielőtt elküldi a csomagot
- Ugyanígy továbbítják az ATM kapcsolók a cellákat
  - lényegesen egyszerűbb, mint a leghosszabb egyezés
  - az LSR adott teljesítmény esetén olcsóbb a hagyományos routernél
  - a csomagtovábbítási döntések összetettebbek lehetnek
  - csomagfolyamok összevont kezelése is lehetséges

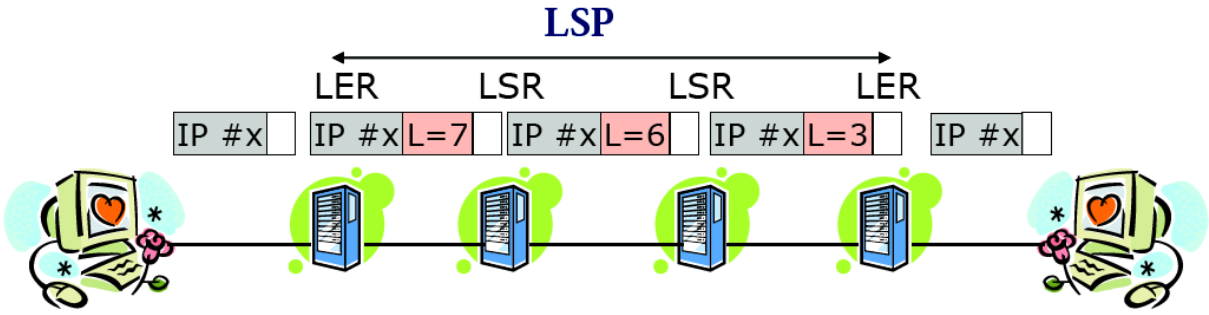
Azt az utat, ami egy-egy csomat címkekapcsolva bejár, LSP-nek, Label Switched Path-nak hívjuk. Ezen az úton LSR-ek, Label Switching Routerek és LER-ek, Label Edge Routerek találhatóak.

LSR: az MPLS gerinchálózaton vannak elhelyezve.

LER: elsődleges eszköze a csomagcímkézésnek, vagyis itt dől el melyik csomag milyen címkét kap belépéskor, ezek az MPLS hálózat szélein helyezkednek el (edge). Az egyes címkék kiválasztására a szabványosított protokoll az LDP (Label Distribution Protocol), de alkalmas lehet más routing protokoll is, vagy az RSVP. A BGP-t és az RSVP-t kibővítették úgy, hogy a saját adatai mellett szállíthat címkeinformációt is, ezért felelnek meg.

