

Internet szolgáltatások és alkalmazások

SLA, QoS (QoE), átviteli minőség...

Tartalom

- Átviteli szolgáltatásminőség az interneten
 - alap IP
 - integrált szolgáltatás (Integrated Services – IntServ)
 - differenciált szolgáltatás (Differentiated Services – DiffServ)
 - Multi-protokoll címkekapcsolás (Multi-Protocol Label Switching – MPLS)

- Áttérés nyílt rendszerekre
 - OSA – Open Service Access (nyílt szolgáltatás hozzáférés)

IntServ

RSVP

Tartalom

- Átviteli szolgáltatásminőség az interneten
 - (alap IP)
 - **integrált szolgáltatás (Integrated Services – IntServ)**
 - differenciált szolgáltatás (Differentiated Services – DiffServ)
 - Multi-protokoll címkekapcsolás (Multi-Protocol Label Switching – MPLS)

Integrált szolgáltatás (IntServ)

- **IntServ** (**integrált szolgáltatás**) volt az első próbálkozás (1994-ben) a QoS biztosítására IP hálózatokban.
- Alapötlet: egyedi alkalmazás példányok erőforrásokat igényelnek a hálózattól.
- Következmény: az alkalmazás forgalmi útvonalán minden útválasztó (router) **folyamankénti forgalomkezelést** kell alkalmazzon.
- Egy IntServ **folyam** (**flow**) definíciója: *Azonos QoS kezelést megkövetelő csomagok osztályozható halmaza a forrástól a célig.*
- Az IntServ jelzésátviteli protokollja az **RSVP** (**Resource Reservation Protocol** – erőforrásfoglaló protokoll)
 - az *egyedüli* IP-alapú jelzésátviteli protokoll

RSVP Resource Reservation Protocol

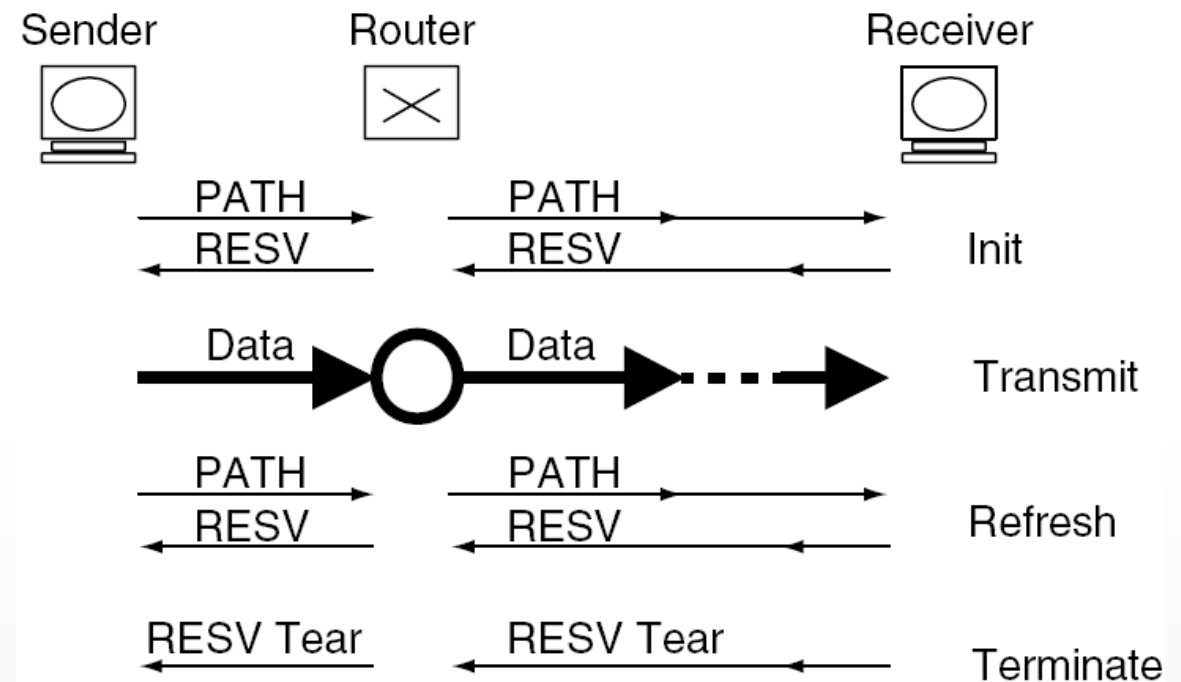
- Erőforrás foglalást állít fel egy egyirányú adatfolyamnak.
- „*Hop-by-hop*” protokoll
 - Az útvonal mentén az összes útválasztóval kommunikál.
- Független az útvonalválasztó protokolltól
 - így nem implementálható QoS-alapú útvonalválasztás(!)
- *Vevő-orientált* jelzésátviteli protokoll
 - Az adatküldő hirdeti a QoS követelményeket.
 - A hirdetés üzenet megkeresi az útvonalat a hálózaton keresztül a célig (vevő).
 - A lefoglalás visszafelé a vevőtől kiindulva történik.
- Legnagyobb problémája: **állapotmenedzsment szükséges**
 - „soft-state” megközelítés: az állapotokat frissíteni kell.

RSVP működése

- A PATH üzenet tartalmazza a QoS követelményeket.
- A RESV foglalja le ténylegesen az erőforrásokat *minden egyes* csomópontban.
- Periodikus frissítések szükségesek a foglalás fenntartásához.

- **RSVP problémái:**

- meglehetősen komplex (pl., többesadás támogatás)
- skálázhatósági problémák (nagy folyamatszám esetében)



RSVP – biztosított szolgáltatások

- Két átviteli szolgáltatást biztosít:
 - Guaranteed Service (**garantált szolgáltatás**)
 - Controlled Load Service (**kontrollált terhelés szolg.**)
 - (lehetne többet is definiálni, egyik sem szabványosított)
- **Garantált szolgáltatás (Guaranteed Service)**
 - **biztos (matematikailag igazolható) határértékeket garantál a vég-vég csomag sorbanállási késleltetést tekintve;**
 - *sávszélességre és késleltetésre* egyaránt garanciát biztosít;
 - intoleráns, *valósídejű* alkalmazásokhoz megfelelő;
 - NEM garantál minimális vagy átlagos késleltetést, dzsittert, stb.
 - szolgáltatási paraméterek: küldési sebesség, borsztméret, megkövetelt késleltetési határérték.
 - **gyakorlatilag nem tesz lehetővé multiplexálást(!).**

RSVP – biztosított szolgáltatások (folyt.)

- **Kontrollált terhelés szolgáltatás (Controlled Load Service)**
 - „a terheletlen hálózatban tapasztalható szolgáltatáshoz hasonló értékeket garantál” (azaz a hálózatban nincs torlódás)
 - Lehetővé teszi az alkalmazások számára, hogy „jó” szolgáltatást kapjanak, de nem „teljes mértékben” garantált az átvitel.
 - Magasabb szintű multiplexelést tesz lehetővé.

- RSVP konklúzió
 - IntServ nem nyújt QoS-alapú útvonalválasztási képességet.
 - IntServ **NEM SKÁLÁZHATÓ** → Gerinchálózatokban nem nyújt megfelelő megoldást (skálázhatóság hiánya)

DiffServ

edge, core, shaping, policing...

