

**PA159 – Protokoly kvality služeb**  
**20. 11. 2009**

# Protokol RSVP

- Resource reSerVation Protocol
  - Protokol, definující, jak jsou rezervovány zdroje na aktivních prvcích mezi odesílatelem a příjemcem
- Popsán primárně v RFC 2205
  - Řada návazných RFC (2206–2210, 2379, 2382, 2490, 2745, 2746, 2747, 2961)
- Přijat IETF jako Internetový standard
- Další informace např. na  
<http://www.isi.edu/div7/rsvp/rsvp.html>

# RSVP – základní vlastnosti

- Podpora unicastu i multicastu
  - Navržen primárně pro multicast
- Jednosměrnost: rezervace pro každý směr zvlášť
- Rezervaci iniciuje přijímající strana
  - Obě vlastnosti vyplývají z charakteru multicastu
- Soft stav
  - Brání nekonečnému blokování rezervovaných prostředků
- Různé styly rezervací (agregace)

# RSVP – Inicializace

- Zahájena příjemcem dat
  - „Proti“ toku dat
  - Postupně až k vysílajícímu
    - \* Možnost agregace požadavků od různých příjemců
    - \* Kombinace s konkrétními protokoly přenosu (viz. „mixers“ v RTP, prezentováno později)
- Nezbytný princip pro multicastové skupiny
  - Vysílající nezná přijímající

# RSVP – základní funkcionalita

- Toky
- Relace
- Filtrace

# RSVP – Toky

- Základní rezervační jednotka
- Specifikace toku
  - Třída služby (včetně informace, zda směrovač smí spojovat požadavky)
  - Rspec: požadovaná kvalita služby
  - Tspec: charakteristika toku
- Oba poslední identifikátory jsou pro RSVP protokol transparentní
  - RSVP požadavek přenese
  - Ale sám rezervaci zdrojů neprovede (to udělá aktivní prvek)

# RSVP – Relace

- Zevšeobecnění toku (flow) pro multicastovou skupinu
- Základní atributy:
  - Cílová IP adresa
  - Identifikátor použitého IP protokolu
  - Cílový port

# RSVP – Filtrace

- Konkretizace práce s tokem/relací
- Specifikace
  - Zdrojová adresa
  - Zdrojový port (UDP/TCP)
- Tři typy filtrů
  - Fixní: explicitní specifikace vysílajícího a současně požadavek na oddělenou rezervaci (příjem videa z konkrétního zdroje)
  - Explicitně sdílená: specifikace vysílajícího, avšak rezervace možno sdílet (audio, pokud předpokládáme vždy jediný zdroj v konkrétním okamžiku)
  - Wildcard: není specifikován vysílající a rezervace může být sdílena (audio)



# RSVP – Problém nedostatečných zdrojů

- RSVP sdružuje požadavky
  - Linky u vysílajícího mohou být přetíženy
- killer-reservation problem:
  - Jedna velká rezervace blokuje ostatní
    - \* I když je rozeznán neúspěch „výše“, ostatní rezervace efektivně blokovány
  - Blockade state – nedovolí sdružení „škodného“ požadavku na rezervaci

# RSVP – shrnutí

- RSVP ustavuje spojení/rezervaci přes každý tok
- Problematicky škálovatelné
  - Aktivní prvky blízko vysílajícího si musí pamatovat příliš mnoho toků
  - Sdílení může částečně pomoci
- Problém oboustranné (všestranné) komunikace
  - Příliš mnoho rezervací
- Další vývoj: odstranění rezervačních požadavků per tok

# Soft State

- Základní princip síťových (webových, gridových, ...) služeb
  - Každá služba je aktivní pouze omezenou dobu
  - Pokud má služba trvat déle, musí být „udržena při životě“ – *keepalive zprávy*
  - Umožňuje korektní reakci na výpadky v distribuovaných systémech
    - \* Nepřijde-li keepalive zpráva, zdroj je uvolněn
- Typicky dynamické rezervace musí být takto realizovány

# Diferencované služby – DiffServ

- Problémy RSVP:
  - Stavová informace v aktivních prvcích
  - Neumožňuje *uspořádání* služeb (tok A má být vždy preferován
    - \* Na druhé straně bez agregace garantuje každému toku jeho požadavky před tokem B)
- Řešení: diferencovaná služba
  - Flexibilní
  - Škálovatelné