Példa a címkekapcsolásra

Útvonalirányítás a határon, kapcsolás a gerincben



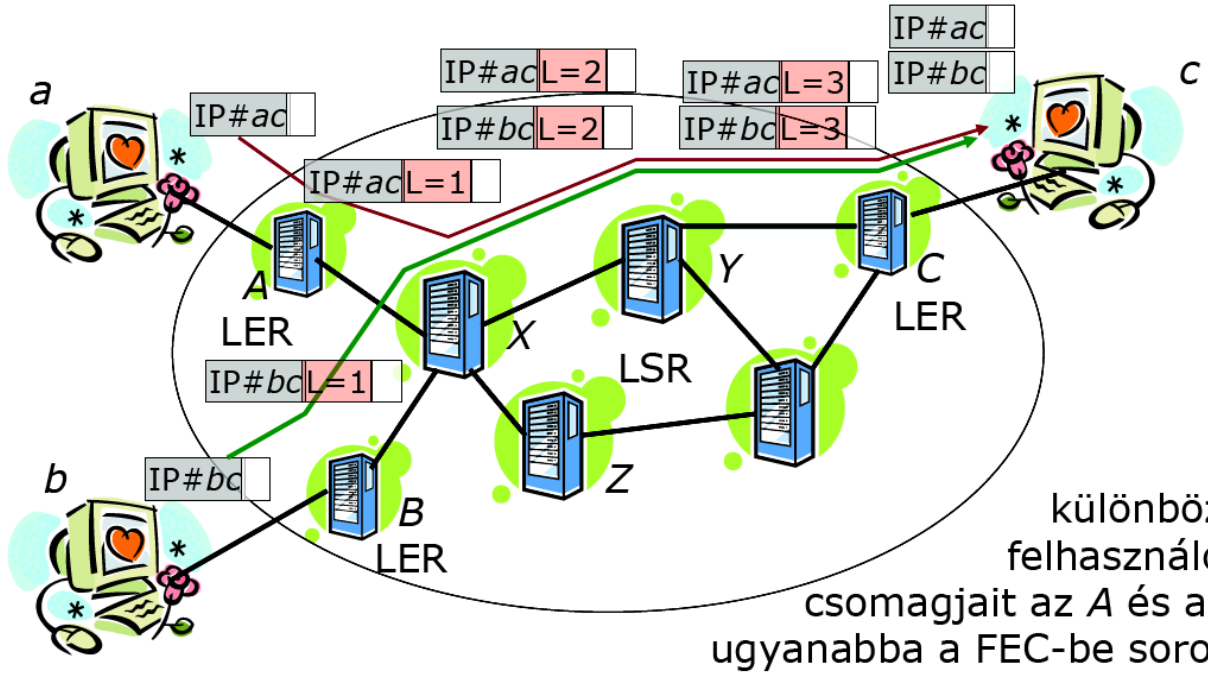
IP cím	KI
152.66/16	7
kijelölés	

BE	KI
7	6
csere	

BE	KI
6	3
csere	

BE	IP cím
3	152.66/16
eltávolítás	

Még egy példa a címke-kapcsolásra



különböző felhasználók csomagjait az *A* és a *B* ugyanabba a FEC-be sorolta

A FEC és a címkék

- FEC:
  - a forward equivalence class (FEC) a csomagoknak egy csoportját képviseli, amelyek továbbítása azonos
  - a besorolás csak egyszer, a belépésnél
- Címkék:
  - A címkét a csomag magával viszi egy 2. réteg jelölésként
  - A közbenső router a címke tartalma alapján határozza meg a csomag következő lépését
  - Ha egyszer egy csomagot „felcímkéztünk”, akkor útjának hátralévő részén a csomópontok már ennek alapján kezelik
  - A címkék értéke lokális jelentésű

Címkekapcsolat utak létrehozása: MPLS képességű eszközök egy csoportját MPLS tartománynak nevezzük – domain. Egy ilyen tartmányon belül az egyes FEC-ekhez (csoportok hasonló kapcsolási tulajdonságokkal) rendelünk LSP-ket (MPLS-út). Az egyes LSP-ket az adatátvitel megkezdése előtt kell persze létrehozni, ami az erőforrások lefoglalását is jelenti ilyen szempontból.

Két módszer létezik LSP létrehozására:
lépésenkénti: minden LSR (címkekapcsoló router) függetlenül választ következő lépést egy FEC számára, bármilyen routing-protkollal -> lépésenként megy végbe a felépítés.

explicit: forgalommenedzsment vagy QOS szempontok szerint épül fel az útvonal.

GMPLS

Ez az MPLS általánosított változata, ami nem a csomagok továbbítását hanem az LSP-k létrehozását helyezi előtérbe. Vagyis így nem csak csomagkapcsolt hálózatokra alkalmazható a címkekapcsolás:

A **GMPLS** (általánosított MPLS)

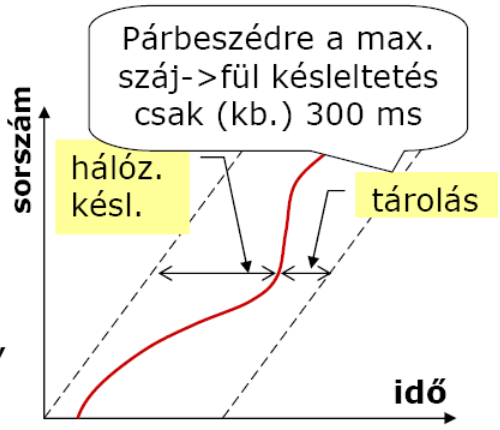
- Lehet, hogy az MPLS-nek nem is a csomag-továbbítási módszer a lényege?
  - az LSP-k létrehozása a jelentős!
- GMPLS: címke-kapcsolt utak létrehozása **nem csak** csomagkapcsolt hálózatokban:
  - PSC (Packet Switch Capable interfaces)
  - L2SC (Layer-2 Switch Capable interfaces)
  - TDM (Time-Division Multiplex Capable intrfs)
  - LSC (Lambda Switch Capable interfaces)
  - FSC (Fiber-Switch Capable interfaces)

GMPLS összefoglaló

- A csomaghálózat címke-kapcsolt útjai (LSP) a csomaghálózat csomópontjait összekötő linkeken jönnek létre
- Ezeknek a linkeknek a kialakítása idő-, hullámhossz-, és térosztású fizikai hálózatokon történik
- A fizikai átviteli utak vezérlése (jelzése) történhet meg a GMPLS révén a csomaghálózat hálózati szintjéről

A szolgáltatásminőség (QoS)

- Hálózati szolgáltatást igénylő alkalmazások:
  - Valódi idejű, átvitel- és késleltetés-érzékeny
  - Nem-valódi idejű, elasztikus, átvitel- és késleltetéstűrő
- Példa: beszédalkalmazás
  - Minták 125 µs-onként
  - Csomagok 20 ms-onként
    - Továbbítás ingadozik:
      - Határidőn belül -> tárba
      - Határidőre -> „lejátszás”
      - Határidőn túl -> eldobás



Hogyan teljesítsük a követelményeket?