Tartalom

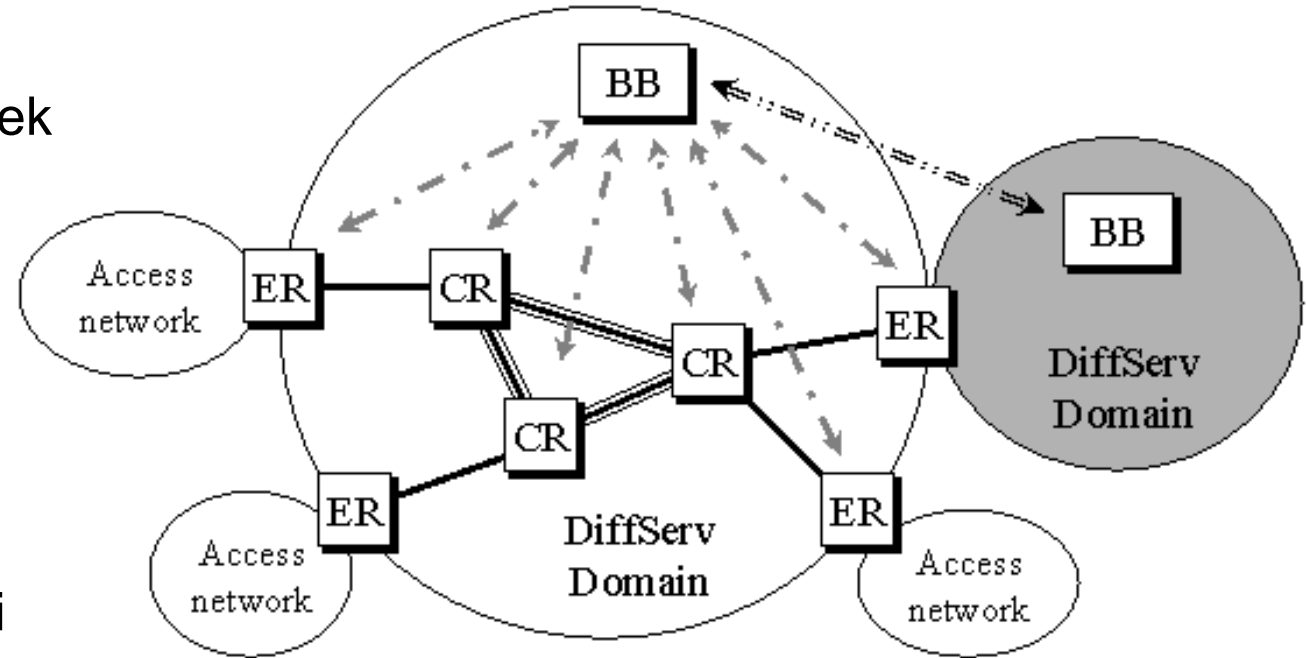
- Átviteli szolgáltatásminőség az interneten
 - alap IP
 - integrált szolgáltatás (Integrated Services – IntServ)
 - **differenciált szolgáltatás (Differentiated Services – DiffServ)**
 - Multi-protokoll címkekapcsolás (Multi-Protocol Label Switching – MPLS)

Differenciált szolgáltatás (DiffServ)

- **DiffServ** – Válasz az IntServ skálázhatósági problémájára
- **Ötlet:** A forgalmat forgalmi osztályok egy halmazába képezzük le a DiffServ hálózat határán.
- **Cél:** A költséges adat-sík funkciókat a hálózat peremén (**edge**) végezzük el, a mag (**core**) útválasztók funkcionalitását nagyon egyszerűnek hagyva.
- DiffServ (**megkülönböztetett szolgáltatású**)...
 - feladja a vég-vég megkötést;
 - inkrementális a meglévő internethez;
 - nem egyedi folyamatokkal foglalkozik, hanem ún. **szolgáltatási osztályokkal** (CoS – **Classes of Service**).

DiffServ útvásztók

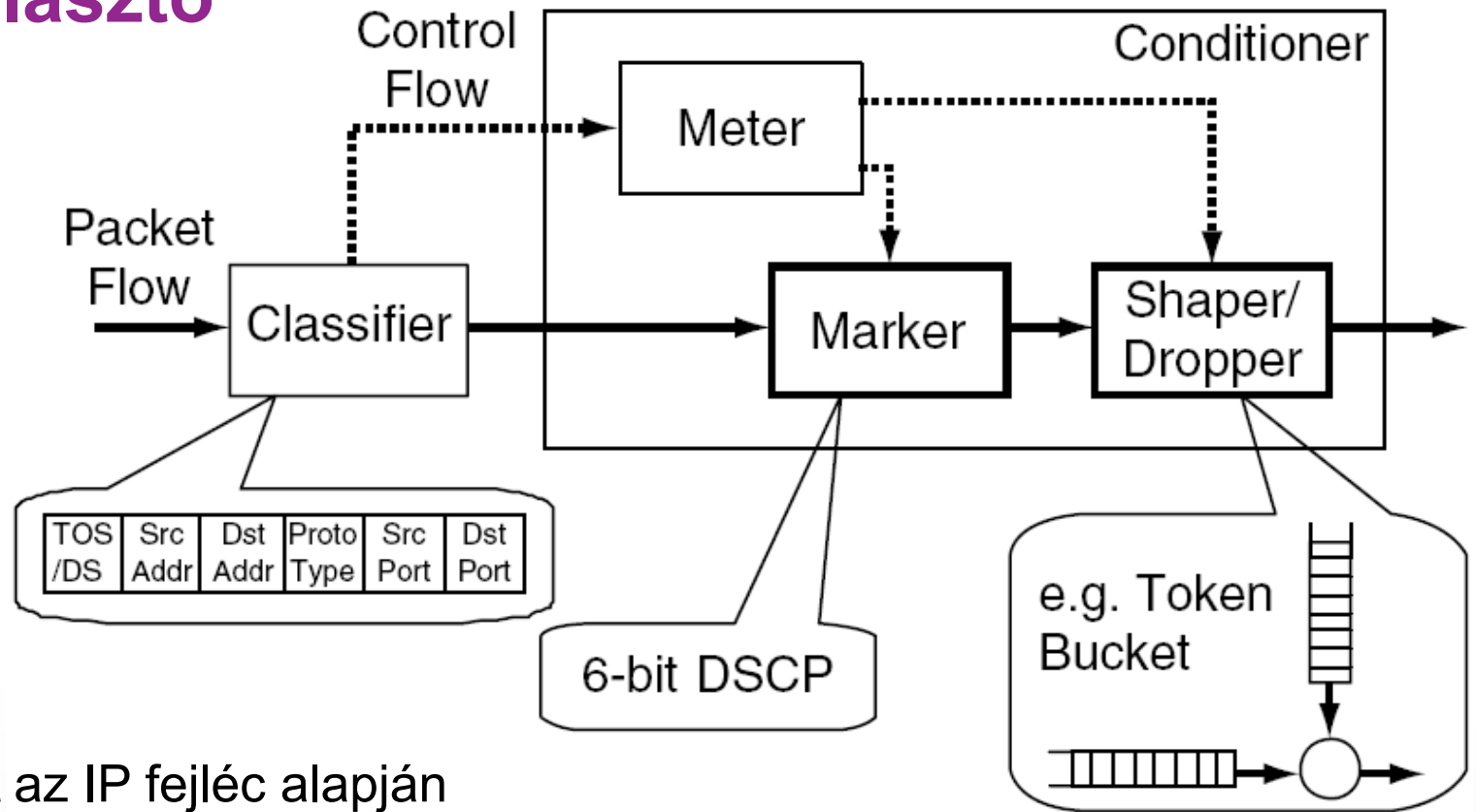
- A **mag (core)** útvásztók...
 - kis számú forgalmi osztályt kezelnek
 - előre definiált útvásztónkénti viselkedést leíró PHB-k (Per Hop Behavior) segítségével.
- A **határ (edge)** útvásztók...
 - A bejövő csomagokat szolgáltatási osztályokba sorolják,
 - felügyeleti (**policing**) és formázási (**shaping**) feladatokat végrehajtva.



Policing & shaping

- **Felügyelet (policing):** annak ellenőrzése, hogy a forgalom-folyam megfelel-e az előzetesen kialakított rátának, és ha nem, a folyam csomagjainak **eldobása**.
- **Formázás (shaping):** annak biztosítása, hogy a forgalom a megállapodásnak megfelelő legyen, pufferelés segítségével (azaz a rövidtávú változékonyság **kisimításával** a folyamon belül).

