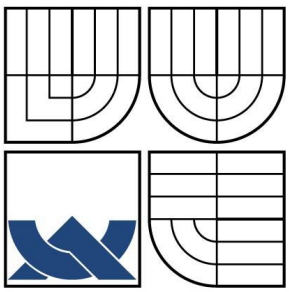
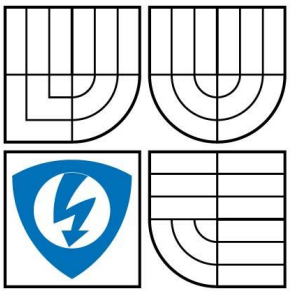


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

POROVNÁNÍ ARCHITEKTUR SYSTÉMOVÝCH DATABÁZÍ MIB URČENÝCH PRO ŘÍZENÍ KVALITY SLUŽEB V DATOVÝCH SÍTÍCH

MIB database for management QoS in data networks comparison

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE **TOMÁŠ JELÍNEK**
AUTHOR

VEDOUCÍ PRÁCE **ING. KAROL MOLNÁR, PH.D.**
SUPERVISOR

BRNO 2007

Poděkování

Rád bych poděkoval vedoucímu mé práce za jeho konzultace, které mi umožnily pochopit danou problematiku a své přítelkyni za tolerantní přístup, když jsem místo zařizování svatby pracoval na své bakalářské práci.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Porovnání architektur systémových databází MIB určených pro řízení kvality služeb v datových sítích jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v bakalářské práci a uvedeny v seznamu literatury na konci bakalářské práce. Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne .....

.....
Podpis autora

Seznam zkratek

SNMP (Simple Network)

NMS (Network Management Station)

MIB (Management Information Base)

SMI (Structure of Management Information)

QoS (Quality of Service)

UDP (User Datagram Protocol)

ToS (Type of Service)

DSCP (Differentiated Service Code Point)

Diffserv (Differentiated Service)

PHB (Per Hop Behaviour)

AF (Assured Forwading)

EF (Expected Forwarding)

SLA (Service Level Agreement)

BA(Behaviour Agregate)

MF (Multi—Field)

TCA (Traffic Conditioning Agreement)

PIB (Policy Information Base)

PDP (Policy Decision Point)

PRC (Provisioning Class)

1 Úvod

Internet byl vybudován v době studené války, kdy USA potřebovalo spojit nejdůležitější armádní, vládní, akademické a jiné strategické výpočetní systémy. Nejvhodnější systém, který splňoval požadavky ministerstva obrany USA, byl navržen firmou Rand Cooperation v roce 1961. Na základě tohoto systému byl v roce 1968 vybudován základ sítě ARPANET. Později připojováním dalších a dalších sítí k ARPANETu vznikla síť Internet a ARPANET dále plnil pouze úlohu páteřní sítě.

Pro provoz Internetu platily v počátečních letech pravidla:

Žádnému typu provozu nebude odmítnut přístup na síť

Ke kterémukoli typu provozu se bude přistupovat rovnoprávně

Provoz bude přenesen co nejlepším způsobem (Best Effort)

Tato pravidla však v pozdějších letech přestávala stačit v důsledku rozšiřování datových sítí a množství nových služeb využívajících síťové propojení přes Internet. Na rychle se zvyšující velikost sítí bylo potřeba reagovat změnami systému správy a konfigurace sítí. V počátcích internetu veškerá konfigurace síťových zařízení probíhala ručně přímo na konkrétních zařízeních. Tato situace nebyla, při tak velkém nárůstu velikosti sítí, udržitelná. Hledalo se řešení usnadňující správu sítí Velké usnadnění přinesly dvě změny, automatizace konfigurování síťových zařízení a vzdálená správa síťových zařízení.

Dále bylo potřeba vyřešit rychle rostoucí množství služeb využívajících síťové služby a Internet. Různé služby měli různé nároky a požadavky na přenos sítí Internet. Pro názornou představu různých nároků na přenos sítí může posloužit rozdíl mezi přenosem hlasu a videa. Při telefonickém rozhovoru přes síť Internet je nejdůležitější veličinou zpoždění paketů. Pokud by jednotlivé pakety měli výrazně jiná zpoždění, asi bychom se nedokázali domluvit, hovor by byl v nejlepším případě hodně trhaný a v nejhorším by se pravděpodobně změnilo i pořadí slov. Pro přenos videa jsou nejpotřebnější hned veličiny dvě, odchylka ve zpoždění paketů a ztráta paketů. Hledalo se řešení, které by zajistilo různým službám různou kvalitu přenosu jimi požadovanou, protože původní řešení rovnoprávnost a přenos co nejlepším způsobem (Best Effort) již nestačili. [7,8]

2 Protokol SNMP

Mechanizmus pro vzdálenou správu síťových zařízení byl nazván Simple Network Management Protocol nebo dnes více používaná zkratka SNMP. Tento velmi univerzální protokol pomohl řešit nejen vzdálenou správu, ale i systém řízení kvality služeb. SNMP (Simple Network Management Protocol) je asynchronní protokol založený na jednání síťového agenta a manažera. Jednání probíhá formou dotazů a odpovědí. Vyměňované zprávy používají UDP protokol tzn. službu bez spojení, UDP porty 161, 162. Pro dotaz i pro odpověď se používá pouze jeden datagram. [9,10]

2.1 Manažer

Ve stanici síťového managementu (Network Management Station) jsou sbírána data z určité oblasti pomocí tzv. manažeru, což je software, např. Tivoli Netview, nainstalovný ve stanici síťového managementu, který sbírá data ze síťových zařízení umístěných v oblasti, kterou spravuje. Kromě sběru dat ze síťových zařízení je zde vykonávána většina řídících úkolů

např. monitoring, konfigurace,.... [9,10]

2.2 Agent

Je software nainstalovaný na koncovém síťovém zařízení, které komunikuje a posílá data do stanice síťového managementu - Manažeru. Data má agent uložena v databázi MIB (Management Information Base). Komunikace mezi agentem a manažerem probíhá v sítích TCP/IP pomocí protokolu SNMP. [9,10]

2.3 Databáze řídících informací (Management Information Base)

Databáze řídících informací MIB (Management Information Base) je databáze objektů popisujících chování síťového zařízení. Objekt je popsán a definován pomocí standartizované struktury řídících informací (Structure of Management Information – SMI). Popis objektu se skládá ze tří částí tj. název, definice typu a kódování. Databáze řídících informací je spravována agentem.

2.3.1 Struktura databáze řídících informací

Objekty databáze řídících informací MIB (Management Information Base) jsou uspořádány do stromové struktury Obr.1. Na jednotlivé objekty ve stromové struktuře se můžeme odkazovat pomocí jmen, např. iso.org.dod.internet.private nebo pomocí identifikačních čísel 1.3.6.1.4. Oba zápisy mají stejný význam. [6,10]

Obr.1 Stromová struktura MIB databáze

2.4 Operace protokolu SNMP

Základem pro spolupráci protokolu SNMP a MIB databází jsou dva příkazy (get, set):

-get přečte hodnotu v databázi MIB:

- get-request — žádost o získání hodnoty určité proměnné
- get-next-request — žádost o získání hodnoty proměnné bez znalosti jejího přesného jména.
- get-response — odpověď na žádost
- get-bulk — všechny instance ve spravovaném objektu možno načíst v jediné opraci (Nemusíme opakovaně zadávat příkaz get-next), možno použít pouze od SNMP verze 2

-set nastaví hodnotu v databázi MIB:

- set-request — nastavení hodnoty určité proměnné

-trap zpráva způsobená událostí, např. odpojení koncového zařízení z portu síťového zařízení

-inform nařizuje stanici NMS předávání informací přijatých od agentů dalším stanicím NMS

SNMP protokol má v současné době tři verze. [9]

3 Kvalita služby QoS (Quality of Service)

Vzniklá potřeba rozlišit různé nároky a požadavky, různých služeb a aplikací byla řešena kombinací parametrů a nazvána kvalita služby QoS (Quality of Service).

Nejdůležitější byly shledány parametry ztráta paketů, zpoždění a kolísání zpoždění.[7]

3.1.1 Ztráta paketů

Pokud provoz v síti naroste nad kapacitu dané sítě, pak síťová zařízení nestíhají odbavovat příchozí datagramy. Následně jejich fronty přetečou a další datagramy musí být zahozeny.

Tento parametr je nejdůležitější pro aplikace používající transportní protokol UDP. Jsou to většinou aplikace pracující v reálném čase bez mechanizmu kontroly doručení. [7]

3.1.2 Zpoždění

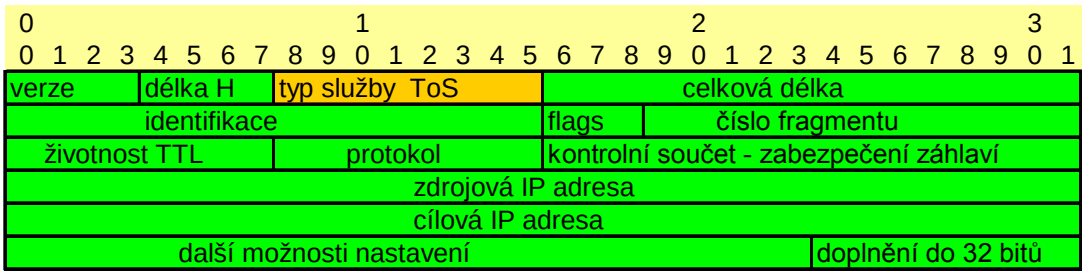
Zpoždění je doba, za kterou datagram urazí cestu od vysílacího zařízení k přijímacímu zařízení. Zpoždění má více příčin. Může to být např. přípravou paketů pro přenos médiem, fyzické omezení sítě, zpoždění ve frontě na směrování . [7]

3.1.3 Kolísání zpoždění

Každý datagram má jiné podmínky v přenosu od vysílacího zařízení k přijímacímu zařízení. Datagramy mohou putovat jinou cestou, mít různou velikost fronty v síťových zařízeních a těmito vlivy vznikají mezi datagramy jiné časové intervaly v příjmu datagramu u přijímacího zařízení a při vyslání datagramu u vysílacího zařízení. [7]

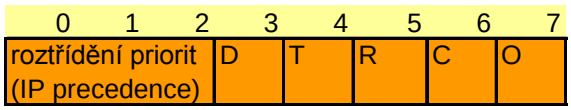
3.2 Struktura hlavičky IP datagramu

Pro rozčlenění služeb podle jejich nároků bylo potřeba nalézt místo v paketu, které by tento mechanismus umožnilo. V hlavičce IP protokolu, viz Obr2, se nabízely dva možné mechanizmy v poli typ služeb ToS (Type of Service). Různé typy služeb ToS (Type of Service), které určují různé chování paketu v cestě Internetem. Druhou možností byla část pole ToS tříbitové pole roztřídění priorit (IP Precedence).



Obr.2 Hlavička IP datagramu verze 4

Tříbitové pole IP Precedence z Obr.3, které třídilo provoz do osmi tříd služeb, bylo nahrazeno



Obr.3 Pole služeb ToS detail

DSCP (Differentiated Service Code Point) na Obr.4 používajícím šest bitů ,tři bity zIP precedence a tři bity zrůzných typů služeb, které označovali D-zpoždění (Delay), T-propustnost (Troughput), R-spolehlivost (Reliability), poslední dva bity z pole typ služeb ToS byly rezervovány pro použití vaktualizacích managementu, směrování nebo explicitní oznámení o zahlcení vIP. Podíváme se podrobněji na jednotlivé mechanizmy a principy differencovaných služeb. [7]

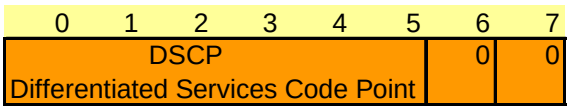
4 Diferencované služby (DiffServ)

Diffserv je metoda, která řeší problém rozčlenění služeb podle jejich nároků na síť. Aplikace jsou přiděleny do jednotlivých tříd a sledujeme pouze jednotlivé třídy různých služeb místo původně sledovaného toku datagramů.