# DiffServ – principy

- RFC 2475
- Nedefinuje třídy (classes) služeb
  - Místo toho pouze (predefinované) prioritní třídy s různou (opět predefinovanou) rezervací zdrojů
    - \* Uspořádání (preference), nikoliv absolutní garance
- Poskytuje rámec, tj. funkční komponenty, z nichž je možno flexibilní službu sestavit
  - Hraniční (edge) funkce
  - Funkce jádra (core)

## DiffServ – principy (2)

- Vnitřní (v jádře sítě) směrovače znají několik (málo) tříd a jejich prioritu
- Hraniční směrovače každý příchozí paket zařadí do jedné konkrétní třídy
  - Podle dohody mezi poskytovatelem služby a uživatelem
- Jádro prioritizuje pakety podle návěstí toku
- Uvnitř toků opět pouze best effort

# DiffServ – hraniční funkce

- Klasifikace paketů
  - Nastaví DS (Differentiated Service) pole IP paketu
  - Rozdělí pakety do tříd
- Úprava toku (traffic conditioning)
  - Pakety zahodí, zdrží, pošle dál
    - \* Podle nastavené dohody (SLA, Service Level Agreement) a skutečného chování uživatele

# DiffServ – funkce jádra

- Přeposílání (forwarding)
- per-hop behavior (PBH) spojen s každou třídou
  - Závisí pouze na označení paketu (poli DS)
  - Nepotřebuje stavovou informaci o tocích ve směrovačích
  - Pojem *behavior aggregate*



# DiffServ – Klasifikace

- DS pole nahrazuje původní TOS pole (Type of Service)
- 8 bitů, dvě části
  - DSCP (Differentiated Service Code Point): 6 bitů, definuje PHB
  - CU (Currently Unused): 2 bity, jméno říká vše
- Dva kroky
  - Klasifikace: např. výchozí/cílová IP adresa, číslo portu, ...
    - \* Do které třídy patří
  - Označení (marking): v závislosti na klasifikaci
    - \* Nastavení příznaku DSCP
- Pravidla klasifikace nedefinována v rámci DiffServ skupiny

# DiffServ – úprava toku

- Profil toku (traffic profile): dohodnut s uživatelem
- Měřicí funkce: kontroluje shodu
- Shaper/Dropper: podle výsledku měření buď zdrží nebo zahodí pakety, které neodpovídají dohodnutému profilu

# DiffServ – jádro

- Per-hop behavior

- Každý směrovač odpovídá za přenos na další směrovač v cestě
- Rozhodování striktně podle označení paketů
- Definice:

A description of the externally observable forwarding behavior of a DiffServ node applied to a particular DiffServ behavior aggregate.

# DiffServ – PHB

- Expedited forwarding (EF), RFC 2598
  - Rychlost přeposílání paketů pro danou třídu musí být alespoň dohodnuté úrovně
  - Implikuje *izolaci* tříd
  - Jednoduchá abstrakce spojení s minimální garantovanou kapacitou
- Assured forwarding (AF), RFC 2597
  - Složitější – viz dále

# DiffServ – Assured Forwarding

- 4 třídy, každá s konkrétní minimální garantovanou kapacitou a velikostí bufferů
- Každá třída rozdělena do tří kategorií podle „drop preference“
  - Použity při přetížení: prioritně zahazovány pakety s nejvyšší drop preferencí
- AF (i EF) musí být vždy doplněno „policy“, tedy konkrétní dohodou s uživatelem a současně zajištěním, že celková kapacita sítě umožňuje garance dodržet.

# DiffServ – shrnutí

- Výrazné zjednodušení
  - Shora omezený počet různých tříd
  - Nezávislý na počtu uživatelů
- Pouze statistické garance
- Rozdělení sítě na „hranici“ a „jádro“
  - Odlišná role
    - \* „Intelligence“ soustředěna na hranici
    - \* Jádro pouze „přesouvá“ pakety co největší rychlostí

# ATM a kvalita služeb

- Asynchronous Transfer Mode
- Snaha o spojení datově orientovaných a přepojovaných sítí
- Příliš dlouhý standardizační proces
- V současné době považováno za mrtvou větev
  - Používá se
  - ale dále vývoj nepokračuje (není definováno pro skutečně vysoké rychlosti přenosu)
- Zajímavé koncepty v oblasti QoS
  - Zejména s ohledem na dobu vzniku

# ATM – kategorie služeb

- Služby reálného času
  - CBR (Constant Bit Rate)
  - rt-VBR (real-time Variable Bit Rate)
- Ostatní služby
  - nrt-VBR (non-real-time Variable Bit Rate)
  - ABR (Available Bit Rate)
  - UBR (Unspecified Bit Rate)