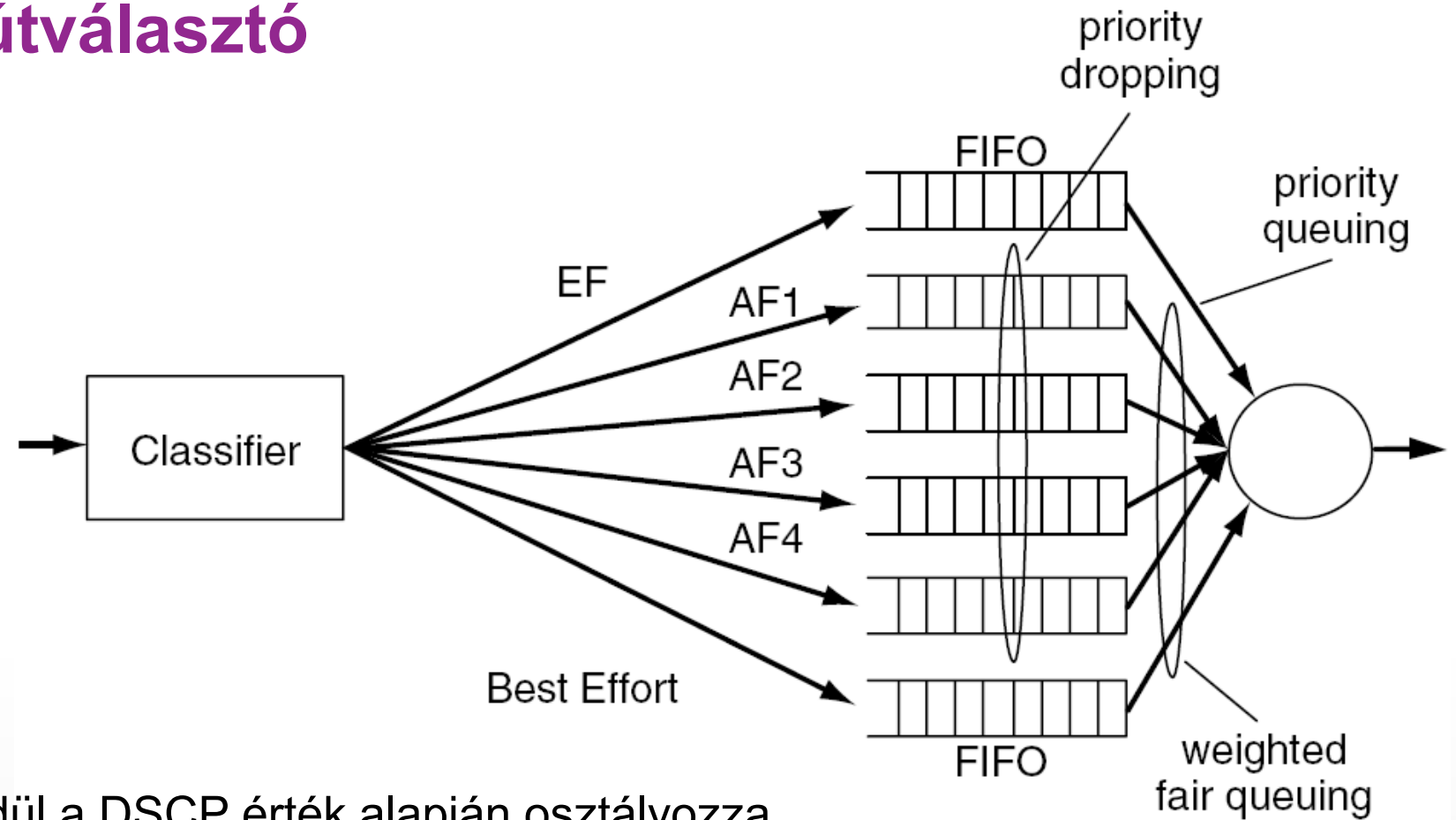
DiffServ határ-útvásztó



A DiffServ határ-útvásztó...

- **osztályozza** a csomagokat az IP fejléc alapján
 - (forrás/cél cím/port, protokoll típus);
- **megjelöli** az IP csomagokat az ún. DSCP (**DiffServ Code Point - kódpont**) segítségével az IP fejrész ToS mezőjében;
- ezután a forgalmat **felügyeli** és **formázza**.

DiffServ mag-útválasztó



- A csomagokat egyedül a DSCP érték alapján osztályozza,
- a csomagokat külön sorokba helyezi a helyi DSCP-PHB leképezésnek megfelelően.

(Pl., EF csomagokat kiemelt prioritással kezeli, az öt további AF sor pedig osztozik a maradék sávszélességen, a súlyozott round-robin (weighted round-robin) ütemezéssel.)

DiffServ szabványosított Per-Hop viselkedés

- Három PHB szabványosított:
 - Class Selector (**osztályválasztás**) PHB
 - Expedited Forwarding (**gyorsított továbbítás**) PHB (EF)
 - Assured Forwarding (**biztosított továbbítás**) PHB (AF)
 - (best-effort PHB – az alap IP viselkedés)
- **Class Selector PHB**
 - Visszafelé kompatibilis a IP fejlécben szereplő ToS mező definíciójával;
 - *„Az útválasztóknak biztosítaniuk kell az azonos vagy magasabb valószínűségű, időzített továbbítást a csomagok számára az alacsonyabb osztályválasztó értékkel rendelkezőkhöz képest.”*
- **Expedited Forwarding PHB**
 - egyszerű magas prioritású továbbítás;
 - *„Az útválasztóknak legalább olyan gyorsan kell kiszolgálniuk a csomagokat, mint ahogyan az EF PHB csomagok érkeznek.”*

DiffServ szabványosított PHB-k (folyt.)

- **Assured Forwarding PHB**

- valójában a PHB-k *egész csoportja(!)*;
- a *csoportban* lehet négy PHB osztály, osztályonként három dobás elsőbbségi szint (**drop precedence level**) definiált;
- az osztály egy adott szolgáltatási osztályt definiál a határon;
- a dobás elsőbbségi szint (drop precedence) határozza meg a csomagok eldobási sorrendjét.

- Példa: a csomagokat megjelölhetjük **zöld**, **sárga** vagy **piros** „színnel” (=szintek). A **piros** csomagok nagyobb valószínűséggel kerülnek eldobásra a **sárgákhoz** képest, stb.

DiffServ átviteli szolgáltatások

Kérdés:

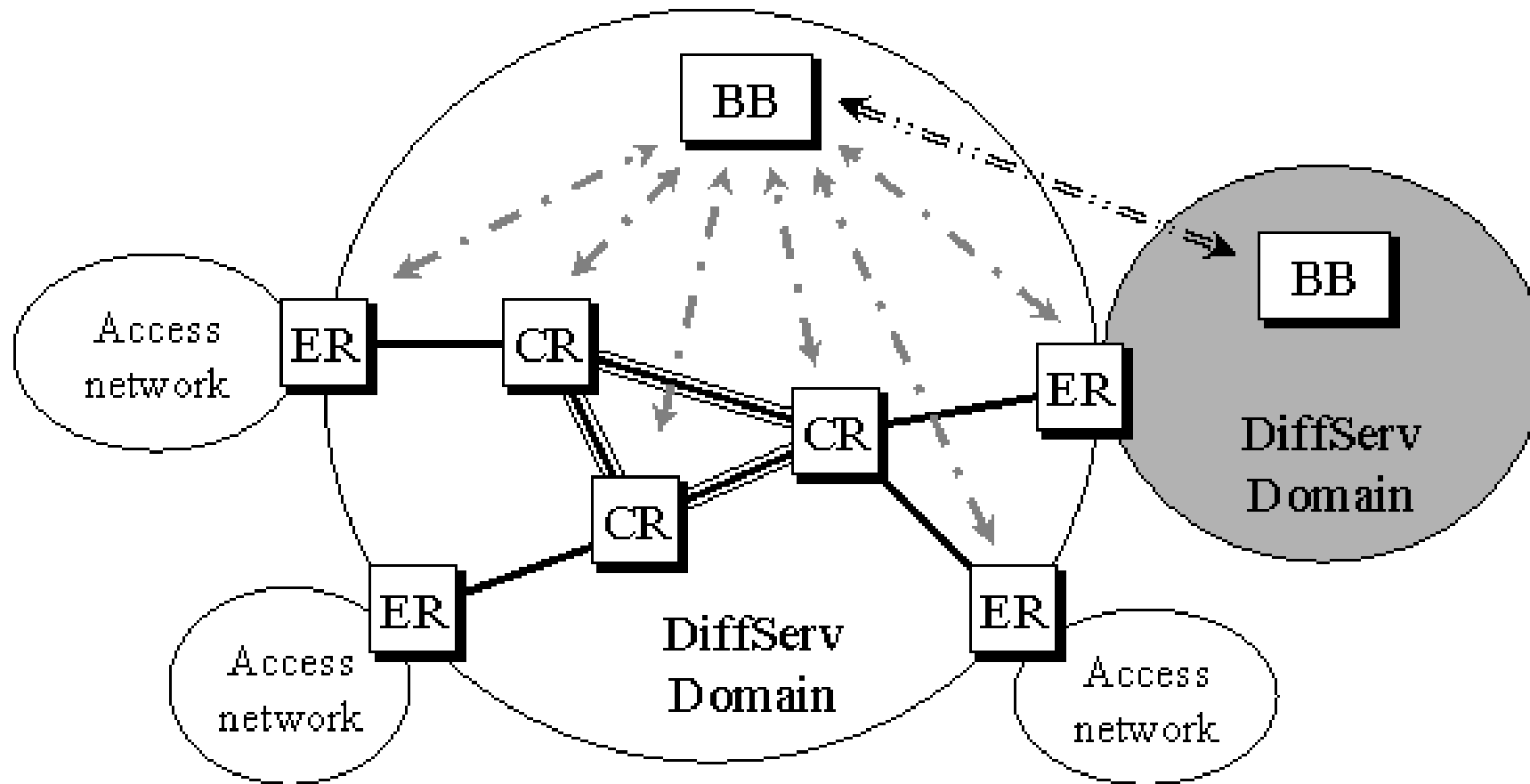
*De hogyan definiálható
vég-vég szolgáltatásminőség
lokális PHB-k segítségével?*