4.1 DSCP - Differentiated Service Code Point

DSCP je v současné době nejvýkonější možnost řešení kvality služeb (QoS).

Jedná se o prostor 6 bitů z pole Diffserv na Obr.4, které určují způsob zacházení s datagramem při průchodu sítí podporující tuto metodu a umožňující rozčlenění na 2^6 tříd, to je 64 tříd, v rámci mechanizmu diferencovaných služeb (Diffserv). Pokud je přijat datagram u kterého není kód DSCP rozpoznatelný, je s ním nakládáno podle standartního chování (Default Behaviour) a daný kód zůstává stejný. [1]



Obr.4 Pole Diffserv

4.2 PHB (Per Hop Behaviour)

PHB (Per Hop Behaviour) je definované chování datagramů určené označením v poli DSCP. Dané chování je rozděleno do tří kategorií.

4.2.1 Standartní PHB (Per Hop Behaviour)

Standartní chování (default PHB) musí být dostupné v každém Diffserv vyhovujícím zařízení. Jedná se o běžné chování síťových zařízení známé pod jménem Best Effort a dostupné ve všech směrovačích podle standartu rfc 1812. Doporučená hodnota kódu DSCP pro tento způsob chování je “000000“. Datagram s tímto chováním může být překódován na jiné chování na hranici Diffserv domény. [1]

4.2.2 Zajištěné předávání (Assured Forwarding)

Služba je určena k přenosu datagramů sítí garantovanou rychlostí. Používá TCP protokol na základě stanovení priorit pro různé typy provozu. Pokud dojde k přetížení sítě, bude zahozen předem určený typ paketu. Zajištěné předávání (Assured Forwarding) poskytuje čtyři nezávislé třídy a v každé třídě jsou dále tři úrovně priority zničení. Každá třída má jinak přidělené zdroje např. šířku pásma, množství paměti, aby aplikace získaly zdroje rozdílné požadované úrovně. Přehled tříd je znázorněn v Tab.1.[1,7]

priorita zahození	class1	class2	class3	class4
nízká (Low)	001010 (AF11)	010010 (AF21)	011010 (AF31)	100010 (AF41)
střední (Medium)	001100 (AF12)	010100 (AF22)	011100 (AF32)	100100 (AF42)
vysoká (High)	001110 (AF13)	010110 (AF23)	011110 (AF33)	100110 (AF43)

Tab.1 Priorita zahození

4.2.3 Urychlené předávání (Expedited Forwarding)

Urychlené předávání (Expedited Forwarding) poskytuje absolutní záruky na kolísání zpoždění pro svou třídu provozu což je, v podstatě, přímé spojení, které zvyšuje zatížení sítě. Tento mechanismus je velmi složitý.

Urychlené předávání EF (Expedited Forwarding) má DSCP 101110.[7]

4.3 Architektura Diffserv modelu

Diffserv architektura je založena na jednoduchém modelu, kde je provoz vstupující do sítě tříděn podle možných podmínek hned na hranici sítě a přidělen do rozdílných skupin sdruženého chování (Behavior Aggregate). Agregát chování (Behavior Aggregate) je skupina paketů se stejným kódem v poli DSCP. Tato skupina při průchodu sítí vyžaduje stejný přístup, stejné chování a i stejnou prioritu odbavení. Uvnitř sítě jsou datagramy přeposílány v závislosti na PHB (Per Hop Behaviour) spojené s DSCP. [2]

4.3.1 Diffserv doména (Differentiated Services Domain)

Diffserv doména (Differentiated Services Domain) je sada závislých Diffserv uzlů, síťových zařízení, které řídí společné služby a nastavuje PHB (Per Hop Behaviour) skupiny v každém uzlu. Na hranici Diffserv domény jsou hraniční směrovače, kde je tříděn příchozí provoz a je zajištěno, aby provoz procházející doménou byl řádně nastaven na nějakou PHB skupinu (Per Hop Behaviour Group) podporovanou v doméně. [2]

4.3.2 Třídění a úprava provozu

Diffserv služby jsou rozšířeny v doméně zřízením SLA (Service Level Agreement), které určuje třídění datagramů a přeznačení pravidel. Úpravu provozu má na starost více nástrojů . [2]

4.3.2.1 Třídič datagramů (Classifier)

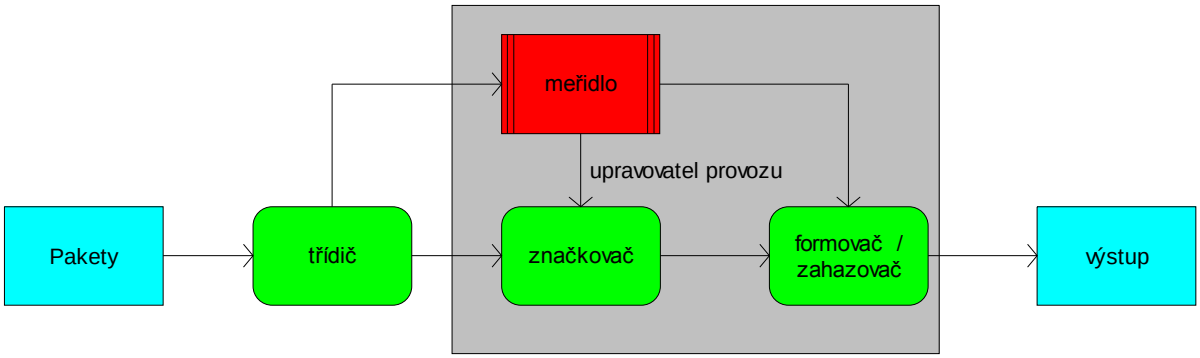
Třídič vybírá datagramy podle obsahu části hlavičky datagramu a definovaných pravidel. Jsou dva typy třídičů. BA (Behaviour Agregate) třídič používá k třídění princip založený pouze na DSCP. MF (Multi—Field) třídič vybírá datagramy podle hodnoty kombinace jedné nebo více polí hlavičky datagramu např. adresa přijímacího nebo vysílacího zařízení, Diffserv pole, ID protokol, číslo zdrojového a cílového portu. Třídiče jsou používány k řízení datagramů. Třídiče jsou konfigurovány podle příslušného TCA (Traffic Conditioning Agreement). [2]

4.3.2.2 Profily provozu (Traffic profiles)

Profily provozu (Traffic profiles) určují dočasné vlastnosti provozu vybraného třídičem. Profil poskytuje pravidla pro určení, zda je jednotlivý datagram součástí profilu nebo do něj nepatří. Příkladem může být profil se systémem Token Bucket. [2]

4.3.2.3 Systém úpravy provozu- upravovatel provozu (Traffic Conditioners)

Upravovatel provozu (Traffic Conditioner) na Obr.5 se může skládat z následujících částí: měřidlo (Meter), značkovač (Marker), formovač (Shaper) a zahazovač (Dropper).Proud datagramů nejdříve prochází třídičem, který směřuje datagramy k dalším logickým stupňům upravovatele provozu (Traffic Conditioner). Ve vhodných situacích srovnáme proud datagramů s profilem provozu (Traffic profiles) pomocí části upravovatele provozu zvanou měřidlo (Meter). Stav měřidla může ovlivnit značení (Marking), zahození (Dropping) a formování (Shaping). V době, kdy datagramy opouští upravovatel (Traffic Conditioner) hraničního Diffserv zařízení, musí mít všechny řádně nastaven vhodný DSCP kód. [2]



Obr.5 Systém úpravy paketů (datagramů)

4.3.2.3.1 Měřidlo (Meter)

Měřidlo (Meter) porovnává dočasné vlastnosti proudu datagramů vybraných třídičem (Classifier) s profilem provozu (Traffic profiles) určeným v TCA (Traffic Conditioning Agreement). Měřidlo předává získané informace ostatním funkcím a mechanizmům v upravovateli provozu (Traffic Conditioner) ke spuštění jednotlivých akcí se všemi pakety v obou směrech, z Diffserv domény i do Diffserv domény.[2]

4.3.2.3.2 Značkovač (Marker)

Značkovač (Marker) nastavuje DSCP jednotlivých datagramů v jejich hlavičce. Tím přiřadí označený datagram k jednotlivým skupinám sdruženého chování (Agregate Behaviour) Značkovač (Marker) může datagram označit jednoduchým Diffserv kódem (DSCP) nebo ho může označit jedním ze sady Diffserv kódů použitých k vybrání PHB (Per Hop Behaviour) chování ze skupiny různých PHB chování (Per Hop Behaviour group). [2]

4.3.2.3.3 Formovač (Shaper)

Formovač (Shaper) časuje resp. zpožďuje datagramy podle jejich profilu provozu (Traffic Profiles), který určuje prioritu odbavení jednotlivých toků datagramů. Formovač (Shaper) má ohraničenou velikost bufferu, takže pokud není v bufferu dostatek místa k zadržení datagramů, jsou tyto zahozeny a zničeny. [2]

4.3.2.3.4 Zahazovač (Dropper)

Zahazovač (Dropper) zahazuje určité množství paketů, které není možno odbavit, ve shodě s profilem provozu (Traffic profiles). Zahazovač může být součástí Formovače (Shaper). [2]

4.4 Diffserv QoS PIB (Policy Information Base)

Struktura PIB (Policy Information Base) vychází z potřeby konfigurovat postupné zpracování datagramu Diffserv metodou a parametrizovat toto zpracování. Postupné zpracovávání a jeho parametrizace jsou navzájem odděleny v celé struktuře PIB formou oddělených tabulek a definicí. V této struktuře jsou nejdůležitejší části místo prosazování zásad PEP (Policy Enformceement Point) a místo stanovení zásad PDP (Policy Decision Popint). PEP je umístěn přímo na síťových zařízeních a vykonává akce určené PDP. PDP zpracovává události v síti a posílá příkazy do PEP, který je vykoná. Pravidla podle kterých se rozhoduje jsou uloženy v repozitáři zásad, což může být třeba databázový server. PEP a PDP komunikují pomocí protokolu COPS (Common Open Policy Service).COPS je stavový TCP protokol pracující na principu žádost-odpověď. [10]

4.4.1 Nejdůležitější funkce PIB (Policy Information Base)

-Funkce pro třídění datagramů v závislosti na nastavených pravidlech

-Funkce která určí, zda je nebo není tok datagramů částí měřených parametrů

-Funkce nastavující v datagramu Diffserv pole Diffserv kódem (DSCP)

-Funkce aplikující postupy zahození datagramů

-Funkce zajišťující výstupní frontu datagramů ve vhodném pořadí [10]

4.4.2 Tabulky tvořící strukturu PIB (Policy Information Base)

-Tabulka cesty (Data path table):

Tato tabulka popisuje začátek cesty dat Diffserv mechanizmem v jednotlivých Diffserv zařízeních.

-Tabulka třídiče (Classifier Tables):

Obecná konstrukce, která je rozšiřitelná pro určité skupiny nebo filtry podporující třídění datagramů.

-Tabulka měřidla (Meter Tables)

-Tabulka činnosti (Action Tables)

-Tabula zahazovače(Algorithmic Dropper Tables):

Obecná rozšiřitelná konstrukce podporující funkci aplikující postupy pro zahození datagramů.