Valódi idejű, nem rugalmas átvitel esetén tehát biztosítani kell a forgalmat bizonyos szempontok szerint, tehát QOS nyújtása szükséges szemben a rugalmasabb igényű átvitelekkel.
Az ATM-ben a QOS a szabvány szerint támogatott, különböző forgalom-kategóriákba sorolhatjuk be az aktuális adatfolyamot.
Ennek megvalósítása érdekében a forgalmat különböző leírók segítségével kell tudnunk jellemezni, illetve QOS paramétereket is be kell vezetni. A tórolódásvezérlés – mind a reaktív, mind a preventív – szintén része már az ATM-nek.

4 különböző QOS kategória az ATM-ben:

ATM QoS-szolgáltatások

- CBR –continuous bit rate
  - Forgalomleírók: PCR, CDVT
  - QoS jellemzők: maxCDV, MaxCTD, CLR
- rt-VBR – real-time variable bit rate
  - Forgalomleírók: PCR, CDVT, SCR, MBS
  - QoS jellemzők: maxCDV, MaxCTD, CLR
- nrt-VBR – non-real-time variable bit rate
  - Forgalomleírók: PCR, CDVT, SCR, MBS
  - QoS jellemzők: CLR
- UBR – unspecified bit rate
  - PCR specifikálva van, de nem használja a CAC és a policing
  - Nincsenek QoS paraméterek jelezve

CBR: állandó bitsebesség biztosítása
rt-VBR: valósidejű, változó bitsebességű adás, pl. videókonferencia igényelheti
nrt-VBR: nem valósidejű, változó bitsebességű adás, pl. film nézése hálózaton keeresztül, ahol a késleltetés nem olyan fontos szempont
UBR: nem meghatározott bitsebességű, a rendelkezésre álló sávszélességet használja ki

Összeköttetésmentes hálózatoknál QOS:

Megoldaható **IntServ (Integrated Services)** módszerrel, ez finomabb felbontással dolgozik, az egyes csomagfolyamokra adhatunk meg QOS feltételeket, illetve **DifServ (Differentiated Services)** módszerrel, ez a durvább felbonátsú megoldás, ekkor folyam-osztályokra határozunk meg QOS-t.

A QoS módszerek „viszonya”

- Best-Effort:
  - megkülönböztetés nélkül továbbítja a csomagokat
- DiffServ:
  - különböző kiszolgálást biztosít csomagfolyamok csoportjainak (osztály)
- IntServ:
  - a csomagfolyamokat egyedileg szolgálja ki, igényeiknek megfelelően!

IntServ:

A szolgáltatásosztályok:
-Best Effort , ami nem garantál semmit, csupán a lehetőséges szerinti legjobb továbbítást
-Guaranteed Quality, amely garantált korlátokat határoz meg az egyes csomagok késleltetési idejére
-Controlled Load, amely a csomagokat függetleníti az egyéb forgalmaktól ill. ezekből eredető terhelésektől, azt a hatást igyekszik kelteni, mintha csak a kiválasztott csomagok lennének a hálózaton.

Az alkalmazások igényei és az IntServ-szolgáltatások

1. „Elastic applications”
   - nincs késleltetési vagy bármilyen korlát
   - tipikus TCP/IP adat-alkalmazások

   Best effort service

   3 alosztály:
   - interactive burst (pl. WEB)
   - interactive bulk (pl. FTP)
   - asynchronous (pl. e-mail)

   ATM-analógia: UBR

Az alkalmazások igényei és az IntServ-szolgáltatások (folyt.)

2. „Real-time tolerant (RTT)” alkalmazások
   - gyenge késl. korlátok, alkalmanként csomagvesztés megengedett
   - pl.: tárolásos videóalkalmazások

   Controlled load service
   - átlagos késleltetés garantált
   - mennyiségi biztosítékok nélkül

   ATM analógia: nrt-VBR
3. „Real-time intolerant (RTI)” alkalmazások
   - min. késleltetés és késleltetés-ingadozás
   - Pl.: beszéd, élő videókonferencia

   Guarranteed service
   - átvitelisebesség- és késleltetéskorlátok

   ATM analógia: rt-VBR, CBR

Használt mechanizmusok:

Az alkalmazás megadja a hálózatnak az általa generált forgalom jellemzőit a forgalomleírók segítségével. Ezt figyelembe véve eldönti a hálózat, hogy elbírja-e ezt a terhelést, tudja-e biztosítani a

kért QOS feltételeket (beengedés-szabályozás, Admission Control). Ezt követően megtörténik az erőforrások lefoglalása, pl. az RSVP-protokoll használatával IntServ-ben, illetve összeköttetés alapú hálózatok esetén a hívásvezérlő protokollokkal.
Traffic Policing alkalmazásával a hálózat biztosítja azt, hogy a forrás forgalma ne térjen el a megadottól. Scheduling-el a hálózat az egyes csomópontokon meghatározza a csomagok kiszolgálásának sorrendjét, a torlódást elkerülendő.