Válasz:

*Garantált átviteli szolgáltatás biztosításához
szükség van beengedés-vezérlésre (admission
control function) is.*

DiffServ átviteli szolgáltatások

- A beengedés-vezérlést egy QoS menedzsmnt rendszer végzi (más néven QoS kiszolgáló vagy sávszélesség bróker (**BB - Bandwidth Broker**)).



DiffServ Bandwidth Broker

- A **QoS kiszolgáló (BB)**...
 - **hívásbeengedés-vezérlést (CAC)** végez a forgalmi osztályokon;
 - ismeri az útvonalválasztó táblákat, és módosítani is képes azokat
→ **QoS routing(!)**;
 - ismeri az összes foglalást ami az adott forgalmi osztályt illeti;
 - felelős az útválasztók QoS-sel kapcsolatos konfigurálásáért (pl., forgalom formázók, osztályozók, ütemezők).

DiffServ példa szolgáltatás

- **Premium Service** (prémium szolgáltatás)
 - sávszélesség garancia, korlátos késleltetés, meghatározott dzsitter, csomagvesztés nélkül;
 - *EF PHB segítségével*;
 - *a folyamatok szigorú felügyelete* (azaz a felügyelő eldobja az összes nem-konform csomagot);
 - *beengedés-vezérlés szükséges* (ellenőrizendő, hogy van-e még elegendő szabad kapacitás az új igény befogadására);
 - *a hálózat globális ismeretét követeli meg* (útválasztó táblák, hálózati erőforrások, aktuális foglaltságok).

DiffServ példa szolgáltatás (folyt.)

- **Assured Service** (biztosított szolgáltatás)
 - biztosított sávszélesség garancia, közel nulla csomagvesztés a konform csomagokra;
 - AF PHB segítségével;
 - kétszínű jelöléssel a határ útválasztónál:
 - a megállapodás szerinti rátával érkező csomagokat **zöldre**, a ráta feletti csomagokat **sárgára** festi.
 - hívásbeengedésnél a „beígért” sebességet veti össze a rendelkezésre álló szabad sávszélességgel;
 - ha egy linken torlódás lép fel, a **sárga** csomagokat dobja el.

DiffServ példa szolgáltatás (folyt.)

- **Better than Best-Effort Service** (legjobb szándéknál jobb)
 - tipikus relatív garantált szolgáltatás, azaz nincs kvantitatív garancia;
 - a legkönnyebben megvalósítható Class Selector PHB alkalmazásával;
 - a forgalom magasabb prioritású mint a best effort.

Ellenőrző kérdés: *Ez mire jó?!?*

- Megjegyzés: további szolgáltatások is lehetségesek a komponensek tetszőleges kombinálásával.

DiffServ - összefoglaló

- Alap DiffServ...
 - nagyon jól skálázódik;
 - nem nyújt lehetőséget bizonyos típusú QoS biztosításához;
 - legtöbbször QoS menedzsment rendszerrel együtt kell megvalósítani
 - ...ez a megoldás azonban jelentősen rosszabbul skálázódik(!),
 - nem feltétlenül megvalósítható maghálózatokban.
- **Általánosságban, a DiffServ gerinchálózatokban hasznos, ahol egyedi alkalmazás folyamatok helyett forgalom aggregátumoknak biztosít garanciát.**
- *(Megjegyzés: a DiffServ nyitott definíciója túl sok és túl flexibilis megoldást tesz lehetővé. Épp ezért nehéz egy általánosan elfogadott megoldást létrehozni különböző gyártók hardver és szoftver eszközeivel!)*

MPLS (+QoS)

Tartalom

- Átviteli szolgáltatásminőség az interneten
 - alap IP
 - integrált szolgáltatás (Integrated Services – IntServ)
 - differenciált szolgáltatás (Differentiated Services – DiffServ)
 - **Multi-protokoll címkekapcsolás (Multi-Protocol Label Switching – MPLS)**

Multi-Protocol Label Switching (MPLS)

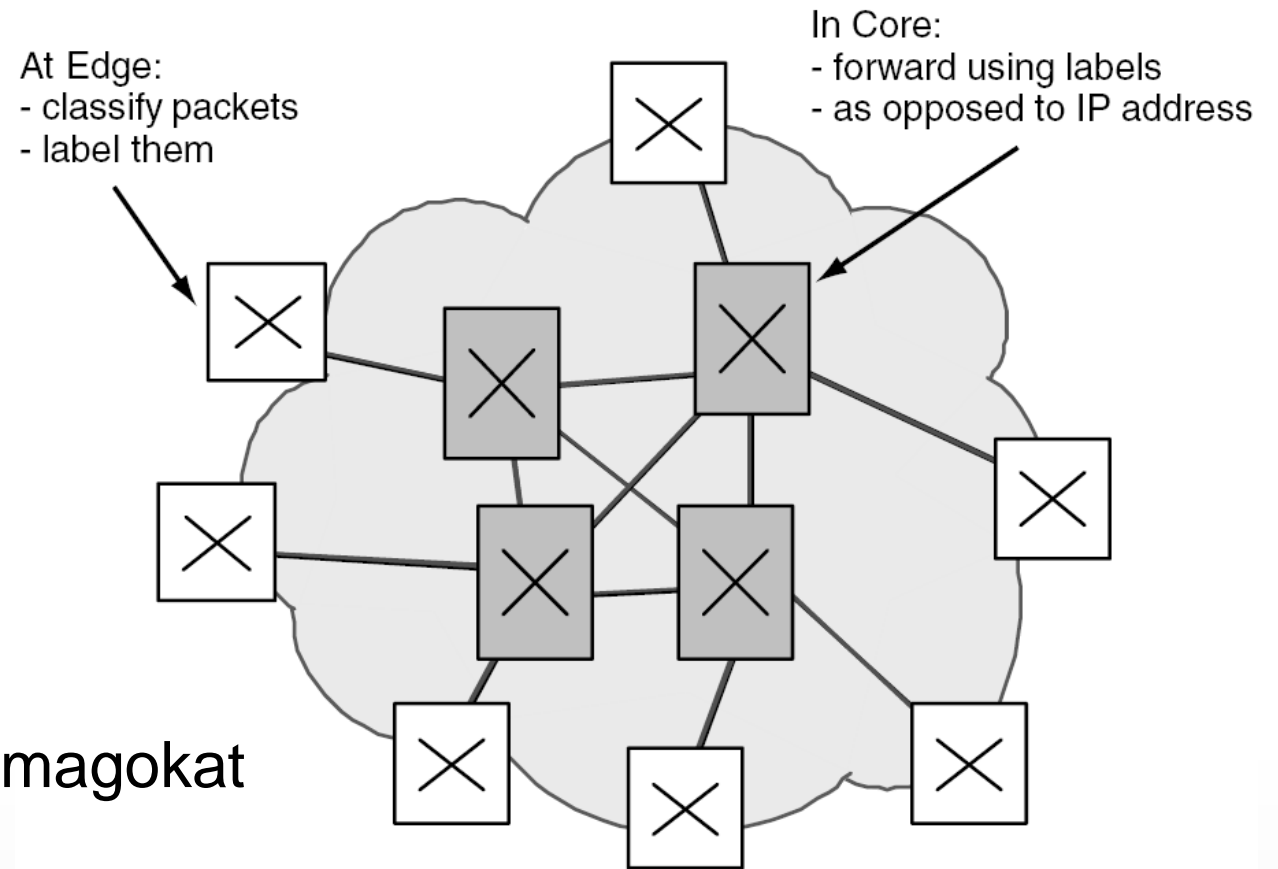
- Az MPLS nem kimondottan egy QoS technológia, de támogathatja a QoS biztosítását az IP routing kiterjesztéseként.