# Služby reálného času

- CBR
  - Abstrakce vlastní (soukromé) linky
  - Trvalý datový tok
- rt-VBR
  - Proměnlivý datový tok
  - Umožňuje flexibilnější alokaci kapacity sítě

# Ostatní služby

## ■ nrt-VBR

- Proměnlivý datový tok se slabšími požadavky na zpoždění a rozptyl
- Definována
  - \* Špičkovou kapacitou (počet buněk za sekundu)
  - \* Průměrnou přenosovou kapacitou
  - \* „Burstiness“, tj. parametrem definujícím variabilitu toku v čase

## ■ UBR

- Emulace klasických (best effort) datových sítí
- Využívá kapacitu, která „zbude“ (není využita) VBR a CBR toky

# ABR

- Řízený tok
  - Reaguje na aktuální zatížení sítě
- Parametry
  - Špičková kapacita
  - Minimální kapacita

# Atributy/charakteristiky toku dat

- Popis toku
  - Použity při sestavování obvodu
- Parametry kvality služby
  - Charakterizují výkon (kvalitu) konkrétního spojení
- Další charakteristiky specifické pro ABR

# Popis toku

- Charakteristika zdroje
  - Špičková kapacita (PCR)
  - Průměrná (udržitelná) kapacita (SBR)
  - Minimální kapacita (MBR)
  - Maximální velikost špičky (MBS, počet buněk)
- Tolerance variace zpoždění (CDVT); vnesena sítí
- Definice shody
  - Dnes analogie se značkováním v DiffServu

# QoS parametry

- Variace zpoždění mezi špičkami (Peak-to-peak cell delay variation, CDV)
  - dohodnuto, na rozdíl od CDVT, které je nastaveno
  - musí vyhovovat požadavkům uživatele
  - CDVT představuje horní mez CDV
- Maximální zpoždění (Maximum cell transfer delay, maxCTD)
- Podíl ztracených buněk (Cell loss ratio, CLR)
  - Konkrétní buňky používají navíc CLP pole (Cell Loss Priority, 1 bit)

# Síťový kontrakt

- Při ustavení spojení
  - Kategorie služby
  - Parametry toku
  - Požadované a akceptovatelné hodnoty QoS parametrů
- Požadavek akceptován pouze je-li dostatek prostředků
  - Uzavřen *síťový kontrakt* (traffic contract)
  - Vytvořen virtuální okruh (VC, statický (PVS) nebo dynamicky (SCV))

# Řízení užití sítě

- Usage Parameter Control, UPC
  - Monitoruje síť
  - Kontroluje shodu kontraktu s realitou
  - Zakročuje proti porušení kontraktu
- Použití již prezentovaných principů
  - Token/leaky bucket algoritmy
  - Využití CLP pole



# ABR

- Používá uzavřený okruh (určitá analogie TCP)
- Základní parametry
  - Povolená rychlost (Allowed Cell Rate, ACR)
  - Minimální rychlost (Minimum Cell Rate, MCR)
  - Špičková rychlost (Peak Cell Rate, PCR)
  - Počáteční rychlost (Initial Cell Rate, ICR)
- Zpětná vazba

# ABR – zpětná vazba

- Začátek přenosu:  $ACR = ICR$
- Hodnota ACR se dynamicky upravuje podle stavu sítě
- *Resource Management* (RM) buňky

## Tři pole

- Indikace *zahlcení* (CI)
- Indikace *ne zvýšení* (NI)
- *Explicitní přenosová rychlost* (Explicit Cell Rate, ECR)

# ABR – algoritmus

if CI = 1

    sniž ACR proporčně k jeho aktuální hodnotě,  
    avšak nejvýše na MCR

else if NI = 0

    zvyš ACR proporčně k hodnotě PCR, avšak nejvýše na PCR

if ACR > ER

    ACR = max(ER, MCR)

# ABR – řídicí buňky

- Forward RM (FRM)
  - Generovány zdrojem dat (obvykle každá 32. buňka)
  - Pole nastavena zdrojem (CI=0, NI=0 nebo 1, ER nějak)
  - Pole modifikována ATM přepínači
- Backward RM (BRM)
  - Reakce příjemce na obdrženou FRM
  - Pole opět modifikována přepínači

# Provoz s real-time vlastnostmi

- Typicky multimediální data
  - nízké zpoždění
  - nízký rozptyl
  - nemusí mít vysoké požadavky na propustnost
- Hard a soft real-time aplikace:
  - Soft: toleruje určitý (malý) počet/frekvenci výpadků
  - Hard: žádná tolerance k výpadkům