Csomagfolyam-specifikáció:

A csomagfolyam-specifikáció (flowspec)

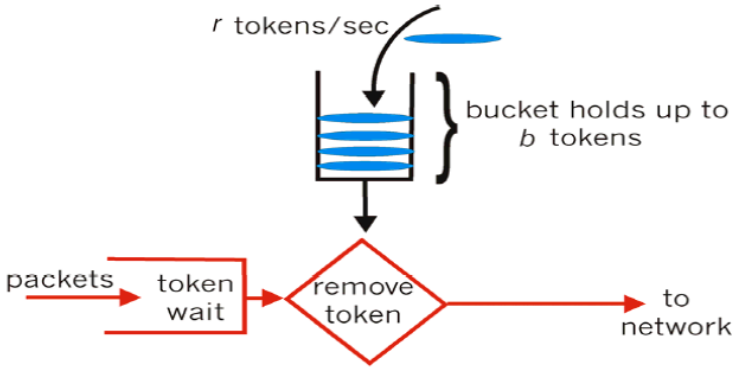
- Két eleme van:
  - *TSpec:* leírja a folyam forgalmi jellemzőit, informálva ezzel a hálózatot a szükséges sávszélességről → lehetővé téve a döntést a beengedés-szabályozás részére
    - *TSpec: a Traffic Specification-ból jön*
  - *RSpec:* leírja a kiszolgálási igényt (pl. *controlled load*, vagy késleltetési korlát)
    - *RSpec: a Request Specification-ból*

Vagyis a Tspecben megadja az alkalmazás, hogy hogyan fog kinézni az, amit ad, Rspecben pedig megfelelő elbánásmódot kér a csomagfolyamnak.

Forgalomleírás és policing

A policing és a forgalomleírás eszköze: a token bucket (tokenvödör)
A vödör formálja (szűri) a forgalmat: küldhetünk *b* méretű börsztöt, de az átlagsebesség csak *r* lehet
Ezeket a paramétereket (is) használjuk a TSpec-ben

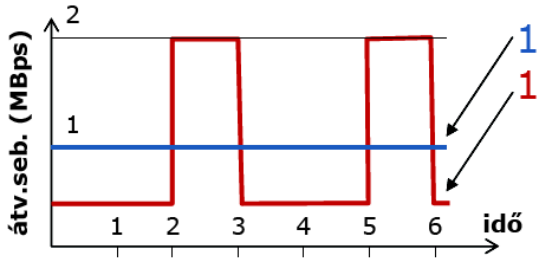
- A vödörbe *r* sebességgel töltődnek a tokenek
- Legfeljebb *b* token lehet benne, ha már tele van, a beérkező tokenek törlődnek
- Ha egy *n* hosszú csg érkezik, kivesz *n* tokent a vödörből (ha van annyi) és továbbításra kerül
- Ha nincs: ...



Token Bucket működése: A Token Wait pufferbe érkeznek az elküldendő csomagok, de minden csomag csak akkor kerül elküldésre, ha megfelelő mennyiségű token van a fent látható vödörben. Ezek a tokenek gyakran méretet jelképeznek, nem csomagszámot, tehát pl. ha éppen 45 Kbytenyi token van a vödörben, de egy 46 Kbyteos csomagot kell elküldeni, akkor abban a menetben nem lehetséges a teljes csomagot elküldeni, a maradék majd csak akkor megy el, ha újra lesz elegendő token hozzá a vödörben, így biztosítja a rendszer az állandó kimeneti sebességet.

Token bucket az IntServ-ben

- Az IntServ *b*-t byte-ban,
  r-et pedig byte/sec-ban méri
  - Igen tág határok a *b*-re és az *r*-re
- A TB elég „sajátos” forgalom-specifikáció



1 MB/s állandó (*r* =1 MB/s, *b* =1MB)
1 MB/s átlag (*r* =1MB/s, *b* =1MB)

Sőt! Egy csomagfolyam többféle TB paraméter-párral is leírható! Kerülni kell a túlzott foglalást!