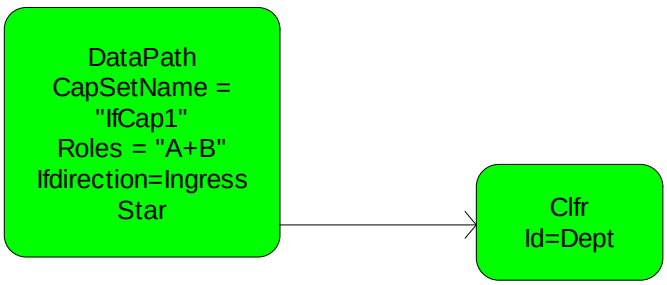
-Tabulka formovače a plánovače (Queue and Scheduler Tables)

5 Použití PIB (Policy Information Base)

Podíváme se na možný příklad různých tabulek PIB použitých v Diffserv zařízení. Všechny použité parametry jsou nadefinovány přímo v samotné PIB jednotce. [6]

5.1 Příklad cesty (Data Path example)

Všechny záznamy tabulky Cesta dat (Data Path Table) jsou použity pro určitý typ rozhraní ovládajícího tok datagramů určitým směrem s určitou kombinací funkcí. Na Obr.6 máme příklad jednoho takového záznamu. Vstup "IfCap1"



Obr.6 Tabulka cesty (Data Path example)

5.2 Třídič a příklad základní struktury

Použijeme tabulky třídiče zPIB (Policy Information Base) k vytvoření víceúrovňového třídiče. Řekněme, že chceme vybudovat následující funkci třídění k vytvoření různých toků podle příslušnosti k oddělení a různým typům aplikací:

if (Dept1) then take Dept1-action

{

if (Appl1) then take Dept1-Appl1-action.

if (Appl2) then take Dept1-Appl2-action.

if (Appl3) then take Dept1-Appl3-action.

}

if (Dept2) then take Dept2-action

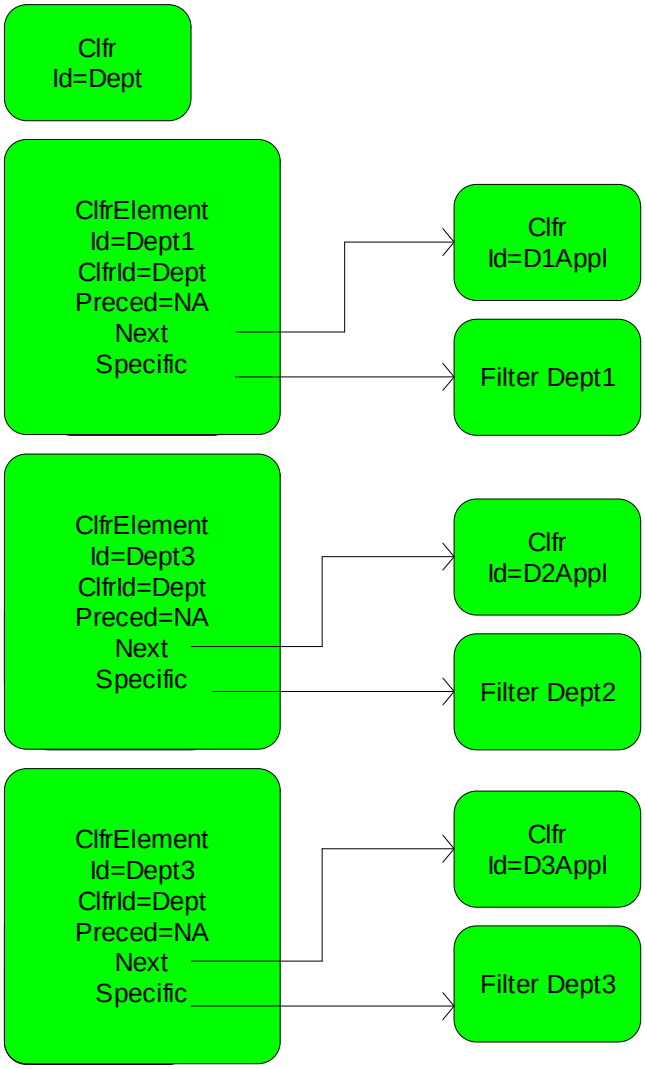
{

if (Appl1) then take Dept2-Appl1-action.

if (Appl2) then take Dept2-Appl2-action.

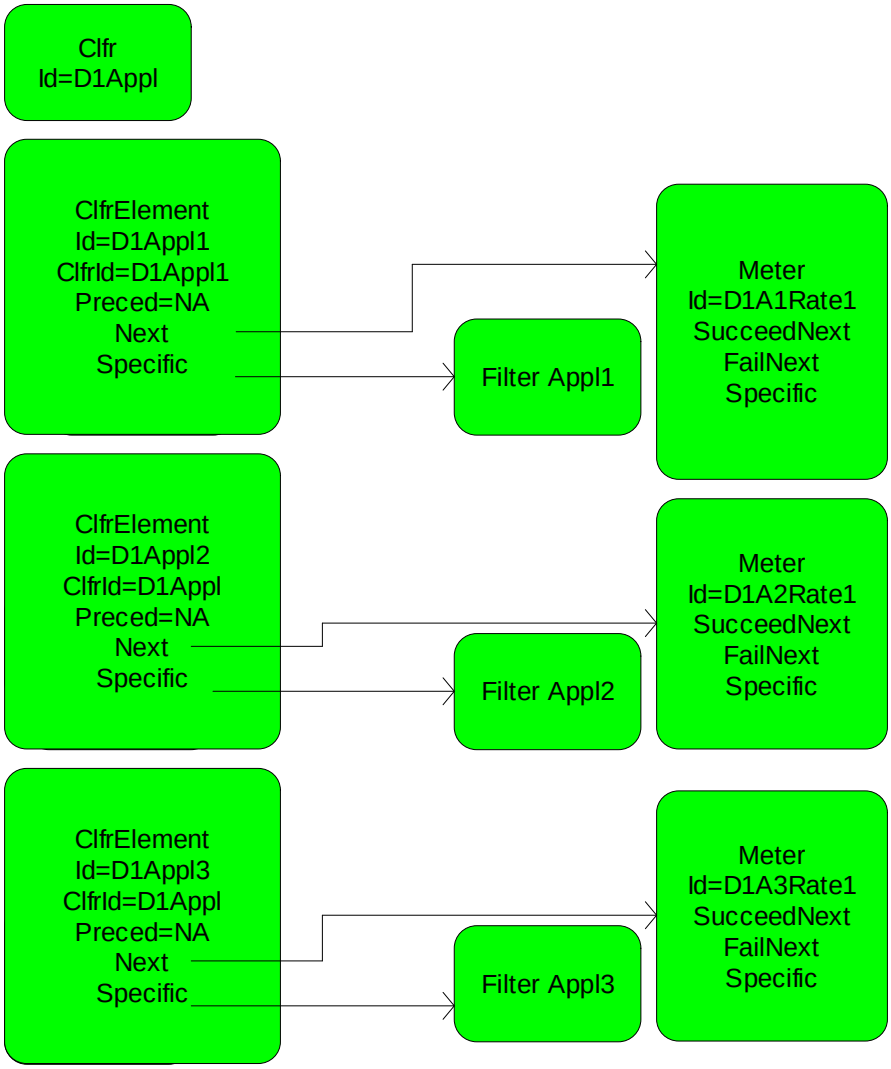
```
  if (Appl3) then take Dept2-Appl3-action.
}
if (Dept3) then take Dept3-action
{
  if (Appl1) then take Dept3-Appl1-action.
  if (Appl2) then take Dept3-Appl2-action.
  if (Appl3) then take Dept3-Appl3-action.
}
```

Tuto třídící logiku využijeme v následujících vstupech PIB tabulky se dvěmi úrovněmi třídění. Jedno třídění bude pro oddělení (Department) a druhé pro typ aplikace (Application). Mechanizmus pro třídění toků datagramů podle oddělení vidíme na Obr.7 .



Obr.7 Mechanizmus třídění podle oddělení

Mechanizmus třídění podle typu aplikace si ukážeme pouze pro oddělení č.1,viz Obr.7, pro další oddělení platí analogický postup, kde by se pouze zaměnilo číslo oddělení, což by znamenalo v Obr.8 záměnu D1 za D2 resp. D3.



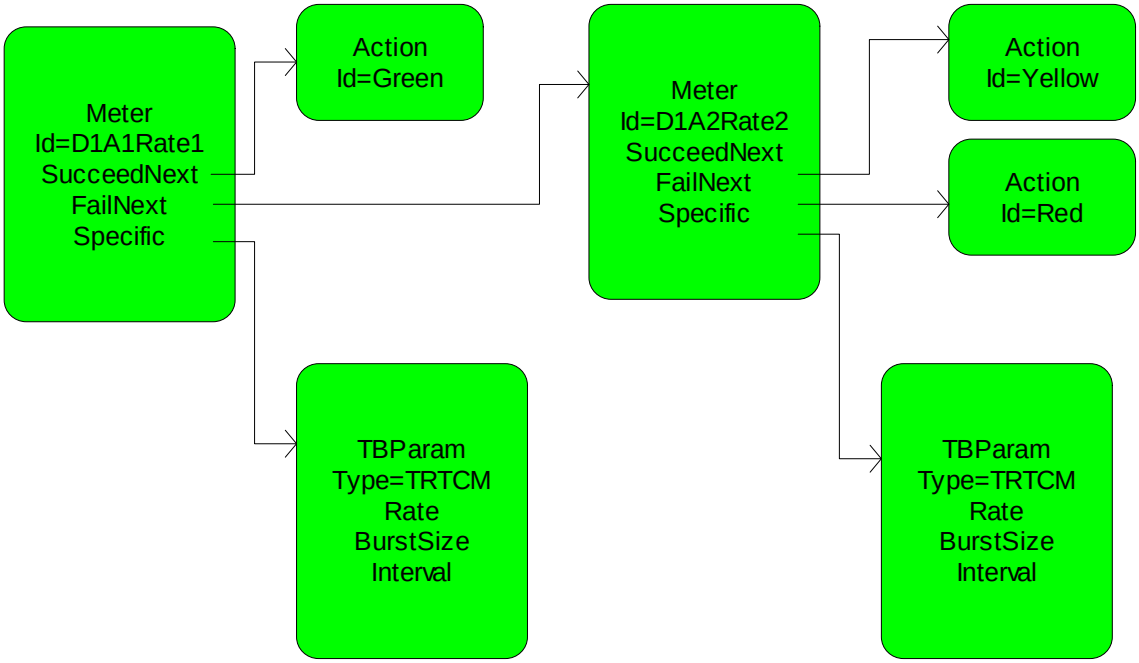
Obr.8 Mechanizmus třídění podle typu aplikace pro oddělení č.1

Po třídiči následuje v cestě mechanizmu Diffserv Měřidlo (Meter). [6]

5.3 Měřič (Meter) a jeho základní struktura

V tomto příkladu použijeme jednoduchý model Měřidla Obr.9, který použije dva záznamy v tabulce Měřidla s různými rychlostmi a dva záznamy v TBParam tabulce. První úroveň TBParam záznamu je propojena s informací o rychlosti. Ostatní měřidla budou záviset na třídě služby každého použitého roztříděného toku datagramů, ale jejich konstrukce bude podobná konstrukci Měřidla (Meter) z tohoto příkladu. Záznamy tabulky TBParam mohou být sdíleny více záznamy v tabulce Měřidla (Meter).

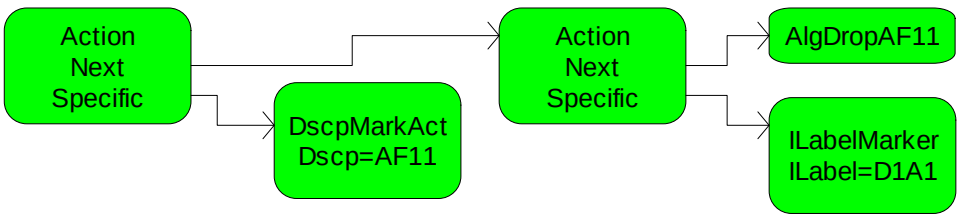
Po Měřidle (Meter) následuje v cestě mechanizmu Diffserv tabulka činnosti (Action table).