- MPLS alkalmazásakor...
 - csomagokat **megcímkézzük**, amikor belépnek a hálózatba, amelyek ezek után
 - egy fix LSP-t (**Label Switched Path** - **címkekapcsolt útvonal**) követnek.

Multi-Protocol Label Switching (MPLS)

- Az MPLS főbb sajátosságai: Az IP csomagok egy rövid címke alapján kerülnek továbbításra (címkekapcsolás - **label switching**) a leghosszabb prefix cím keresés (IP routing) helyett, így
 - a továbbítás *egyszerű és gyors*;
 - a címkének csak **link-szintű** jelentősége van,
 - a címke minden csomópontban **megváltoztatható** az útvonalon.
- LSP-khez a csomagok **flow szinten** rendelhetők.
 - Azaz: **Egy forrás-cél páros között több LSP is lehetséges, eltérő QoS karakterisztikával.**

MPLS architektúra



- A határ csomópontoknál a csomagokat *osztályozzák és felcímkézik*.
- A csomagok a helyi LSR (**címkekapcsoló útválasztó - Label Switching Router**) konfiguráció alapján továbbítódnak a következő csomóponthoz.
- A maghálózatban csak a címke használható az útvonal megtalálásához.

MPLS és QoS

- MPLS használható QoS biztosítására, ha egyéb QoS technológiákkal kombináljuk.
 - PI.1, MPLS leképezi az LSP-eket ATM virtuális kapcsolataira, így IP QoS-t biztosítva LSP-nként az ATM QoS-re alapozva.
 - PI.2, MPLS + DiffServ: különböző LSP-k használata szolgáltatási osztályonként, így lehetővé téve az osztályok egymástól független útválasztását.

MPLS összegzés

- Az MPLS önmagában nem egy QoS technológia, de nyújt olyan funkcionalitást amely megkönnyíti a QoS biztosítását. QoS könnyebben biztosítható az útvonalak rögzítésével.
- A hálózatot monitorozva és a forgalom elterelése a torlódásos szakaszokról bizonyos szintű minőséget biztosít önmagában is.
- Másrésztől megkérdőjelezhető, hogy jó döntés-e még egy járulékos réteget hozzáadni az IP alá(?!)
 - menedzsment komplexitását növeli;
 - a hálózati rétegnek is többletköltséget jelent;
 - IP-re támaszkodik (jelzés a címkék terjesztéséhez és útválasztó protokollokhoz).

IntServ, DiffServ, MPLS összefoglalás

- **IntServ** (RSVP) biztos (matematikailag igazolható) QoS határértékeket, de **nem skálázható**.
- **DiffServ** gerinchálózatokban hasznos, ahol egyedi alkalmazás folyamatok helyett forgalom **aggregátumoknak biztosít garanciát**.
- **MPLS** önmagában nem egy QoS technológia, de nyújt olyan funkcionalitást amely **megkönnyíti a QoS biztosítását**.



Áttérés nyílt rendszerekre

API-k

Tartalom

- Átviteli szolgáltatásminőség az interneten
 - alap IP
 - integrált szolgáltatás (Integrated Services – IntServ)
 - differenciált szolgáltatás (Differentiated Services – DiffServ)
 - Multi-protokoll címkekapcsolás (Multi-Protocol Label Switching – MPLS)

- **Áttérés nyílt rendszerekre**
 - OSA – Open Service Access (nyílt szolgáltatás hozzáférés)

Áttérés nyílt szolgáltatásokra

- **Konvergens hálózati trend**
 - RÉGI: **zárt**, magántulajdonban kezelt rendszerek
 - természeténél fogva rugalmatlan, nem egykönnyen programozható
 - Új szolgáltatások bevezetése gyakran jelentős anyagi és idő- ráfordítást igénylő erőfeszítés.



- ÚJ: **nyílt rendszerek** lehetővé teszik bárki számára szolgáltatás nyújtását a felhasználóknak

De milyen „rendszerekről” is van szó?

Áttérés nyílt rendszerekre – „minek?”

- A hálózat megnyitásának célja:
Lehetővé tenni az *alkalmazások* számára a közcélú hálózatok *kommunikációs képességeinek* elérését.

Motivációk:

- A **(hálózati) szolgáltatók** érdekeltek a forgalom (így a **bevétel!**) növelésében, attraktív szolgáltatások nyújtása által.
- A **stratégia**: **független szoftverfejlesztők** bátorítása olyan alkalmazások kifejlesztésére, amelyekkel a (hálózati) kommunikációs szolgáltatások fejleszthetők.
- Az **információs szolgáltatások fejlesztői** keresik a hálózati kommunikációs lehetőségeket saját IT alkalmazásaik kiterjesztéséhez.
 - (pl., banki tranzakció értesítők)

Áttérés nyílt rendszerekre – „hogyan?”

- **Mire van szükség?**

- Az alkalmazások (azaz felhasználók) *elválasztása* a generikus funkcionalitástól.
- *Biztonságos, megbízható és számlázható* felhasználói kontroll biztosítása a *hálózati funkciók* kezeléséhez, amelyeket különböző hálózati elemek (pl. kapcsolók, adatbázisok) biztosítanak.
- Fontos: *Míg az internet természetesen bátorítja és elősegíti (pl. intelligens végberendezések alap átviteli szolgáltatással) a nyílt hálózati felhasználást, addig a távközlési szolgáltatók hálózatainak megnyitásánál figyelembe kell venni a hálózat **szolgáltatási képességeit**.*
- Ez megtehető egy **nyílt interfész** biztosításával, azaz az **API** (alkalmazás programozói interfész) koncepció használatával.

Nyílt szolgáltatás hozzáférés (OSA)

- **OSA** (**Open Service Access** – **nyílt szolgáltatás hozzáférés**).
- 3GPP szabvány definíciója szerint: „*az OSA egy koncepció az új szolgáltatások bevezetésének gyártófüggetlen módjára*”.
- A 3GPP szabvány definiál egy **OSA architektúrát** amely lehetővé teszi a távközlési szolgáltató (telco) hálózatán *kívüli* alkalmazás fejlesztők számára, hogy...
 - *kihasználják a hálózati funkcionalitást, és*
 - *információt kapjanak a hálózatból***...szabványos interfészeken (API) keresztül.**

Az API-k koncepciója

- **API** (**Application Programming Interface** – alkalmazás programozói interfész)
 - Def: **Szolgáltatás képes API-k** alatt a hálózat által nyújtott funkciók halmazát értjük, amely lehetővé teszi programozók számára a felhasználóknak szolgáltatást nyújtó programok írását.
- A **hálózati operátorok** által hozzáférhetővé tett **nyílt szabványos API-k** konvergens hálózatokban:
 - Lehetővé teszi **független szolgáltatás fejlesztők** számára a hálózat által támogatott alkalmazások fejlesztését **anélkül, hogy ezzel veszélyeztetnék a hálózatot vagy annak szolgáltatásait.**

Rétegek, protokollok, API-k

- A **távközlő hálózatok** világában a hangsúly a (hálózati) *rétegek specifikálásán* van.
 - A protokollok specifikálása és szabványosítása teszi lehetővé a hálózatok együttműködését.
 - PI. TCP/IP stack: (HTTP + TCP + IP protokollok)
 - PI. SS7 stack: (ISUP, TUP, MTP-x protokollok)
- A **szoftverek** világában a hangsúly a *rétegek közötti interfészek*en (API – Application Programming Interface) van.
 - *API-k szabványosításával platformfüggetlen alkalmazások* hozhatók létre.
 - PI. POSIX (Unix és Linux rendszerek)
 - PI. WIN32 (Microsoft Windows 95/98/2000, stb.)