Beengedés-szabályozás (admission control)

- Megvizsgálja a forgalomleírást (*TSpec*) és a kiszolgálási igényt (*RSpec*), ezután dönt az igényelt szolgáltatás teljesíthetőségéről
- Eltérő a *Controlled Load* és a *Guaranteed*:
  - Az előbbi esetén heurisztikus is jó lehet:
    - Például eddig a hasonló esetek jók voltak, engedjük meg most is, vagy korábban hasonló esetben már nem felelt meg a kiszolgálás, ezért tagadjuk meg
  - *Guaranteed* esetben ennél sokkal szigorúbb vizsgálat szükséges
- Az admission control konkrét módszerei nem képezik az IntServ specifikáció részét

RSVP – Jelzésátvitel:

Jelzésátvitel – az RSVP

- **R**esource Re**S**er**V**ation **P**rotocol
  - *R.S.V.P.: "Répondez/Réservez s'il-vous-plaît"*
- Lényeges jellemzői :
  - Vevőoldali erőforrás-foglalás:
    - Nem az adó „erőlteti" a képességeit, hanem a vevő foglalja le azt az igényei szerint *
  - „Soft state": igények „elhalnak", ha nem …
    - nem igényli a session explicit lezárását **
    - viszont időnként „frissíteni" kell a foglalást
- Eredmény: robosztusság
  - * kedvező a többescímzésnek (multicast)
  - ** megkíméli a hálózatot a CPE-k hibáitól
- Specifikáció: az RFC 2210-ben
- Az RSVP-t nemcsak az IntServ használja (említettük a H.323-nál is)

Egy protokoll aminek segítségével erőforrásokat igényelhetünk a hálózaton, a QOS biztosítása érdekében. Fontos, hogy NEM az adó igényli ezeket az erőforrásokat, hanem a vevő, és az igény fenn kell tartani, bizonyos időközönként (pl. tipikus érték a 30 sec) meg kell újítani a kérelmet, különben timeout lesz és újra le kell foglalni mindent.

RSVP – alapvető jellemzők, lépések

- Alapötlet:
  - a vevőnek ismernie kell, hogy az adó milyen forgalom generálására képes → tudnia kell az adó *TSpec* jellemzőjét, valamint
  - tudnia kell a forgalom (csomagok) útját, és
  - így erőforrást foglalhat az út mindegyik csomópontjánál

1. Az adó *PATH* üzenetet küld, benne a *TSpec*
   - az úton lévő valamennyi csomópont feljegyzi
2. A vevő *RESV* üzenettel foglal, benne *RSpec*

RSVP – „út”: egy adó <-> egy vevő

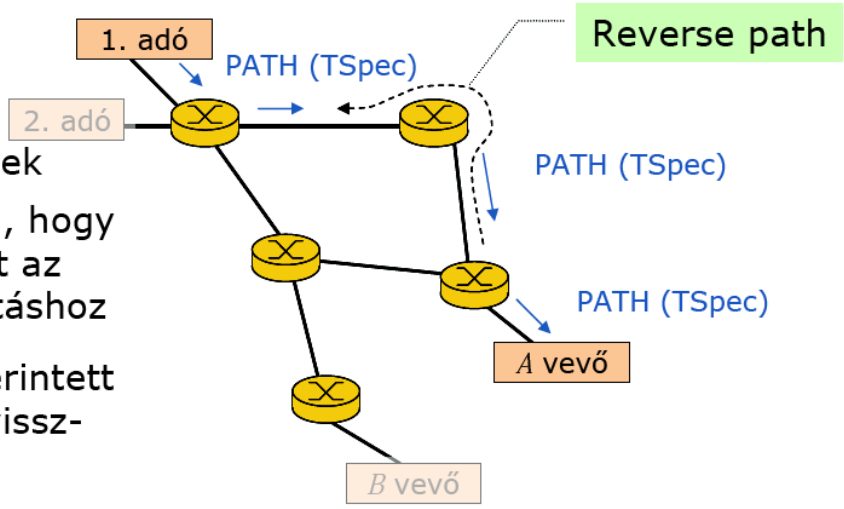
- Első fázis:

•a vevőnek ismerni kell az adást (**TSpec**)

•PATH üzenet a vevőnek

•a vevőnek tudnia kell, hogy hol foglaljon erőforrást az általa kívánt szolgáltatáshoz

•a PATH üzenet által érintett routerek feljegyzik a vissz-irányt (reverse path)