# RTP

- Real-time protocol, RTP (RFC 1889)
- Vhodný pro soft real-time provoz
- Základní principy:
  - Rámce na aplikační úrovni
  - Integrace přenosových vrstev
- Určitá náhrada TCP tam, kde striktní požadavky TCP nejsou třeba
  - aplikace toleruje jisté výpadky
  - aplikace nemůže tolerovat přílišné zpoždění (vnesené retransmisí)

# Rámce na aplikační úrovni

- Aplikace může akceptovat určitou míru výpadků
  - O retransmisi musí rozhodnout aplikace
  - Místo úplné retransmise může zvolit méně náročný formát dat
  - V případě retransmise může poslat jinak formátovaná data
  - Nemusí ukládat všechna data pro případ retransmise (data se přepočítají nebo se nepošlou)

# Integrace přenosových vrstev

- Propojení několika v ISO OSI modelu oddělených vrstev
- Hlavním cílem efektivita
- RTP běží nad UDP, přidává např. číslování, ale neposkytuje plný zabezpečený transportní protokol
  - Musí být kombinováno s aplikační vrstvou
  - Ta dodává další vlastnosti, patřící aplikační vrstvě (např. zabezpečení, retransmisi, ...)



# RTP – základy

- RTP garantuje přenos v rámci relací (session)
- Každá relace je definována:
  - RTP číslem portu
  - RTCP číslem portu (RTCP je Real-time Control Protocol)
  - IP adresami účastníků; může jít buď o skupinu unicastových IP adres nebo o adresu multicastové skupiny
- Sestavení relace není součástí RTP/RTCP
- Primárně určen pro multicastové prostředí

# RTP – hlavička

- \* Verze (2bity)
- \* Zarovnání (1bit), signalizuje, zda je RTP paket zarovnaný
- \* Rozšíření (1bit), experimentální
- \* Počet CSRC (4bity), počet CSRC identifikátorů v datagramu
- \* Značka (1bit), interpretace závisí na typu dat (např. konec videorámce)
- \* Typ dat (7bitů), např. GSM audio (3) nebo nv video (28)
- \* Pořadové číslo (16bitů)
- \* Časová známka (32bitů), generována hodinami zdroje, jednotka je funkcí typu dat
- \* Identifikátor zdroje (32bitů), jednoznačný identifikátor vysílacího zdroje v rámci relace; následován jedním nebo více identifikátorů zdrojů dat

# RTP – relays

- RTP používá relays (přenašeče)
  - pokud není možno data doručit přímo, je využít prostředník
  - vůči zdroji vystupuje jako příjemce
  - vůči skutečnému příjemci vystupuje jako zdroj
- Dva typy prostředníků
  - přenašeče (translators)
  - spojovače (mixers)

# RTP – translators

- Jednoduché prostředníky, pouze jeden zdroj dat
- Mohou měnit formát dat
- Mohou použít jiné přenosové protokoly nižších úrovní (jiný pro příjem a jiný pro vysílání)
- Příklady
  - Převod vysokokvalitního videa do méně zatěžujícího formátu
  - Převod multicastu do jednoho nebo více unicastových proudů (do míst, kam nevede multicast)

# RTP – mixers

- Přijímá více vstupních proudů
- Kombinuje (mixuje) vstupní proudy a vytváří jeden nebo více výstupních
- Přidává vlastní časové známky do spojeného proudu (skládané proudy nemusí být synchronní)
- Příklad
  - Spojování audio proudů (obvykle pouze jeden zdroj vysílá)

# RTCP – principy

- RTP přenáší pouze uživatelská data
- Řídící informaci obstarává RT Control Protocol
- Funguje na principu multicastu
- Čtyři základní funkce:
  - Řízení kvality služby
  - Identifikace
  - Odhad počtu účastníků relace
  - Řízení relací

# RTCP – řízení kvality služby

- Sender reports
  - Obsahují informaci o vysílaných datech (rychlost, kvalita, ...)
- Receiver reports
  - Informace o problémech (ztráty, zpoždění, rozptyl, ...)

# RTP – shrnutí

- Protokol určený pro specifikaci real-time dat
- Musí být doplněn konkrétní aplikační vrstvou (přenos zvuku či audia v konkrétním kódování, ...)
- Určen pro multicastové vysílání, použitelný i v rámci unicastu
- Neřeší vlastní problém ustavení cesty, která by garantovala potřebné parametry přenosu
  - Dobře kombinovatelný s RSVP