Obr.9 Příklad mechanizmu měřiče (Meter)

5.4 Příklad zpracování provozu paketů (Action example)

V této ukázce navazujeme na činnost (Action) Green z předchozího mechanizmu Měřidla (Meter) z Obr.9. Připomeneme si, že tato činnost je součástí větve D1A1Rate1 SucceedNext zanmenající tok datagramů z oddělení č.1, aplikace č.1 se závaznou rychlostí a s limitovaným rozsahem pro tento tok datagramů. Teď bychom rádi označili tento tok určitým DSCP a také označením daného síťového zařízení. V příkladu na Obr.10 je použita třída PRC frwkILabelMarker, která ukáže vnitřní označení zařízení použité k ukázání mikro-toku zásobujícímu celkový AF(Assured Forwarding) tok .



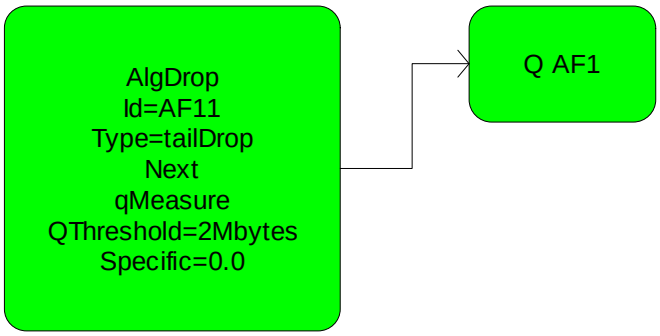
Obr.10 Záznamy tabulky zpracování provozu paketů (datagramům)

5.5 Příklad zahazovače (Dropper)

Pokračujeme v naší ukázce fungování Diffserv mechanizmu, konkrétně jsme v kapitole o činnosti skončili s tokem AF11. Vyzkoušíme si tři různé nastavení zahazovače od jednoduchých ke komplexnějším. [6]

5.5.1 Zahazovač (Tail dropper)

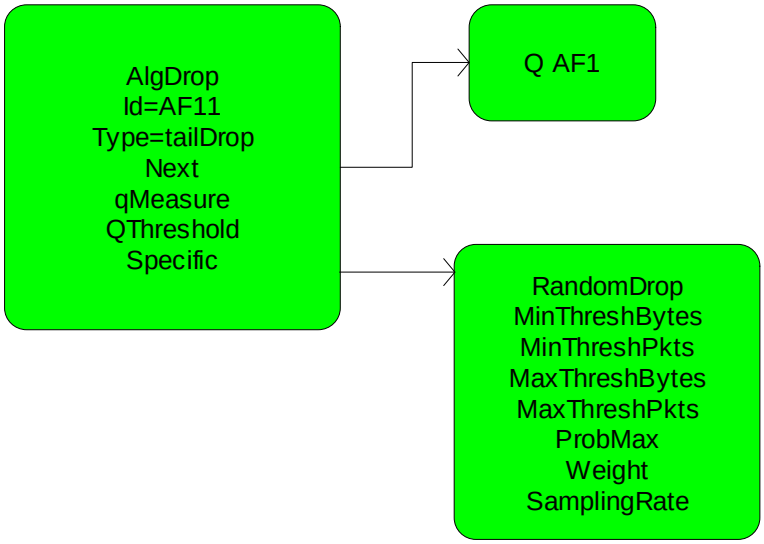
Zahazovač (Tail dropper) Obr. 11 patří k nejjednoduším. V naší ukázce pouze zahodíme část toku, která překročí kapacitu bufferu 2MB.



Obr.11 Zahazovač (Tail Dropper)

5.5.2 Jednoduchá fronta s náhodným zahazováním

Náhodné zahazování předvedeme se záznamem dsRandomDropEntry. Záznam dsAlgDropQThreshold obsahuje maximum průměrné šířky fronty. Změřená šířka fronty je v záznamu dsAlgDropQMeasure. Zbytek parametrů pro náhodné zahazování je určeno záznamy v dsAlgDropSpecific. V naší ukázce odkazují dsAlgDropNext i dsAlgDropQMeasure na stejnou frontu.



Obr.12 Jednoduchá fronta s náhodným zahazováním

5.5.3 Vícenásobná fronta s náhodným zahazováním

Pokud implementace síťového zařízení požaduje měření vícenásobných front k určení chování zahazovacího algoritmu. V naší ukázce si předvedeme dvě fronty Q_AF1 a Q_AF2 sdílející stejný zdrojový buffer. Ujistíme se zda je společný zdrojový buffer dostatečný k obsloužení provozu AF11. Změříme obě fronty pro určení zahazovacího algoritmu pro provoz AF11 směrovaného do Q_AF1. Vícenásobná fronta s náhodným zahazováním je určena záznamem mQDrop uloženým v dsAlgDropType z dsAlgDropEntry. Dále mohou být použity následující atributy:

- dsAlgDropType — ukazuje typ použitého algoritmu (mQDrop)
- dsAlgDropNext — určuje další funkční cestu dat k ovládání provozu pokud nenastane zahazování datagramů
- dsAlgDropQMeasure — použito k upoutání záznamů v dsMQAlgDropEntry, jeden pro každou frontu měření
- dsAlgDropQThreshold — určuje velikost sdíleného bufferu
- dsAlgDropSpecific — pokud je vyžadováno více parametrů k popisu sdíleného bufferu je možno použít tento záznam

V naší ukázce jsou dva následující dsMQAlgDropEntrys záznamy, jeden pro každou měřenou frontu s následujícími atributy:

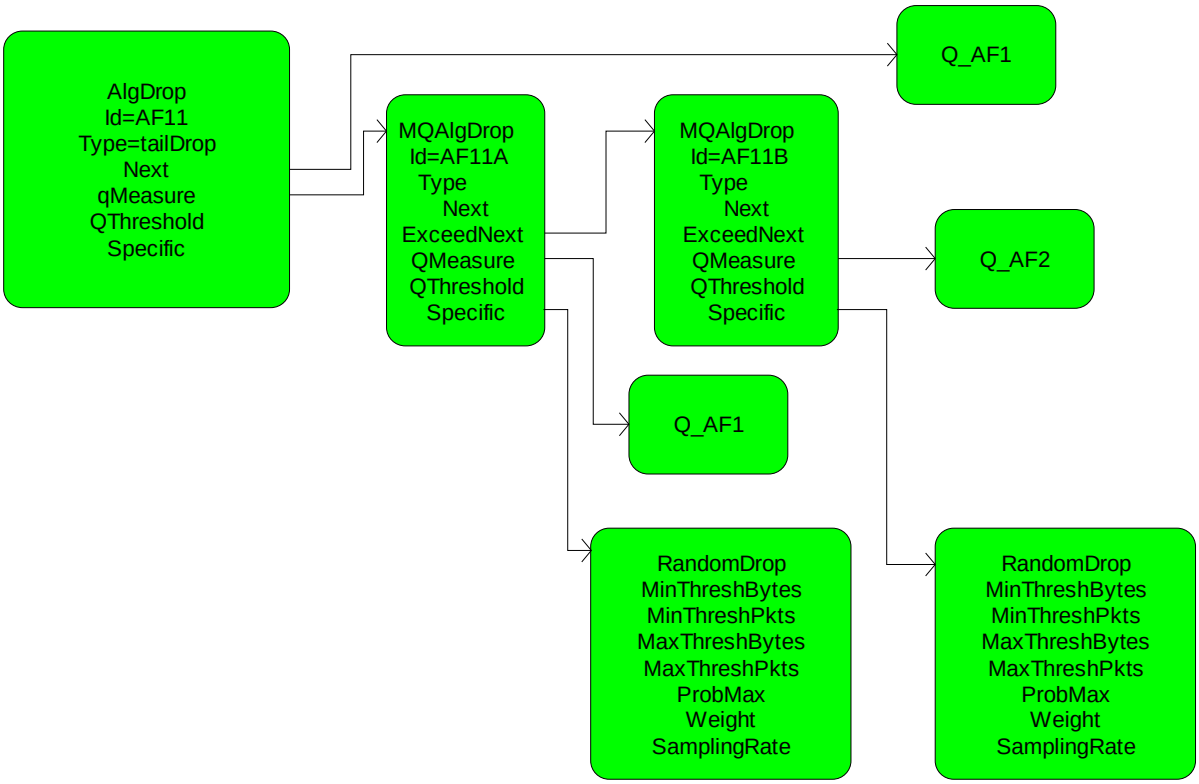
- dsMQAlgDropType — určuje použitý algoritmus (zde random drop)
- dsMQAlgDropQMeasure — ukazuje na frontu, která je měřená (Q_AF1 a Q_AF2 v ukázce)

- dsMQAlgDropNext určuje další funkční cestu dat k ovládání provozu pokud nenastane zahazování datagramů
- dsMQAlgDropExceedNext ukazuje na další frontu, poslední frontu indikuje jako zeroDotzero
- dsAlgDropQThreshold používá se při potřebě jednoduché fronty s náhodným zahazováním
- dsAlgDropSpecific odkazuje na PRID popisující parametry zahazovače při normálním chování (Vnaší ukázce oba tyto záznamy ukazují na dsRandomDropEntrys)

V naší ukázce všechny atributy Next v záznamech v dsAlgDropEntry i vobou dsMQAlgDropEntrys ukazují na Q_AF1.

Z toho nám vyplývá:

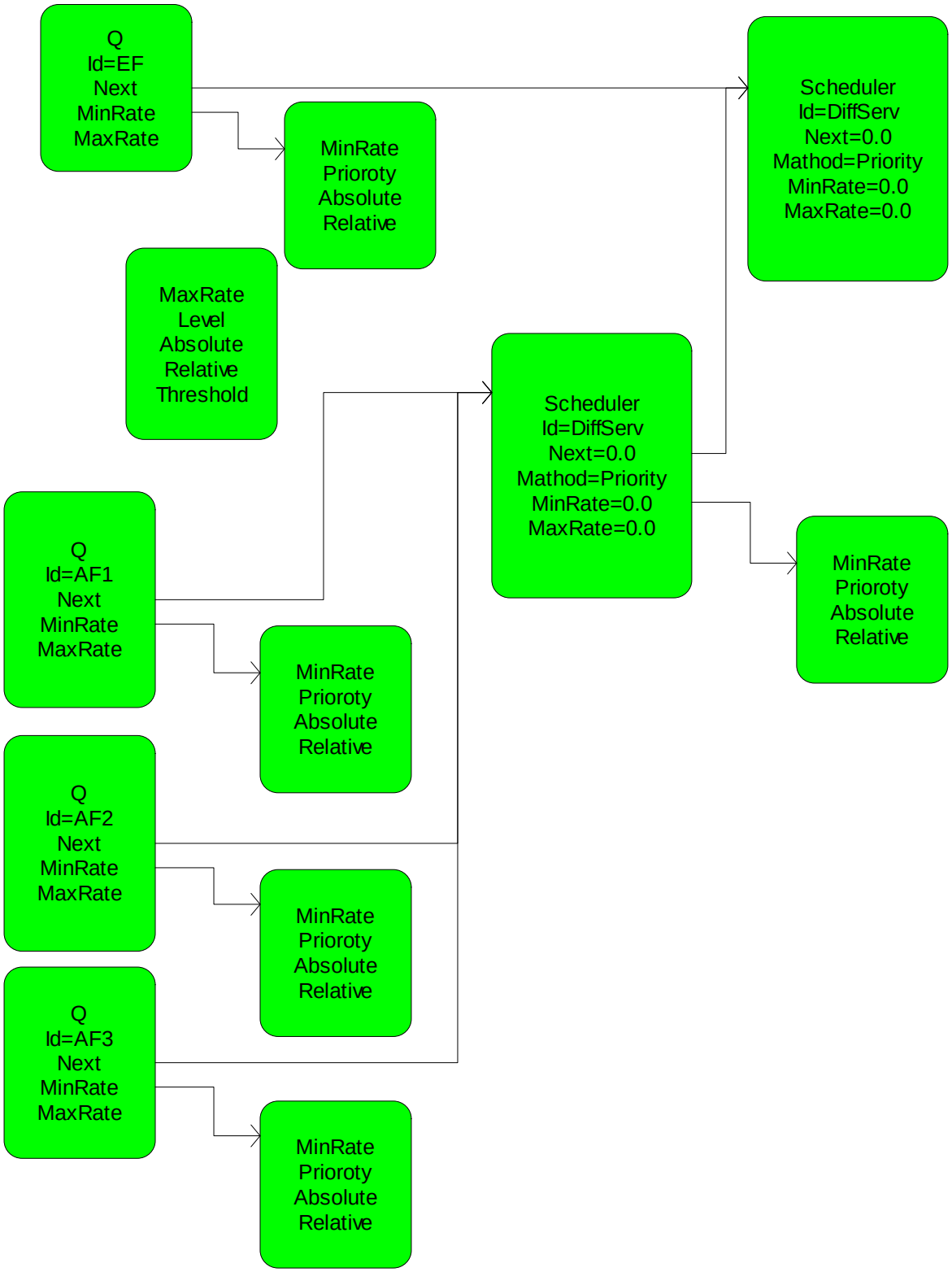
- při nadbytku vyrovnávací paměti (buffer) nemusí být datagram zkontrolovám frontovým zahazovcím mechanizmem, ale může být rovnou poslán na Q_AF1.
- pokud není datagram zahozen v důsledku současných Q_AF1 podmínek, pak je poslán do Q_AF1
- pokud není datagram zahozen v důsledku současných Q_AF2 podmínek, pak je poslán do Q_AF1