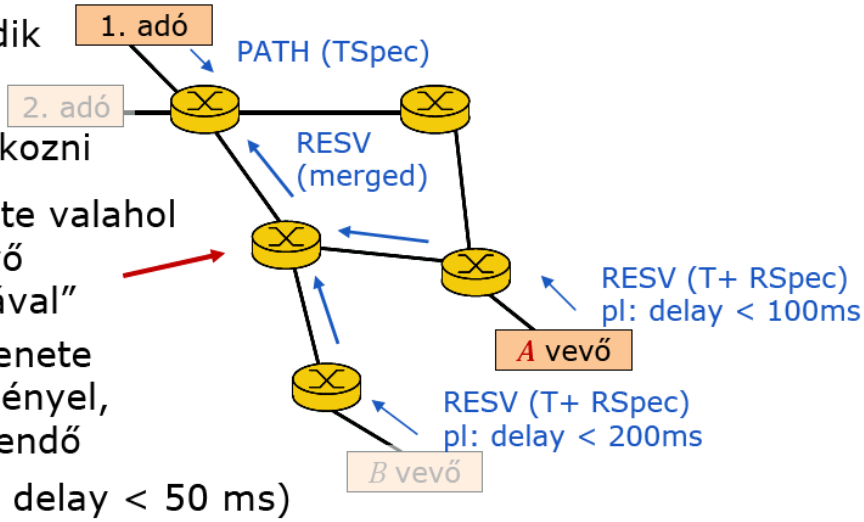
A vevő tehát a forrástól kapott PATH üzenet alapján visszafele végigmegy az útvonalon és elvégzi a lefoglalásokat, mindezt még a tényleges adatforgalom elindítása előtt. Az egyes routerek nem biztos, hogy teljesíteni tudják a kért QOS feltételeket, ekkor hibaüzenetet küldenek vissza a vevőnek.

Az adás során előfordul, hogy egy router valamilyen hiba folytán kiesik, ekkor új útkeresés indul meg, ahol újra le kell foglalni az erőforrásokat az előzőhöz hasonló módon. Eközben a kiesett router, ha még működőképes, csak pl. egy kábel hibásodott meg, törli a foglalást a timeout lejártával.

RSVP: egy adó <-> több vevő

Új vevő csatlakozása:

- az *A* vevő már működik egy ideje
- a *B* vevő kíván csatlakozni
- a *B* vevő RESV üzenete valahol találkozni fog az *A* vevő kiszolgálását végző „fával”
- ha a *B* vevő RESV üzenete kevesebb erőforrást igényel, mint *A*, akkor nincs teendő
- ellenkező esetben (pl delay < 50 ms) az új foglalást tovább kell küldeni, amely azonban mindenképpen összevont (merged) lesz

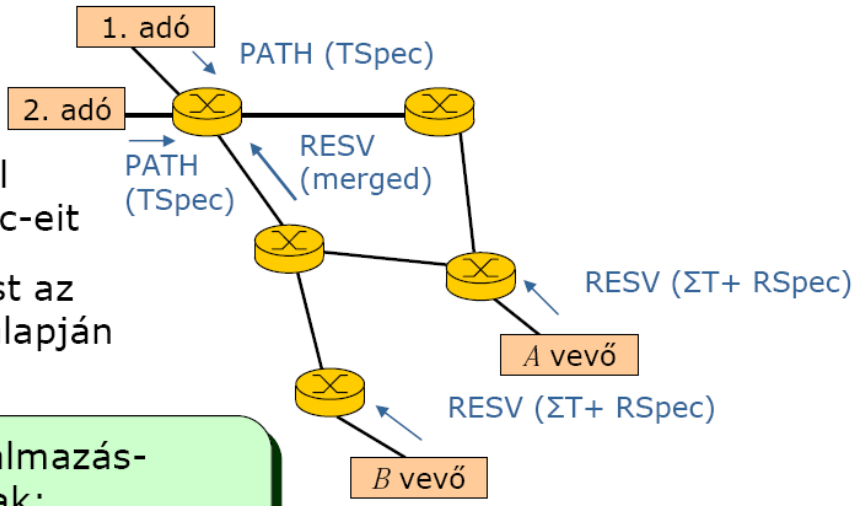


Tehát egy már kiépített, erőforásokat lefoglaló útvonalat, illetve annak egy részét legalábbis felhasználhatja egy másik útvonal. Ha ez az új kapcsolat kisebb QOS igényű, akkor nem szükséges az egybevágó útrészekhez tartozó routereknél újabb erőforrás-lefoglalás.

RSVP: több adó <-> több vevő

Konferencia:

- a vevőknek össze kell gyűjteni az adók TSpec-eit
- a vevőknek a foglalást az összesített TSpec-ek alapján kell kialakítani



A TSpec-ek alkalmazás-specifikusak:
Az **RSVP különböző foglalási “stílusokkal” működik**