TCP	Transmission Control Protocol	HTTP	HyperText Transfer/Transport Protocol
SS7	Signalling System Number 7		ISUP ISDN User Part
TUP	Telephony User Part	MTP	Message Transfer Part
POSIX	Portable Operating System Interface		

Rétegek, protokollok, API-k (2)

- Az **API koncepció** használatával a **távközlésben** lehetőség nyílik **hordozható, technológia- és protokollfüggetlen alkalmazások fejlesztésére**.
 - Pl. konferenciahívás esetén egyes résztvevők PSTN-en, mások mobilon vagy hálózati terminálokön kapcsolódnak.
- Megjegyzés: *Kb. 3 millió professzionális szoftverfejlesztő áll szemben kb. 10 ezer specializálódott távközlési szolgáltatás fejlesztővel!*

API-k jellemzői

- A **nyílt szolgáltatás hozzáférés API-k** hét **tulajdonsága**:
 - **Nyíltság:**
 - Széles körben elfogadott szabvány definiálja.
 - **Biztonság:**
 - A szolgáltatás interfészek hozzáférése csak hitelesített és felhatalmazott fél részére megengedett.
 - **Integritás (sérthetetlenség):**
 - A hálózat funkcionalitása nem veszélyeztethető az alkalmazás esetleges túlzott hívásai következtében.
 - **Rugalmasság:**
 - Az interfész lehetővé teszi változatos alkalmazások támogatását, és az interfész implementációja biztosítja különféle hordozó hálózatok használatát.
 - ...**(folyt.)**

API-k jellemzői (folyt.)

- A **nyílt szolgáltatás hozzáférés API-k** hét **tulajdonsága**:
 - ...
 - **Absztrakció:**
 - A mögöttes hálózat és annak heterogenitása rejtve kell maradjon az alkalmazás programozó előtt. (Azaz a programozótól nem elvárható, hogy megértse az adott hálózati technológiát és jelzési protokolljait.)
 - **Technológia semlegesség:**
 - Az API definíciója nem kötött egyetlen adott nyelvhez vagy implementációs technológiához.
 - **Szolgáltatás felderítés:**
 - Egy alkalmazás képes felderíteni a rendelkezésre álló szolgáltatásokat egy kiszolgáló node-ban.

Az OSA/Parlay

- A bemutatott nyílt API-k három szervezettől eredeztethetők:
 - **A Parlay csoport**
 - a telco-k és gyártók egy konzorciuma, nyílt API-k kifejlesztésének egyedüli céljával;
 - **A 3GPP** (Third-generation partnership project)
 - a harmadik generációs mobil kommunikációs rendszerek minden aspektusával foglalkozik;
 - **ETSI SPAN** (Services and Protocols for Advanced Networks)
 - az ETSI „Szolgáltatások és protokollok fejlett hálózatok számára” „csoportja” OSA és API szabványokat publikál.
- A következőkben tárgyalandó API-k legtöbbször „Parlay/OSA” vagy „OSA/Parlay” néven hivatkoztak.

A Parlay csoport

„Welcome to the Parlay Group

*The Parlay Group aims to intimately **link IT applications** with the **capabilities of the telecommunications world** by specifying and promoting **application programming interfaces (APIs)** that are secure, easy to use, rich in functionality, and based on open standards. Parlay integrates telecom network capabilities with IT applications via a secure, measured, and billable interface. Enabled by Parlay's network-independent APIs, applications are generating new revenue streams for network operators, application service providers (ASPs), and independent software vendors (ISVs).”*

The Parlay API-k története

- **Cél:** nyílt, technológia független API-k létrehozása a tradicionális hálózati operátor körén *kívüli* alkalmazás fejlesztők számára.
- 1. fázis
 - 1998: A Parlay csoport létrehozása 5 vállalat által: British Telecom, Microsoft, Nortel Technologies, Siemens and Ulticom.
 - 1999. szept: 1.2 verzió – A következő API-k létrehozása: Framework, Call Control, Messaging, és User Interaction
- 2. fázis
 - Hat új tag: AT&T, Cegetel, Cisco, Ericsson, IMB, Lucent Technologies
 - 2000. jan: 2.1 verzió – csomagkapcsolt és vezeték nélküli hálózatok támogatása

The Parlay API-k története

ETSI	European Telecommunications Standards Institute
3GPP	Third Generation Partnership Project
JAIN	Java API for Integrated Networks
OSA	Open Service Access

- 2. fázis (folyt.):
 - 1999: ETSI és 3GPP közös munkába kezdett az OSA (Open System Architecture) kifejlesztésére a Parlay specifikációk felhasználásával.
 - ETSI elkezdte a Parlay API-k szabványosítását.
- 3. fázis:
 - 2000 jún: A Csoport kibővült, és hangsúlyt fektetett a Parlay „reklámozására”.
 - Együttműködés az ETSI-vel és 3GPP-vel, input a JAIN résztvevőitől → „Joint API Group” megalakítása.
 - 2001. júl: *Parlay 3* specifikáció elkészült.
- 4. fázis:
 - 2002. nov: Policy Management, Presence and Availability Management API-k hozzáadása
 - Egy új middleware technológia, a **Web szolgáltatások** támogatása

The Parlay API-k története

- 4. fázis (folyt.)
 - **Web szolgáltatások:**
 - A Parlay API-k WSDL (Web Service Description Language) publikálása, lehetővé téve a Parlay Gateway és az alkalmazás interakcióját HTTP és SOAP (Simple Object Access Protocol) használatával.
 - új API garnitúra **Parlay X** néven
 - 2004. jún: *Parlay 4.1* és *Parlay X Web szolgáltatás specifikáció* kiadása
- fázis... legújabb
 - 2004. szept: *Parlay X 2.0*
 - 2005. ápr: *Parlay 5.0*
 - 2007. jan: *Parlay 5.1* kiadása
 - 2007. nov: Draft *Parlay 6.0 Spec* kiadása (ETSI ES 204 915)
 - 2007. szept: *Parlay X 3.0*
 - 2007-ben befejezte működését.