Obr.13 Vícenásobná fronta s náhodným zahazováním

5.6 Příklad fronty a plánování

Zde nám cesta mechanizmem Diffserv ze Zahazovače (dropper) navazuje na fronty a plánovač (Scheduler). Všimněme si, že formovač (shaper) používá frontu, zatímco plánovač (scheduler) používá parametr MaxRate. V naší ukázce systém ovládá frontu provozu s toky EF, AF1, AF2 a AF3. Provoz z AF11, AF12 a AF13 směřují do fronty AF1 (analogicky to bude pro provoz AF2x a AF3x).Fronty AF1,AF2 a AF3 obsluhuje pomocí metody Weighed Round Robin přímo AF plánovač (Scheduler). Obsluha všech front pomocí AF plánovače je založena na parametru minimum rate každé fronty. AF a EF provoz je obsloužen v Diffserv plánovači (Scheduler) metodou Strict Priority.Tato metoda je založena na jejich parametru Priority. Diffserv plánovač (Scheduler) používá oba parametry (maximum rate i prioritu) pro obsluhu EF fronty. [6]



Obr.14 Příklad fronty a plánovače

V jeho Next parametru je zeroDotZero z čehož plyne, že Diffserv plánovač (Scheduler) je poslední funkční prvek kterým data musí projít na své cestě tímto mechanizmem.

6 Porovnání architektur systémových databází MIB

V této kapitole se podíváme na srovnání dvou nejpoužívanějších struktur MIB databází. Jedná se o část struktury MIB databáze organizace IETF kterou porovnáme s její jinou částí a to konkrétně z větve Privat, která je používaná soukromími firmami. V poslední době je jednoznačná jednička mezi síťovými řešeními, zařízeními a technologiemi firma Cisco Systems, Inc. proto použijeme právě její část struktury MIB databáze pro porovnání se strukturou MIB databáze organizace IETF. Nejdříve se seznámíme se strukturou MIB databáze organizace IETF po podrobném prozkoumání se budeme věnovat MIB databázi firmy Cisco Systems, Inc. Po poznání obou struktur je porovnáme a vyhodnotíme oba systémy. Identifikujeme společné vlastnosti, rozdíly a hlavní účel, ke kterému se dané databáze využívají. Obě databáze mají stejný začátek a liší se až po několika obecných větveních. Kvůli této vlastnosti musíme vybrat z obou struktur MIB databáze ty nejvhodnější části používané k podobným účelům. Nabízí se nám jednoznačně srovnání jednotlivých management objektů. Jsou to objekt management neboli mgt s OID: 1.3.6.1.2 u organizace IETF a objekt ciscoMgmt sOID: 1.3.6.1.4.1.9.9. Bohužel tak rozsáhlé téma nemůžeme prozkoumat podrobně. Zde si určíme pouze základní objekty a podrobněji se zanoříme pouze do určitých větví nejvíce souvisejících s naším tématem. Pro názornost si zde uvedeme i celé cesty k těmto objektům:

Cesta k objektu ve větvi IETF organizace:

iso(1).identified-organization(3).dod(6).internet(1).mgmt(2)

Cesta k objektu ve větvi Cisco Systems, Inc.:

iso (1).org (3).dod (6).internet (1).private (4).enterprises (1).cisco (9).ciscoMgmt (9)

6.1 Struktura MIB databáze organizace IETF

Strukturu objektů MIB databází od organizace IETF budeme zkoumat od objektu

mgt s OID: 1.3.6.1.2,

což je i, shodou okolností, první objekt rozlišující tuto cestu od cesty k objektu ciscoMgmt s OID: 1.3.6.1.4.1.9.9. Tento objekt identifikuje objekty definované a schválené IAB (Internet Architecture Board) což je komise organizace IETF. Obsahuje následující objekty

reserved(0), mib-2(1), pib(2), http(9), 27(27), ianamallocMIB(102), ipMIB(148). Pro naše potřeby je nejzajímavější objekt

pib s OID: 1.3.6.1.2.2

Tento objekt obsahuje Policy Information Base (PIB), kterou jsme podrobněji již popsali.

Zde máme na výběr zpěti možností frwkBasePibClasses(1), frwkDeviceCapClasses(2), frwkClassifierClasses(3), frwkMarkerClasses(4), frwkBasePibConformance(5).Z těchto objektů vybereme

dsPolicyPib s OID: 1.3.6.1.2.2.4

Tento PIB modul obsahuje množinu tříd popisujících QoS (Quality of Service) pro metodu Diffserv. A jak můžeme vidět, dostali jsme se MIB databází až k našemu hlavnímu tématu. Principy popsané v dřívějších kapitolách jsou zde prakticky uplatňované. Všechny modely, nakreslené dříve pro větší názornost, jsou reálně používané právě v této části MIB databáze resp. jsou konkrétní požadavky na kvalitu služeb konfigurovány na konkrétních zařízeních v konkrétní oblasti vreálné síti. Mámenásledující podskupinu dsCapabilityClasses(1), dsPolicyClasses(2), dsPolicyPibConformance(3) ze které je nejdůležitější

dsPolicyClasses s OID: 1.3.6.1.2.2.4.2

Zde máme následující třídy nebo objekty chcete-li dsDataPathTable(1), dsClfrTable(2), dsClfrElementTable(3), dsMeterTable(4), dsTBParamTable(5), dsActionTable(6), dsDscpMarkActTable(7), dsAlgDropTable(8), dsMQAlgDropTable(9), dsRandomDropTable(10), dsQTable(11), dsSchedulerTable(12), dsMinRateTable(13), dsMaxRateTable(14). Pokud si názvy těchto objektů prohlédneme pozorně, jistě nám něco připomínají. Jako příklad si vezmeme dsMeterTable(4). Meter bylo nám dobře známé Měřidlo. Podobných analogií zde jistě najdeme více. Můžeme začít podrobnější zkoumaní jednotlivých částí, protože jsme na požadovaném místě. Jednotlivé objekty z této skupiny si rozdělíme pro větší přehlednost do kapitol.

dsDataPathTable s OID: 1.3.6.1.2.2.4.2.1

Tato tabulka určuje počátek přenosové cesty přes dané síťové zařízení pro jednotlivé rozhraní a směry dat. Každý záznam určuje první funkční elementy v průběhu cesty zpracování toku dat pro jednotlivé cesty zpracování dat. Tento objekt potřebuje pro každé nastavení rozhraní dva záznamy. Záznam pro příchozí data a záznam pro odchozí data jednotlivých rozhraní. Jednotlivé záznamy definuje následující objekt

dsDataPathEntry s OID: 1.3.6.1.2.2.4.2.1.1

,který navazuje na objekt předcházející. Výborně to vidíme z OID vyjádření. V tomto objektu každý záznam ukazuje jednotlivé konkrétní cesty přes síťové zařízení resp. přes jeho jednotlivá rozhraní fyzická nebo i logická. První funkční element cesty zpracování dat sloužící k ovládání toku dat je objekt

sDataPathStart s OID: 1.3.6.1.2.2.4.2.1.1.5

Tento objekt vybírá první funkční element cesty zpracování dat pro tuto konkrétní cestu zpracování dat pomocí mechanismu diffserv. Může odkazovat na tyto proměnné dsClfrEntry, dsMeterEntry, dsActionEntry, dsAlgDropEntry, dsQEntry.

dsDataPathIfDirection s OID: 1.3.6.1.2.2.4.2.1.1.4

Objekt určuje směr provozu dat tj. příchozí nebo odchozí.

dsDataPathRoles s OID: 1.3.6.1.2.2.4.2.1.1.3

Rozhraní, kterého se tato cesta zpracování dat týká, musí určit podmínky jednotlivých funkcí. K tomu musí existovat záznamy v frwkIfRoleComboTable a dsDataPathCapSetName

dsDataPathCapSetName s OID: 1.3.6.1.2.2.4.2.1.1.2

Objekt nastavující schopnost přiřazení určité cesty zpracování dat. K této vlastnosti musí být nastavena proměnná frwkCapabilitySetTable

dsDataPathPrid s OID: 1.3.6.1.2.2.4.2.1.1.1

Objekt obsahující index, který identifikuje jednotlivé případy různých tříd.

dsClfrTable s OID: 1.3.6.1.2.2.4.2.2

Tabulka dsClfrTable vyjmenovává všechny Diffserv funkční elementy třídiče z cesty zpracování dat, které se v daném síťovém zařízení vyskytují. Aktuální definice třídění jsou detailně popsány v objektu dsClfrElementTable patřící ke každému třídiči. Objekt dsClfrId každého záznamu v tabulce dsClfrTable se používá k uspořádání všech elementů patřících k jednomu třídiči. Na každý třídič se odkazuje pomocí elementu obsahujícího jeho ID.

dsClfrEntry s OID: 1.3.6.1.2.2.4.2.2.1

Tento záznam v tabulce popisuje jednotlivé třídiče. Každý element patřící k nějakému třídiči musí mít atribut roven dsClfrId.

dsClfrPrid s OID: 1.3.6.1.2.2.4.2.2.1.1

Objekt dsClfrPrid obsahuje index identifikující jednotlivé případy různých tříd.

dsClfrId s OID: 1.3.6.1.2.2.4.2.2.1.2

Slouží k identifikaci jednotlivých třídičů. Všechen provoz ovládaný třídičem se musí shodovat nejméně s jedním funkčním elementem uvnitř třídiče.

dsClfrElementTable s OID: 1.3.6.1.2.2.4.2.3

Záznamy v této tabulce slouží jako kotva pro všechny vzory, které mají být tříděny. V podstatě jsou zde definovány jednotlivé filtry, podle kterých se třídí jednotlivé toky dat V každém záznamu této tabulky je také určen následující uvažovaný funkční element mechanismu Diffserv. Parametry třídění jsou definovány záznamy tabulek ukazujícími na objekt dsClfrElementSpecific.

dsClfrElementEntry s OID: 1.3.6.1.2.2.4.2.3.1

Záznamy zde uvedené popisují jednotlivé funkční elementy konkrétního třídiče. Postupně se podíváme na objekty dsClfrElementPrid(1), dsClfrElementClfrId(2), dsClfrElementPrecedence(3), dsClfrElementNext(4), dsClfrElementSpecific(5).