Visszatekintés: az IntServ-ben használt mechanizmusok

- a felhasználó (alkalmazás) specifikálja a forgalmát a hálózat számára
  - *ATM-nél:forgalomleírók – traffic descriptor-ok*
- meghatározott szolgáltatást kér a hálózattól, melynek alapján a hálózat eldönti, hogy tudja-e ezt vállalni, beengedi-e az új folyamot a hálózatba (*admission control*, beengedés-szabályozás)
  - *ATM-nél is van, kulcskérdés*
- információt cserélnek az erőforrásfoglalásról (jelzésátvitel, az IntServ-ben az *RSVP-protokoll* segítségével)
  - *áramkörkapcsolt és összeköttetés-alapú hálózatokban a hívásvezérlő protokollok csinálhatják*
- a hálózat gondoskodik arról, hogy a forrás forgalma ne térjen el a megadottól (*traffic policing*)
- a hálózat meghatározza a csomagok sorbaállítását és kiszolgálását a csomópontokon (*scheduling,* ütemezés)
  - *megnéztük az általános módszereknél*

Csomagkezelés:

1. Csomagminősítés:
   Az egyes csomagokat hozzá kell rendelni az arra vonatkozó foglaláshoz. Ezt a csomag alábbi tulajdonságai alapján lehet megtenni: forrás címe, cél címe, cél posrtszáma, forrás portszáma.

2. Csomagkezelés a sorokban:
   Guaranteed Service esetén:
   WFQ-val (Weighted Fair Queue) biztosítja a végpontok közti késleltetést.

IntServ scalability (skálázhatóság, „növekedési” képesség)

- Bár az IntServ nagyon jelentős előrelépés a *best effort* kiszolgáláshoz képest,
- nem teljesíti az Internet egyik alapvető célkitűzését: a növekedési képességet
  - *best effort* kiszolgálásnál nincs folyamhoz kötött állapot-nyilvántartás a routereknél
  - így a hálózat növekedésével elég, ha a routerek a linkek sebességének növekedésével lépést tudnak tartani
- Az IntServ-nél: lényegében összeköttetés-alapú szolgáltatást csináltunk
  - annak minden nehézségével együtt,
  - de anélkül, hogy annak előnyét, a lokális címzést kihasználtuk volna

DiffServ:

A leglényegesebb eltérés az Integrated Serv.-hez képest, hogy itt már csak forgalom-osztályokhoz lehet erőforrásokat rendelni, azokból is viszonylag kisszámút lehet deifinálni. Tehát nem lehet minden adatfolyamhoz explicit megmondani hogy milyen QOS szabályok érvényesek rá. Éppen ezért viszont csökken a QOS-hez szükséges számítási teljesítmény is, és így alkalmasabb a gerinchálózatok számára, ahol nagy mennyiségű adatot kell kezelni.

NINCS szükség mindenféle külön jelzésrendszer/protokoll kialakítására, mint pl. az RSVP az IntServnél, hanem elég csak a csomagfejben néhány bittel jelezni a csomópontoknak, hogy az aktuális csomagfolyambeli csomag mely osztályba tartozik.

DifServ hálózat felépítése:

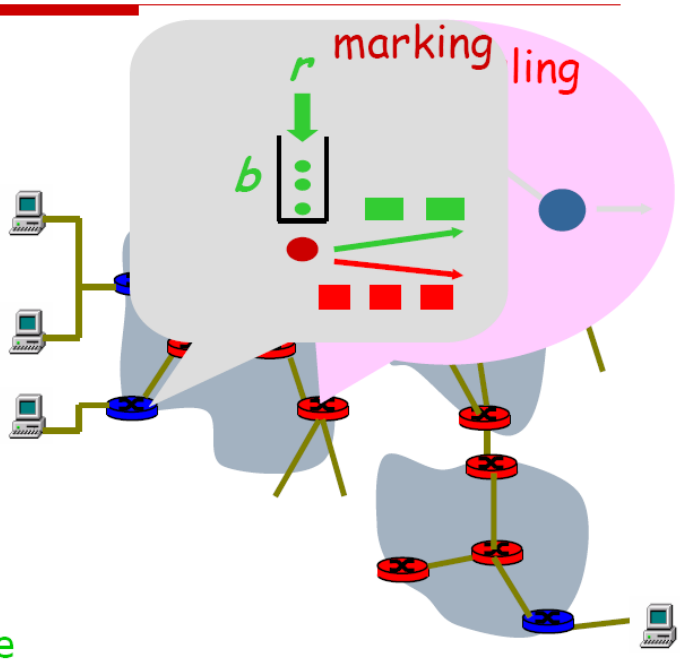
A DiffServ architektúrája

Edge router:

- folyamonkénti forgalom-menedzselést végez
- megjelöli a csomagokat in-profile ill. out-profile -ként

Core router:

- osztályonkénti forgalom-menedzselést végez
- pufferelés és ütemezés a széleken történt megjelölésnek megfelelően
- elsőbbség adása az in-profile csomagoknak



A csomagokhoz az adatfolyamuk szerint profilokat rendelünk a DifServ hálózat szélén (gyk. belépésnél). A profilban vannak meghatározva, hogy milyen paraméterek szerint megy át a hálózaton a csomag, pl. ütemezésnél a vödör mérete.