Parlay csoport – teljes tagok



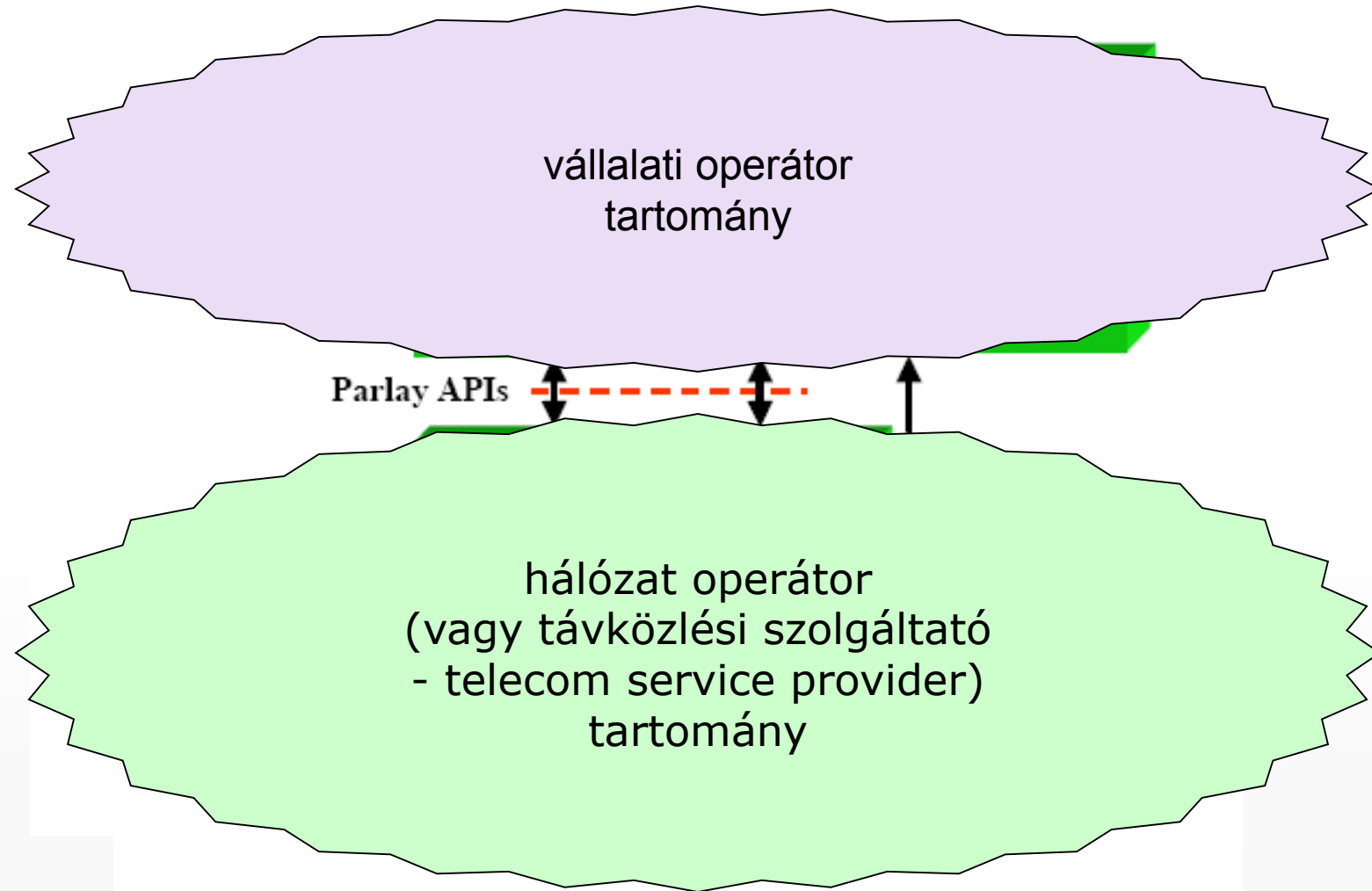
Parlay csoport – kapcsolattartó tagok



Parlay/OSA architektúra

- A **Parlay/OSA architektúra** két nagy tartományt definiál nyílt API-val a tartományhatáron:
- A **hálózat operátor tartomány**...
 - a *mögöttes hálózat* és a *funkcionalitás*, amely **implementálja a nyílt API-t**;
 - más néven a *telecom szolgáltatói tartomány* (telecom service provider domain).
- A **vállalati operátor tartomány**...
 - egy vagy több *alkalmazást* tartalmaz (kliens alkalmazások) amely a **nyílt API-t használja**.
 - A **vállalati (Enterprise) operátor**...
 - egy *független szolgáltató* amely *üzleti kapcsolatban* áll a hálózat operátorral.;
 - *feliratkozik* a hálózati operátor által nyújtott *hálózati szintű szolgáltatásokra*.

Parlay/OSA architektúra

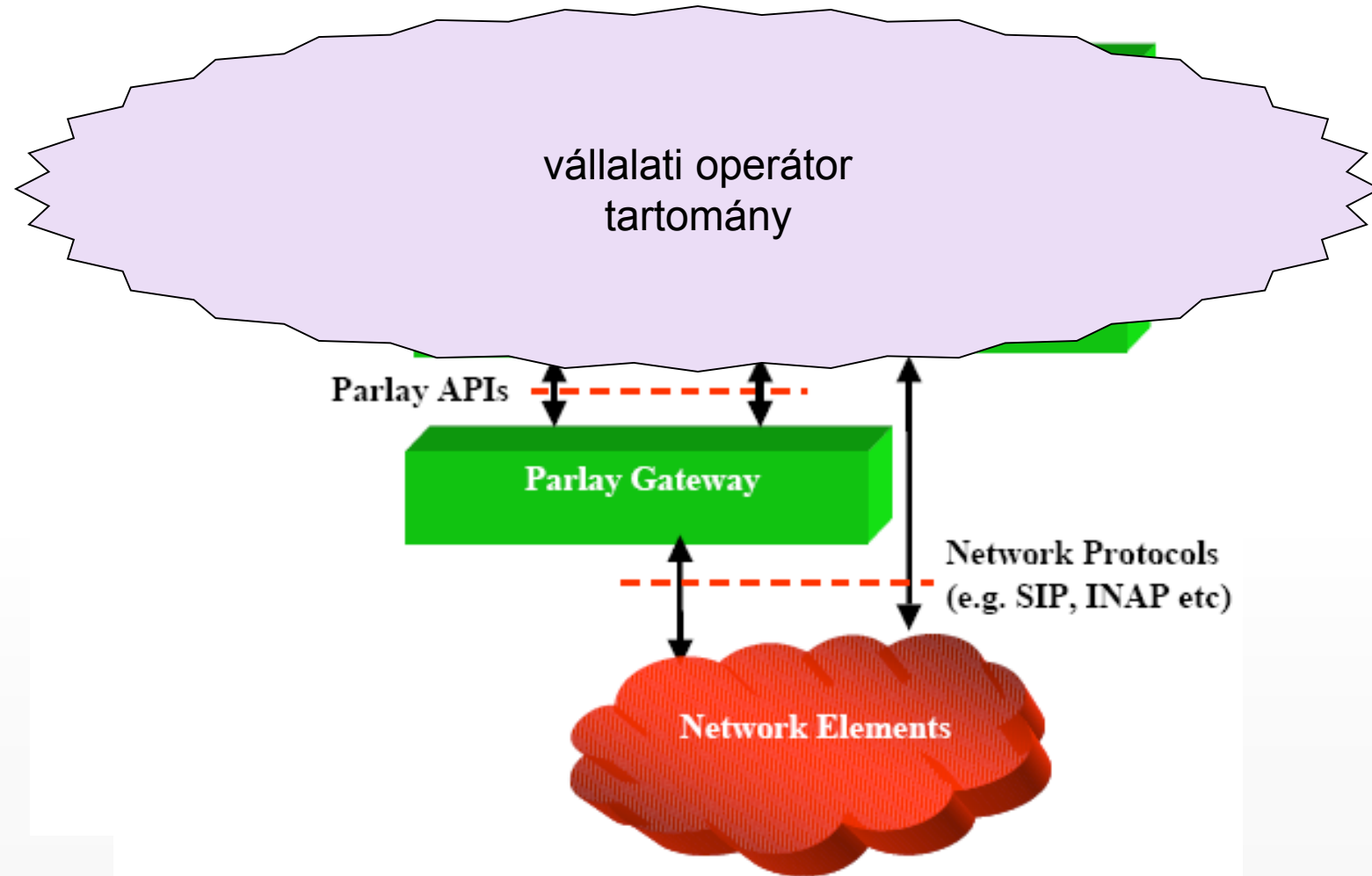


forrás: parlay.org

Parlay/OSA Framework

- A **hálózati operátorok** és szolgáltatók részéről kulcs-követelmény, hogy a hálózat megnyitásával a **hálózati infrastruktúra ne legyen kiszolgáltatva** jogosulatlan hozzáféréseknek vagy egyéb fenyegetéseknek.
- Ezt a funkciót a **Framework API** látja el, amely egy Parlay/OSA **Gateway-en** (átjárón) implementált.