dsClfrElementPrid s OID: 1.3.6.1.2.2.4.2.3.1.1

Index jednoznačně identifikující případy této tabulky.

dsClfrElementClfrId s OID: 1.3.6.1.2.2.4.2.3.1.2

Třídič je složen z jednoho nebo více třídících elementů. Každý element patřící stejnému třídiči používá stejné ID resp. ID třídiče identifikuje který element patří ke kterému třídiči. ID je hodnota atributu dsClfrId v tabulce dsClfrEntry.

dsClfrElementPrecedence s OID: 1.3.6.1.2.2.4.2.3.1.3

Pro prioritu třídění platí pravidlo: Vyšší číslo představuje element třídiče s vyšší prioritou. Elementy se stejnou prioritou jsou zpracovávány souběžně a jednotlivé části se nesmí překrývat resp. kolidovat. Elementy třídiče s různou prioritou se překrývat (kolidovat) ve filtru mohou. Element třídiče s nejvyšší prioritou je zpracováván jako první. Na rozhraní v příchozím směru musí být

vždy aplikován jeden nebo více filtrů, který se musí rovnat všem možným příchozím vzorům, přítomným v příchozích paketech. Na rozhraní v odchozím směru žádné požadavky nejsou.

dsClfrElementNext s OID: 1.3.6.1.2.2.4.2.3.1.4

Tento objekt vybírá další funkční element cesty zpracování dat mechanizmu Diffserv. Hodnota zeroDotZero značí konec zpracování dat pomocí Diffserv mechanizmu. Ostatní hodnoty jsou dsClfrEntry, dsMeterEntry, dsActionEntry, dsAlgDropEntry, dsQEntry a již zmíněná DEFVAL (zeroDotZero).

dsClfrElementSpecific s OID: 1.3.6.1.2.2.4.2.3.1.5

Ukazatel na platný záznam v jiné tabulce, která popisuje vhodný třídící filtr (např. záznam v (frwkIpFilterTable (Framework PIB)). Hodnotě zeroDotZero je rovno vše co není rovno a zpracováno jiným třídícím elementem. V každém třídiči může být pouze jeden takový záznam.

dsMeterTable sOID: 1.3.6.1.2.2.4.2.4

Tato tabulka udává různá měřidla používaná systémem ke kontrole toku dat. Jestli bude tok dat měřen a kterým měřidlem určuje záznam tabulky s názvem dsMeterSpecific.

dsMeterEntry s OID: 1.3.6.1.2.2.4.2.4.1

Záznam v tabulce Měřidla popisuje jednotlivá přizpůsobení jednotlivým úrovním Měřidla.

dsMeterPrid sOID: 1.3.6.1.2.2.4.2.4.1.1

Index jednoznačně identifikující případy této tabulky.

dsMeterSucceedNext s OID: 1.3.6.1.2.2.4.2.4.1.2

Pro stejný typ toku dat se při příštím zpracování mechanismem Diffserv vybere stejný funkční element tohoto mechanismu. Hodnota zeroDotZero v této proměnné znamená, že není potřeba další zpracování pomocí mechanizmu Diffserv. Jiné voby mohou odkazovat na jednu z následujících voleb dsClfrEntry, dsMeterEntry, dsActionEntry, dsAlgDropEntry, dsQEntry a již zmíněná hodnota DEFVAL tj. zeroDotZero.

dsMeterFailNext s OID: 1.3.6.1.2.2.4.2.4.1.3

Pokud se typ toku dat neshoduje s předchozím, tak se při zpracování mechanismem Diffserv vybere další funkční element tohoto mechanismu. Hodnota zeroDotZero v této proměnné znamená, že není potřeba další

zpracování pomocí mechanizmu Diffserv. Jiné voby mohou odkazovat na jednu z následujících voleb dsClfrEntry, dsMeterEntry, dsActionEntry, dsAlgDropEntry, dsQEntry a již zmíněná hodnota DEFVAL tj. zeroDotZero.

dsMeterSpecific s OID: 1.3.6.1.2.2.4.2.4.1.4

Objekt dsMeterSpecific udává chování Měřidla odkazem na záznamy obsahující detailní parametry. Např. dsMeterSpecific ukazuje na záznam v tabulce dsTBMeterTable obsahující případ jednoho nastavení Token-Bucket parametrů.

dsTBParamTable s OID: 1.3.6.1.2.2.4.2.5

Tabulka dsTBParamTable vyjadřuje parametry měřidla Token-Bucket. Tento systém se může použít i na kontrolu datového toku. Každé měřidlo má v parametrech nastavenu svou rychlost a velikost. Pro více velikostí nebo rychlostí je potřeba nastavených více parametrů resp. měřidel.

dsTBParamEntry s OID: 1.3.6.1.2.2.4.2.5.1

Každý záznam popisuje jedno nastavení parametrů Měřidla Token-Bucket.

dsTBParamPrid s OID: 1.3.6.1.2.2.4.2.5.1.1

Index jednoznačně identifikující případy této tabulky.

dsTBParamType s OID: 1.3.6.1.2.2.4.2.5.1.2

Token-Bucket parametry přidružené algoritmu Měřidla. Standardní hodnoty pro tyto parametry jsou diffServTBParamSimpleTokenBucket, diffServTBParamAvgRate, diffServTBParamSrTCMBlind, diffServTBParamSrTCMAware, diffServTBParamTrTCMBlind, diffServTBParamTrTCMAware, diffServTBParamTswTCM. Nebo hodnota zeroDotZero pro neznámý parametr.

dsTBParamRate s OID: 1.3.6.1.2.2.4.2.5.1.3

Parametr dsTBParamRate určuje rychlost (v kbps) Token-Bucket měřidla.

dsTBParamBurstSize s OID: 1.3.6.1.2.2.4.2.5.1.4

Určuje maximální velikost jednoho přenášeného burstu

dsTBParamInterval s OID: 1.3.6.1.2.2.4.2.5.1.5

Nastavení časového intervalu používaného měřidlem Token-Bucket.

dsActionTable s OID: 1.3.6.1.2.2.4.2.6

Tato tabulka vyčísluje akce vykonávané s tokem dat. Vícenásobné akce mohou být spojeny (zřetězeny).

dsActionEntry s OID: 1.3.6.1.2.2.4.2.6.1

Každý záznam v této tabulce povoluje popis jedné určité akce aplikované na datový provoz.

dsActionPrid s OID: 1.3.6.1.2.2.4.2.6.1.1

Index jednoznačně identifikující jednotlivé případy této tabulky (třídy).

dsActionNext s OID: 1.3.6.1.2.2.4.2.6.1.2

Objekt dsActionNext vybírá další funkční element cesty zpracování dat k ovládání provozu pro tuto cestu zpracování dat. Hodnota zeroDotZero v tomto objektu znamená, že žádné další akce mechanizmu Diffserv nejsou na této cestě vykonávány.Hodnoty musí ukazovat na tyto objekty dsClfrEntry, dsMeterEntry, dsActionEntry, dsAlgDropEntry, dsQEntry a DEFVAL (zeroDotZero).

dsActionSpecific s OID: 1.3.6.1.2.2.4.2.6.1.3

Objekt dsActionSpecific je ukazatel na případ záznamu dsActionTable poskytující přídavné informace o typu konkrétní akce.

dsDscpMarkActTable s OID: 1.3.6.1.2.2.4.2.7

Tabulka dsDscpMarkActTable se používá k označení nebo přeznačení DSCP pole vIP datagramu.

dsDscpMarkActEntry s OID: 1.3.6.1.2.2.4.2.7.1

Záznam v tomto objektu popisuje jednotlivé DSCP použité pro označení.

dsDscpMarkActPrid s OID:1.3.6.1.2.2.4.2.7.1.1

Index jednoznačně identifikující jednotlivé případy této tabulky (třídy).

dsDscpMarkActDscp s OID:1.3.6.1.2.2.4.2.7.1.2

Konkrétní DSCP použité v té dané akci pro označení nebo přeznačení datového provozu (DSCP pole v IP datagramu).

dsAlgDropTable s OID:1.3.6.1.2.2.4.2.8

Tabulka dsAlgDropTable obsahuje záznamy popisující funkční element cesty zpracování dat a to element který zahazuje datagramy v závslosti na některém algoritmu.

dsAlgDropEntry s OID: 1.3.6.1.2.2.4.2.8.1

Záznamy popisují proces zahazování paketů vzávislosti na některém zalgoritmů. Další údaje k typům algoritmu mohou být nalezeny v dsAlgDropType s podrobnějšími záznamy ukazujícími do objektu dsAlgDropSpecific.

dsAlgDropPrid s OID: 1.3.6.1.2.2.4.2.8.1.1

Index jednoznačně identifikující jednotlivé případy této tabulky (třídy).

dsAlgDropType s OID:1.3.6.1.2.2.4.2.8.1.2

Typ algoritmu použitý zahazovačem.Možné hodnoty v objektu typu algoritmu (dsAlgDropType) jsou other(1), tailDrop(2), headDrop(3), randomDrop(4), alwaysDrop(5),mQDrop(6). Hodnoty tailDrop(2), headDrop(3), nebo alwaysDrop(5) označují algoritmus, který je kompletně specifikovaný v PIB (Policy Information Base). Hodnota other(1) udává vlastnosti zahazovacího algoritmu určeného jiným PIB modulem. Atribut dsAlgDropSpecific ukazuje na připad PRC v tom PIB modulu, který určuje informace nutné k implementaci vlastního zahazovacího algoritmu.Algoritmus tailDrop(2) je popsán následovně:

Objekt dsAlgDropQThreshold reprezentuje šířku fronty, ukazuje na objekt dsAlgDropQMeasure ve kterém jsou nově příchozí datagramy určené k zahození.

Algoritmus headDrop(3):

Pokud datagram přijde v době, když šířka fronty odkazovaná objektem dsAlgDropQMeasure na objekt dsAlgDropQThreshold, tak jsou datagramy v čele fronty zahozeny, aby vytvořily místo pro nové datagramy zařazené na konci fronty.

Algoritmus randomDrop(4):

Na příchozí datagramy je aplikován algoritmus náhodného zahazování datagramů. Jednotlivá specifika se liší podle výrobce. V tomto algoritmu se objekt dsAlgDropSpecific odkazuje na objekt dsRandomDropEntry popisující

tento algoritmus. V něm objekt dsAlgQThreshold určuje maximální velikost fronty. Přídavné parametry popisuje tabulka dsRandomDropTable.
Algoritmus alwaysDrop(5):
Zahazuje datagramy vždy. Funguje v případě, že nejsou ostatní algoritmy nakonfigurovány resp. jsou nastaveny na hodnoty zeroDotZero.
Algoritmus mQDrop(6):
Slouží v případech, kdy je měřeno více front v zahazovacím algoritmu. V tomto algoritmu je odkaz z dsAlgDropQMeasure na dsMQAlgDropEntry.

dsAlgDropNext s OID:1.3.6.1.2.2.4.2.8.1.3

Objekt dsAlgDropNext vybírá další funkční element cesty zpracování dat mechanizmu Diffserv. Hodnota zeroDotZero v tomto objektu ukončuje zpracování dat pomocí mechanizmu Diffserv. Ostatní hodnoty musí ukazovat na platný případ jednoho z objektů dsClfrEntry, dsMeterEntry, dsActionEntry, dsAlgDropEntry, dsQEntry. Pokud je v objektu dsAlgDropType hodnota alwaysDrop(5) je objekt dsAlgDropNext úplně ignorován.

dsAlgDropQMeasure s OID: 1.3.6.1.2.2.4.2.8.1.4

Objekt dsAlgDropQMeasure ukazuje na PRI indikující, zda je zahazovací algoritmus monitorován, když se rozhoduje, jestli zahodí datagram. Pro objekt alwaysDrop(5) je hodnota nastavena na zeroDotZero. Pro objekty tailDrop(2), headDrop(3), randomDrop(4), odkazuje na záznam v dsQTable.