Az adatfolyam folyamatos policing alatt áll, ha egy csomag nem konfrom, vagyis nem megfelelő valamilyen paramétere, akkor azt ha lehet Shapeing-el alakítja a hálózat, vagy eldobja, vagyis a forrásoknak be kell tartani az egyeztetett adásmódot.

DifServ hálózatokban a csomópontok nem működnek együtt közvetlenül mint IntServnél, PHB van (Per-Hop Behavior), ugrásonként máshogy viselkedhetnek. Az egyes viselkedésmódok (PHB-k) közt a csomópontok az IP csomag ToS mezője alapján választhatnak (legalábbis Ipv4-nél). Ezek a bitek a DSCP (DiffServ Code Points).

Két fontos PHB:

Expedited Forwarding (EF):

A csomagok ekkor minimális késleltetéssel és alacsony csomagvesztéssel továbbítódnak. Beszéd, videó-alkalmazásoknál használható jól, többnyire a bemenő forgalom érkezési sebességét nem korlátozzák a csomópontokon, annyi jöhet, amennyit a link elbír. Ez az egyik legegyszerűbb mód, mindössze prioritást kell adni ezeknek a csomagoknak.

Assure Forwarding (AF):

Bonyolultabb PHB, 14 -féle mód közül választhatunk a 6 bittel, a különböző módokat egy táblázatból lehet kiválasztani. A szabvány szerint a táblázat tartalmazza, hogy az egyes módok milyen forgalmi

paraméterek szerint továbbítják a csomagokat, ezeket szvsz nem kell ismerni és nincs is részletezve sehol.

Assured forwarding (AF) alosztályok

Drop Precedence	Class 1	Class 2	Class 3	Class 4
Low *Medium* *High*	010 000	... ...	... ...	... ...

DifServ előnyei:

- 3. rétegbeli megoldás, vagyis IP felett is működik (plusz bármilyen 2. réteg felett nyilván)
- nincs szükség minden csomóponton jelzésre
- a gerenchálózatoknak így nem kell nyilvántartást vezetniük minden csomagfolyamról, így csökkentjük a terhelést
- a hálózat szélén bonyolítja le a komplikáltabb műveleteket (pl. profilokba sorolás), ezkkel a műveletekkel nem kell már foglalkozni a core csomópontoknál

Az IntServ és a DiffServ összehasonlítása

Jellemző	IntServ	DiffServ
Szolgáltatások kezelési módja	Végpontok közötti	Helyi (lépésenkénti)
Szolgáltatások kezelésének hatóköre	Egyedi- vagy többes-című út	Bárhol a hálózatban
A skálázhatóság (scalability) korlátja	Csomagfolyamok száma	Szolgáltatás-osztályok száma
Számlázás alapja	Forgalom- és QoS jellemzők	Forgalmi osztály használat
Forgalommenedzsment fajtája	Hasonló, mint az áramkörkapcsolt	Hasonló, mint az IP hálózaté
Tartományokon átívelő megvalósítás feltétele	Többoldalú megállapodás	Kétoldalú megállapodás

DiffServ és IntServ (folyt.)

1. Az adó PATH üzenete eljut a vevőhöz
2. A vevő RESV válasza megérkezik az adóoldali *Edge router*-hez
3. Ez a router a DiffServ *Bandwidth Broker*-éhez fordul egy igénnyel
4. Ha a *BB* elfogadja, akkor értesíti az adót (RESV)
5. Az adó megkezdi a csomagküldést, az *Edge router* jelöli a csomagokat a megfelelő DSCP (*differentiated services code point*) értékkel
6. Az *Edge router* ellenőrzi az adó forgalmát: megfelel-e a szolgáltatási szerződésnek (SLA - *service level agreement*). „Kilógó” csomagokat vagy eldobja, vagy alacsony prioritásúnak jelöli.
7. A *Core routerek* csak a DSCP-t nézik, végzik a PHB-t

QoS-biztosítási módszerek összefoglalása

- QoS összeköttetés-alapú hálózatokban
  - ATM
- QoS összeköttetésmentes hálózatokban (módszerek a hálózati rétegben, az IP-protokoll alapján)
  - IntServ: folyamonkénti QoS-biztosítás, az összeköttetés-alapú megoldásokhoz hasonlóan, nem jól skálázható
  - DiffServ: osztályonkénti QoS-biztosítás, alkalmas gerinchálózati megvalósításra
  - A végpontok közötti (end-to-end) QoS-hez mindkét módszerre szükség van
- Legközelebb: az IP hatékony kombinálása az összeköttetés-alapú elvvel, a címkekapcsolással: MPLS (Multi-Protocol Label Switching)