Parlay/OSA architektúra

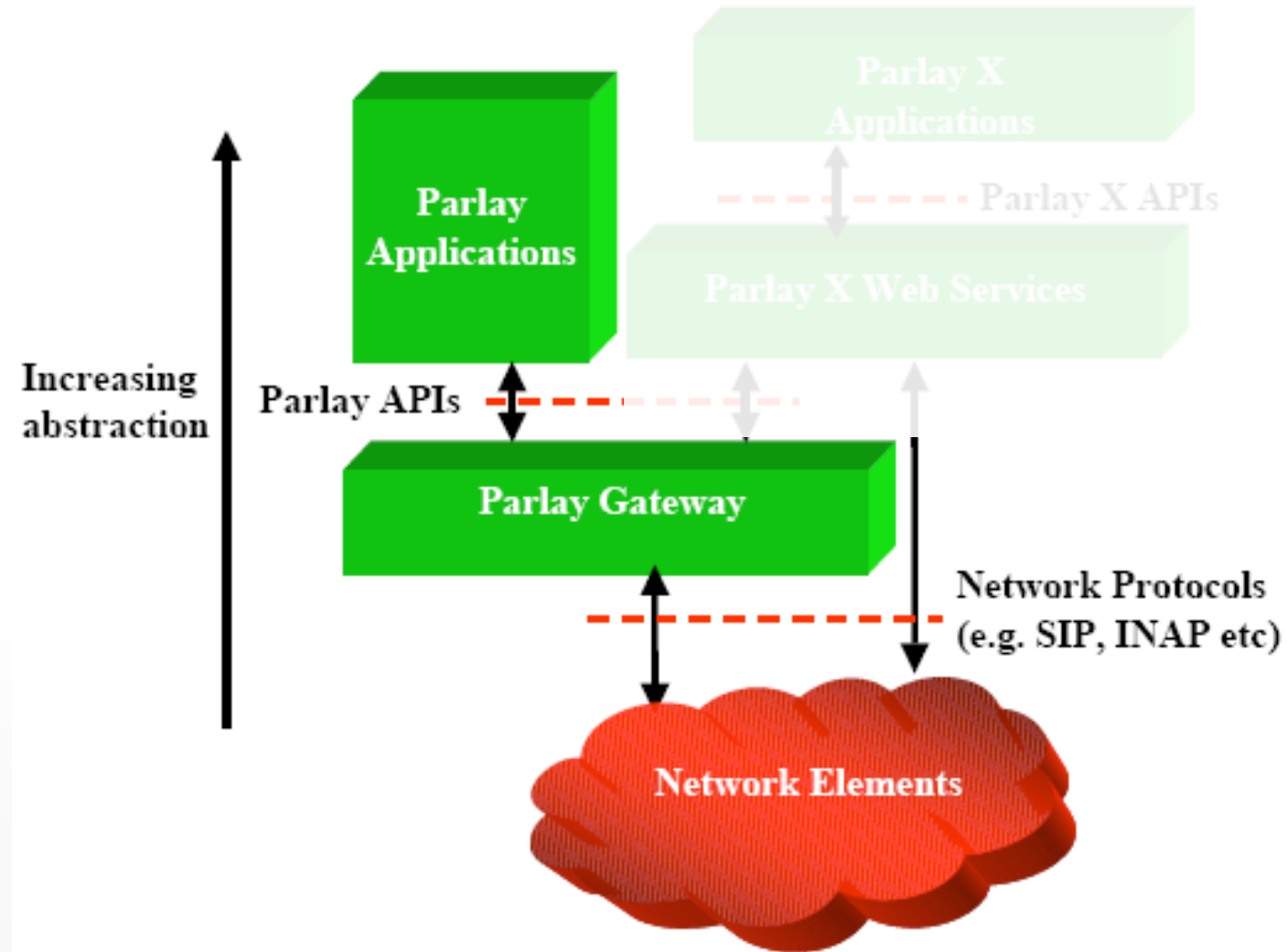


forrás: parlay.org

Parlay/OSA Application Server

- Az *alkalmazás* egy **alkalmazás kiszolgálón** (**Application Server**) fut, és TCP/IP hálózaton kapcsolódik a *gateway*-hez.
- Vállalati alkalmazás kiszolgálók elhelyezési lehetőségek:
 - A kiszolgáló egy *vállalati adatközpontban van*, valamilyen üzleti alkalmazást támogat, és a távközlési szolgáltató hálózatát használja a Parlay API-n keresztül.
 - Egy vállalaton kívüli *harmadik fél* (pl. alkalmazás szolgáltató) ad otthont a kiszolgálónak a vállalat megbízásából.
- Sok (hálózat) operátor telepít alkalmazás kiszolgálókat a saját hálózatán belül.
 - Ebben az esetben u.azokat a Parlay/OSA API-kat használja, mint bármely harmadik fél.

Parlay/OSA architektúra



SIP Session Initiation Protocol
INAP Intelligent Network Application Protocol

forrás: parlay.org

Parlay API-k

- A **Parlay API-k** két fő részre oszthatók:
 - „Framework” (**keretrendszer**)
 - „Services” (**szolgáltatások**)
- A **Framework** rész...
 - Biztosítja a *lehetőséget* a hálózati operátor és az alkalmazást nyújtó fél közötti *egyeztetésre*.
 - Biztosítja a **kapcsolatfelvételi pontot** az alkalmazás számára a hálózat által nyújtott szolgáltatások felderítéséhez.
 - Minden szükséges *biztonsági óvintézkedést* figyelembe vesz.
- A **Services** rész...
 - **Nagyszámú interfészt biztosít** a hálózat képességeinek és lehetőségeinek hozzáféréséhez és kihasználásához.
(Pl. mobilitás menedzsment, hívásvezérlés, jelenlét és elérhetőség, ...)

Parlay API-k (2)

- Parlay v4.0, 3GPP OSA rel.5, ETSI ES 202 915 API-k:
 - General
 - Common Data
 - **Framework**
 - Call Control
 - User Interaction
 - Mobility
 - Terminal capabilities
 - Presence and Availability Management
 - Policy management
 - Generic Messaging
 - Content Based Charging
 - Data Session Control
 - Account Management
 - Connectivity Management

Parlay API-k

▪ Framework

- az alap mechanizmusokat biztosítja az alkalmazások számára a hálózat által felajánlott szolgáltatások igénybevételéhez.
- Mielőtt az alkalmazás használhatná az API-kat, hitelesítésre van szükség az alkalmazás és a Framework között.
- **Framework interfészek:**
 - Megbízhatóság és biztonság menedzsment – Trust and security management
 - Regisztráció – Registration
 - Szolgáltatás felderítés – Service discovery
 - Szolgáltatás életrajz menedzser – Service lifecycle manager
 - Integritás menedzser – Integrity management
 - Esemény értesítés – Event notification
 - Szolgáltatási megállapodás menedzsment – Service agreement management
 - Szolgáltatás előfizetés – Service subscription

Parlay API-k (3)

- **Call Control** (hívásvezérlés)

- Lehetővé teszi az alkalmazásoknak *hívások felépítését* a hálózatban, alapvető hívásvezérlési és többrésztvevős hívásvezérlési funkciókat is biztosítva.

- **User Interaction** (felhasználói interakció)

- Definiálja, hogyan juthatnak az alkalmazások információhoz a *végfelhasználótól*, közleményeket kezel, rövid üzeneteket küld, stb.

- **Mobility** (mobilitás)

- Lehetővé teszi az alkalmazások számára, hogy meghatározzák egy terminál (telefon) *elhelyezkedését*, és lehetővé teszi számukra, hogy értesítést kérjenek amikor a terminál helyet változtat.

- **Terminal** (végberendezés)

- Lehetővé teszi a felhasználói végberendezés (telefon) *képességeinek felderítését*.

Parlay API-k (4)

- **Availability Management and Presence**

(**elérhetőség és jelenlét**)

- Lehetővé teszi az alkalmazások számára, hogy információt kapjanak (ill. információval szolgáljanak) a felhasználók *elérhetőségéről* és *jelenlétéről*. (pl. „Otthon vagyok.”, „Az irodámban ülök.”)

- **Policy Management** (**felügyelet**)

- *Felügyeleti szabályok* létrehozásához, vagy *felügyeleti események* regisztrálásához

- **Messaging** (**üzenetküldés**)

- Hozzáférést engedélyez postafiókokhoz, *üzeneteket küld és fogad*, alkalmazások interakcióját kezeli az *üzenetkezelő rendszerekkel* (pl, hang, fax, e-mail, ...).

Parlay API-k (5)

- **Content Based Charging** (tartalomfüggő számlázás)
 - Vezérli az alkalmazások kéréseit a szolgáltatás *kifizetéséről*.
- **Data Session Control** (adathívás vezérlés)
 - Menedzseli az alkalmazás terminálról indított *adatforgalmát*. (Tipikusan GPRS és 2.5G alkalmazások esetén.)
- **Account Manager** (könyvelés)
 - Lehetőséget ad az alkalmazásoknak hozzáférési információk és múltbéli *tranzakciók listázására*.
- **Connectivity Management** (kapcsolat)
 - Lehetővé teszi az alkalmazások számára a végpontok közötti *szolgáltatminőség* (QoS) befolyásolását ill. vezérlését.

Hálózatok megnyitása

- **Az *API koncepció* használatával a távközlésben lehetőség nyílik *hordozható, technológia- és protokollfüggetlen alkalmazások fejlesztésére.***
- Megjegyzés: *Kb. 3 millió professzionális szoftverfejlesztő áll szemben kb. 10 ezer specializálódott távközlési szolgáltatás fejlesztővel!*