dsAlgDropQThreshold s OID:1.3.6.1.2.2.4.2.8.1.5

Objekt dsAlgDropQThreshold určuje maximální práh šířky fronty v bytech generovaný při spouštění zahazovacího algoritmu. V případě konfigurace objektu dsAlgDropType na hodnotu alwaysDrop(5) je atribut dsAlgDropQThreshold ignorován. Pro algoritmy tailDrop(2) a headDrop(3) objekt dsAlgDropQThreshold udává šířku fronty odkazovanou objektem dsAlgDropQMeasure.

dsAlgDropSpecific s OID: 1.3.6.1.2.2.4.2.8.1.6

Objekt dsAlgDropSpecific odkazuje na záznamy tabulky poskytující další podrobnosti k zahazovacímu algoritmu.

dsMQAlgDropTable s OID:1.3.6.1.2.2.4.2.9

Tabulka dsMQAlgDropTable obsahuje záznamy popisující které jsou měřeny zahazovacím algoritmem pro mnohonásobné fronty.

dsMQAlgDropEntry s OID:1.3.6.1.2.2.4.2.9.1

Záznamy zde popisují proces zahazovaní datagramů podle toho, který algoritmus zahození provádí. Každý záznam je použit pro jednu z různých front, které jsou zrovna používány, a zároveň rozšiřuje základní záznam dsAlgDropEntry přidáním atributu dsMQAlgDropExceedNext. Další podrobnosti jsou v objektu dsAlgDropType.

dsMQAlgDropExceedNext s OID:1.3.6.1.2.2.4.2.9.1.1

Objekt dsMQAlgDropExceedNext spojuje vícenásobný objekt sMQAlgDropEntry s objektem mQDrop. Hodnota zeroDotZero znamená poslední článek řetězu určeného objektem dsMQAlgDropEntry.

dsRandomDropTable s OID: 1.3.6.1.2.2.4.2.10

Tabulka dsRandomDropTable popisuje proces náhodného zahazování paketů. Záznamy v této tabulce jsou určeny kodkazování na objekt dsAlgDropSpecific, pokud je objekt dsAlgDropType nastaven na randomDrop(4).

dsRandomDropEntry s OID: 1.3.6.1.2.2.4.2.10.1

Záznamy dsRandomDropEntry popisují náhodné zahazování datagramů. Jednotlivé záznamy si probereme podrobněji na následujících řádcích.

dsRandomDropPrid s OID:1.3.6.1.2.2.4.2.10.1.1

Index jednoznačně identifikující jednotlivé případy této tabulky (třídy).

dsRandomDropMinThreshBytes s OID:1.3.6.1.2.2.4.2.10.1.2

Objekt dsRandomDropMinThreshBytes udává průměrnou šířku fronty za kterou je nenulová pravděpodobnost zahození datového toku měřeného v bytech.

dsRandomDropMinThreshPkts s OID:1.3.6.1.2.2.4.2.10.1.3

Objekt dsRandomDropMinThreshPkts udává průměrnou šířku fronty za kterou je nenulová pravděpodobnost zahození datového toku měřeného v datagramech.

dsRandomDropMaxThreshBytes s OID:1.3.6.1.2.2.4.2.10.1.4

Objekt dsRandomDropMaxThreshBytes měří průměrnou šířku fronty v závislosti na pravděpodobnosti zahození nebo označení datového toku (indikováno objektem dsRandomDropProbMax). Tento údaj se liší od fyzické šířky fronty uložené v dsAlgDropQThreshold.

dsRandomDropMaxThreshPkts s OID:1.3.6.1.2.2.4.2.10.1.5

Objekt dsRandomDropMaxThreshPkts měří průměrnou šířku fronty za kterou bude datový tok pravděpodobně zahozen nebo označen objektem dsRandomDropProbMax. Tento údaj se také liší od fyzické šířky fronty uložené v dsAlgDropQThreshold.

dsRandomDropProbMax s OID:1.3.6.1.2.2.4.2.10.1.6

Objekt dsRandomDropProbMax vyjadřuje nejhorší možnou pravděpodobnost náhodného zahození vyjádřenou v zahození na tisíc datagramů. Např. pokud je 100% pravděpodobnost zahození datagramů má tento objekt hodnotu 1000 (100% z tisíce datagramů). Takže pokud je zahozeno v nejhorším případě jedno procento z datagramů, pak je hodnota 10 (1% z tisíce datagramů).

dsRandomDropWeight s OID:1.3.6.1.2.2.4.2.10.1.7

Objekt dsRandomDropWeight udává, jak byla v minulosti zatížena funkce Exponentially Weighted Moving Average, která počítá průměrnou šířku fronty dat. Porovnání nového vzorku používá hodnoty z objektů dsRandomDropWeight/MaxValue, které nám dávají potřebný koeficient. Pro porovnání staršího vzorku se používají hodnoty z (MaxValue - dsRandomDropWeight)/MaxValue, kde hodnota MaxValue je reportována pomocí PEP.

dsRandomDropSamplingRate s OID:1.3.6.1.2.2.4.2.10.1.8

Objekt dsRandomDropSamplingRate obsahuje hodnotu, kolikrát za sekundu je pomocí výpočtu Queue Average Calculation vzorkována fronta.

dsQTable s OID:1.3.6.1.2.2.4.2.11

Tabulka dsQTable udává jednotlivé fronty.

dsQEntry s OID:1.3.6.1.2.2.4.2.11.1

Tyto záznamy popisují jednotlivé fronty jako funkční elementy cesty zpracování dat.

dsQPrid s OID:1.3.6.1.2.2.4.2.11.1.1

Index jednoznačně identifikující jednotlivé případy této tabulky (třídy).

dsQNext s OID:1.3.6.1.2.2.4.2.11.1.2

Objekt dsQNext vybírá další plánovač (scheduler) mechanizmu Diffserv. Musí odkazovat na záznam dsSchedulerEntry. Hodnota zeroDotZero v tomto záznamu určuje nedokončený případ objektu dsQEntry a v tomto případě nemá objekt dsQNext na mechanizmus Diffserv žádný vliv.

dsQMinRate s OID:1.3.6.1.2.2.4.2.11.1.3

Objekt dsQMinRate ukazuje na záznam v plánovači v tabulce dsMinRateTable odkazující na objekt dsQNext používaný k obslužení této fronty. Pokud je hodnota objektu dsQMinRate zeroDotZero, pak se použije minimální rychlost a priorita není určena.

dsQMaxRate s OID: 1.3.6.1.2.2.4.2.11.1.4

Objekt dsQMaxRate analogicky ukazuje na záznam v plánovači v tabulce dsMaxRateTable, která se odkazuje na objekt dsQNext používaný k obsloužení této fronty. Při hodnotě zeroDotZero se použije maximální rychlost daného rozhraní. Pokud je hodnota jiná, musí se nastavit priorita se kterou bude fronta dat obsloužena.

dsSchedulerTable s OID:1.3.6.1.2.2.4.2.12

Tabulka dsSchedulerTable uchovává jednotlivé plánovače datagramů. Pro cestu zpracování dat mohou být použity vícenásobné plánovací algoritmy, kde každý z těchto algoritmů je popsán v objektu dsSchedulerEntry.

dsSchedulerEntry s OID:1.3.6.1.2.2.4.2.12.1

Záznamy v objektu dsSchedulerEntry popisují jednotlivé případy plánovacího algoritmu.

dsSchedulerPrid s OID:1.3.6.1.2.2.4.2.12.1.1

Index jednoznačně identifikující jednotlivé případy této tabulky (třídy).

dsSchedulerNext s OID:1.3.6.1.2.2.4.2.12.1.2

Objekt dsSchedulerNext vybírá další funkční element cesty zpracování dat mechanizmu Diffserv. Tento atribut má normálně hodnotu zeroDotZero značící, že na toku dat na této cestě zpracování dat není potřeba další zpracování mechanizmem Diffsev. Pokud je zde uvedená hodnota jiná, musí odkazovat na platnou hodnotu v jednom z objektů dsSchedulerEntrydsQEntry, dsClfrEntry, dsMeterEntry, dsActionEntry, dsAlgDropEntry. Objekt

dsSchedulerNext je určen pro tvorbu vícenásobných plánovačů na jednu cestu zpracování dat.

dsSchedulerMethod s OID:1.3.6.1.2.2.4.2.12.1.3

Objekt dsSchedulerMethod obsahuje přímo plánovací algoritmus používaný plánovačem. Obsahuje tyto hodnoty diffServSchedulerPriority, diffServSchedulerWRR, diffServSchedulerWFQ, které jsou definovány v Diffserv MIB. Případné jiné hodnoty musí být definované v některé z dalších PIB.

dsSchedulerMinRate s OID:1.3.6.1.2.2.4.2.12.1.4

Objekt dsSchedulerMinRate určuje záznam v tabulce dsMinRateTable indikující prioritu nebo minimální výstupní rychlost z konkrétního plánovače. Používá se pouze v případě, když se jedná vícenásobný plánovač. Není-li předepsaná minimální rychlost nebo priorita nastavíme hodnotu zeroDotZero.

dsSchedulerMaxRate s OID:1.3.6.1.2.2.4.2.12.1.5

Objekt dsSchedulerMaxRate ukazuje na záznam v tabulce dsMaxRateTable, který udává maximální výstupní rychlost z plánovače. Tento atribut je použit jenom u vícenásobného plánovače. Není-li předepsaná maximální rychlost nastavíme hodnotu zeroDotZero.

dsMinRateTable s OID:1.3.6.1.2.2.4.2.13

Tabulka dsMinRateTable vyjadřuje individuální nastavení parametrů plánovače.

dsMinRateEntry s OID:1.3.6.1.2.2.4.2.13.1

Objekt dsMinRateEntry popisuje nastavení jednotlivých parametrů plánovače.

dsMinRatePrid s OID:1.3.6.1.2.2.4.2.13.1.1

Index jednoznačně identifikující jednotlivé případy této tabulky (třídy).

dsMinRatePriority s OID:1.3.6.1.2.2.4.2.13.1.2

Objekt dsMinRatePriority určuje prioritu přidružených plánovačů. Datový tok s hodnotou vyšší priority bude obsloužen před datovým tokem s nižší prioritou.

dsMinRateAbsolute s OID:1.3.6.1.2.2.4.2.13.1.3

Objekt dsMinRateAbsolute udává minimální absolutní rychlost v kilobitech za sekundu, která má být přidělena určité frontě. Pokud je hodnota nula , pak není zaručena žádná minimální rychlost.

dsMinRateRelative s OID:1.3.6.1.2.2.4.2.13.1.4

Objekt dsMinRateRelative určuje minimální rychlost, kterou může plánovač přidělit určité frontě. Jedná se o relativní rychlost vzhledem k rychlosti daného rozhraní. Rychlost rozhraní je dána objekty ifSpeed a ifHighSpeed. V případě hodnoty nula není minimální rychlost garantována. V případě nenulové hodnoty je pro tuto frontu určena minimálně rychlost touto hodnotou definovaná.

dsMaxRateTable sOID:1.3.6.1.2.2.4.2.14

Tabulka dsMaxRateTable vyjadřuje individuální nastavení parametrů plánovače.

dsMaxRateEntry s OID:1.3.6.1.2.2.4.2.14.1

Objekt dsMaxRateEntry popisuje nastavení jednotlivých parametrů plánovače.

dsMaxRatePrid sOID:1.3.6.1.2.2.4.2.14.1.1

Index jednoznačně identifikující jednotlivé případy této tabulky (třídy).

dsMaxRateId s OID:1.3.6.1.2.2.4.2.14.1.2

Objekt dsMaxRateId se používá jako identifikátor ke znázornění vícerychlostního formovače (shaper) společně s objektem dsMaxRateLevel. Tento atribut sdružuje všechny atributy rychlosti z vícerychlostního formovače. Každý záznam dsMaxRateEntry z vícerychlostního formovače musí mít v tomto atributu stejnou hodnotu. Rozdílné rychlosti vícerychlostního formovače jsou určeny objektem dsMaxRateLevel. Pro jednorychlostní formovač je v tomto objektu hodnota nula.

dsMaxRateLevel s OID:1.3.6.1.2.2.4.2.14.1.3

Objekt dsMaxRateLevel je index určující, která úroveň vícerychlostního formovače je zadanými parametry určena. Vícerychlostní formovač má více úrovní rychlosti. Závazná rychlost je úroveň jedna, nejvyšší rychlost je určena nejvyšší nakonfigurovanou úrovní. Ostatní rychlosti jsou mezi těmito dvěma úrovněmi. Pro jednorychlostní formovač se používá hodnota jedna.

dsMaxRateAbsolute s OID:1.3.6.1.2.2.4.2.14.1.4

Objekt dsMaxRateAbsolute udává maximální rychlost v kilobitech za sekundu, která má být přidělená určité frontě. Pokud je nastavená hodnota nula není zaručena žádná maximální hodnota.

dsMaxRateRelative s OID:1.3.6.1.2.2.4.2.14.1.5

Objekt dsMaxRateRelative určuje maximální rychlost, kterou může plánovač přidělit určité frontě. Jedná se o relativní rychlost vzhledem k maximální rychlosti daného rozhraní. Rychlost rozhraní je dána objekty ifSpeed a ifHighSpeed. Pokud je nastavena hodnota nula, pak není maximální rychlostní limit nastaven.

dsMaxRateThreshold s OID:1.3.6.1.2.2.4.2.14.1.6

Objekt dsMaxRateThreshold udává počet bytů šířky fronty o kterou se bude velikost a tím i rychlost vícerychlostního plánovače zvyšovat u dalšího rychlostního výstupu. [6]

6.2 Struktura MIB databáze společnosti Cisco

Strukturu objektů MIB databází od společnosti budeme zkoumat od objektu

ciscoMgmt s OID:1.3.6.1.4.1.9.9

Objekt ciscoMgmt je hlavní větví pro nový vývoj MIB databáze firmy Cisco.

ciscoCBQosMIB s OID:1.3.6.1.4.1.9.9.166

Tento objekt je základ Cisco Class-Based QoS MIB databáze. Je zde umožněn přístup pro čtení konfigurace QoS (Quality of Service) a různé statistické informace k síťovým zařízením firmy Cisco. Informace o konfiguraci jsou dostupné v různých objektech a to ClassMap, PolicyMap, Match Statements a konfigurační parametry Feature Actions.

Match Statement - specifikuje srovnávací kritéria pro třídění datagramů.

ClassMap - uživatelem definované třídy obsahující jeden nebo více srovnávacích údajů používaných k třídění datagramů do různých kategorií.

Feature Action – obsahuje vlastnosti QoS, formování datového toku, řazení do front, označování datagramů.

PolicyMap – uživatelem definované postupy, které sdružují všechny uživatelem definované QoS akce do jednotlivých tříd datových toků.

Service policy – je výše uvedená PolicyMap připojená k logickému rozhraní a je jednoznačně definována indexem cbQosPolicyIndex (tento index je většinou shodný s cbQosObjectsIndex)

Logické rozhraní – v této kapitole je tímto pojmem myšleno hlavní rozhraní, pod-rozhraní, Frame Relay DLCI nebo ATM VC.

S každým QoS objektem jsou spjaty dva indexy. Index cbQosConfigIndex identifikuje konfiguraci, která se nemění. Index cbQosObjectIndex identifikuje provozní statistiky. Další důležité vlastnosti, které nám usnadní orientaci v MIB objektech, jsou index cbQosPolicyIndex umožňující identifikovat Service Policy a index cbQosConfigIndex identifikující QoS vlastnosti na určitém síťovém zařízení.

CiscoCBQosMIBObjects s OID: 1.3.6.1.4.1.9.9.166.1

Objekt ciscoCBQosMIBObjects obsahuje nejdůležitější objekty pro Cisco QoS služby a proto se na tuto část MIB databáze podíváme podrobněji.

cbQosServicePolicy s OID: 1.3.6.1.4.1.9.9.166.1.1

Obsahuje jednotlivé tabulky s QoS postupy.

cbQosServicePolicyTable s OID: 1.3.6.1.4.1.9.9.166.1.1.1

Tato tabulka popisuje jednotlivé typy rozhraní a k těmto rozhraním připojené PolicyMap

cbQosServicePolicyEntry s OID: 1.3.6.1.4.1.9.9.166.1.1.1.1

Každý záznam této tabulky popisuje připojenou PolicyMap k určitému logickému rozhraní. V závislosti na typu rozhraní jsou některé záznamy vyplněné a jiné se pro určitý typ rozhraní nepoužijí.

cbQosPolicyIndex s OID: 1.3.6.1.4.1.9.9.166.1.1.1.1.1

Libovolný index přidělený systémem pro všechny služby (Service policy) spojené s určitým typem logického rozhraní.

cbQosIfType s OID: 1.3.6.1.4.1.9.9.166.1.1.1.1.2

Popisuje jednotlivá logická rozhraní. Můžeme si zvolit ze šesti možností: 1:mainInterface, 2:subInterface, 3:frDLCI, 4:atmPVC, 5:controlPlane, 6:vlanPort

cbQosPolicyDirection s OID: 1.3.6.1.4.1.9.9.166.1.1.1.1.3

Objekt cbQosPolicyDirection určuje směr datového toku na který je aplikována Service policy. Volíme ze dvou možností1:input, 2:output

cbQosIfIndex s OID: 1.3.6.1.4.1.9.9.166.1.1.1.1.4

Objekt cbQosIfIndex je ifIndex pro rozhraní ke kterému je připojená určitá služba. Tento index je smysluplný pouze pro logické rozhraní, které má SNMP ifIndex.

cbQosFrDLCI s OID: 1.3.6.1.4.1.9.9.166.1.1.1.1.5

V tomto objektu je hodnota DLCI pro FRVC ke kterému je tato služba připojena. Tento objekt slouží pouze pro rozhraní připojené k FrameRelay DLCI.

cbQosAtmVPI s OID: 1.3.6.1.4.1.9.9.166.1.1.1.1.6

Objekt cbQosAtmVPI slouží podobnému účelu jako objekt předchozí. Hodnota zde uložená určuje VPI pro ATMVC. Použije se pokud je logické rozhraní připojené k ATM síti.

cbQosAtmVCI s OID: 1.3.6.1.4.1.9.9.166.1.1.1.1.7

Objekt cbQosAtmVCI určuje VCI připojené ATM sítě. Platí pouze pro rozhraní připojené k ATM síti.

cbQosEntityIndex s OID: 1.3.6.1.4.1.9.9.166.1.1.1.1.8

Objekt cbQosEntityIndex udává index entity ke kterému je daný postup připojený. Hodnota nula je vrácena pokud není připojena žádná entita.

cbQosVlanIndex s OID: 1.3.6.1.4.1.9.9.166.1.1.1.1.9

Tento index určuje jednotlivé VLAN připojené k danému rozhraní. Platí pouze pro rozhraní, kde je nakonfigurováno více VLAN např. trunk-port.

cbQosInterfacePolicy s OID: 1.3.6.1.4.1.9.9.166.1.2

Objekt obsahuje tabulku s popisem pravidel.

cbQosInterfacePolicyTable s OID: 1.3.6.1.4.1.9.9.166.1.2.1

Tato tabulka popisuje pravidla služeb připojených k hlavnímu rozhraní a podrozhraním.

cbQosInterfacePolicyEntry s OID: 1.3.6.1.4.1.9.9.166.1.2.1.1

Záznamy v této tabulce

cbQosIFPolicyIndex s OID: 1.3.6.1.4.1.9.9.166.1.2.1.1.1

Tento index je přiřazen všem pravidlům určité služby a je totožný s objektem cbQosPolicyIndex.

cbQosFrameRelayVCPolicy s OID: 1.3.6.1.4.1.9.9.166.1.3

cbQosFrameRelayPolicyTable sOID: 1.3.6.1.4.1.9.9.166.1.3.1

Tato tabulka popisuje pravidla služby připojené k Frame Relay DLCI.

cbQosFrameRelayPolicyEntry s OID: 1.3.6.1.4.1.9.9.166.1.3.1.1

Pro jednotlivé směry datových provozů se v této tabulce přidělí kombinace indexů. Touto kombinací jsou určena pravidla připojeného Frame Relay.

cbQosFRPolicyIndex s OID: 1.3.6.1.4.1.9.9.166.1.3.1.1.1

Systémem náhodně přidělený index totožný s objektem cbQosPolicyIndex.

cbQosATMPVCPolicy s OID: 1.3.6.1.4.1.9.9.166.1.4

cbQosATMPVCPolicyTable s OID: 1.3.6.1.4.1.9.9.166.1.4.1

Tato tabulka popisuje pravidla připojené sítě ATM PVC.

cbQosATMPVCPolicyEntry s OID: 1.3.6.1.4.1.9.9.166.1.4.1.1

Pro jednotlivé směry datových provozů se v této tabulce přidělí kombinace indexů. Touto kombinací jsou určena pravidla připojeného ATM.

cbQosATMPolicyIndex s OID: 1.3.6.1.4.1.9.9.166.1.4.1.1.1

Systémem náhodně přidělený index totožný s objektem cbQosPolicyIndex.

cbQosObjects s OID: 1.3.6.1.4.1.9.9.166.1.5

cbQosObjectsTable s OID: 1.3.6.1.4.1.9.9.166.1.5.1

Tato tabulka určuje hierarchii QoS objektů. Poskytuje vztahy mezi jednotlivými indexy.

ConfigIndex je základem pro dotazy z konfiguračních tabulek. Důležitý je také vztah mezi párem PolicyIndex/ObjectsIndex a ConfigIndex.

cbQosObjectsEntry s OID: 1.3.6.1.4.1.9.9.166.1.5.1.1

Záznamy zde popisují QoS objekty. Jsou to dříve zmíněné PolicyMap, ClassMap, Match Statements a Feature Action. Každý záznam je indexován několika indexy (cbQosPolicyIndex, cbQosObjectsIndex) generovanými systémem. Tyto indexy dohromady s indexem cbQosParentObjectsIndex pomáhají určit manažerovi hierarchický vztah mezi objekty.

cbQosObjectsIndex s OID: 1.3.6.1.4.1.9.9.166.1.5.1.1.1

Náhodný index generovaný systémem pro konkrétní případy záznamů v tabulce cbQosObjectsTable.

cbQosConfigIndex s OID: 1.3.6.1.4.1.9.9.166.1.5.1.1.2

Náhodný konfigurační index přidělený systémem přidělený objektu nezávisle na konkrétním případě. Každý objekt se stejnou konfigurací sdílí ten stejný index

cbQosObjectsType s OID: 1.3.6.1.4.1.9.9.166.1.5.1.1.3

Objekt cbQosObjectsType udává typ QoS objektu. 1:policymap, 2:classmap, 3:matchStatement, 4:queueing, 5:randomDetect, 6:trafficShaping, 7:police, 8:set, 9:compression, 10:ipslaMeasure

cbQosParentObjectsIndex s OID: 1.3.6.1.4.1.9.9.166.1.5.1.1.4

Tento objekt udává index případu nadřízeného objektu.

cbQosPolicyMapCfg s OID: 1.3.6.1.4.1.9.9.166.1.6

cbQosPolicyMapCfgTable s OID: 1.3.6.1.4.1.9.9.166.1.6.1

Tato tabulka určuje konfigurační informace PolicyMap.

cbQosPolicyMapCfgEntry s OID: 1.3.6.1.4.1.9.9.166.1.6.1.1

Každý záznam v této tabulce určuje informace ke konkrétní PolicyMap. Obsahuje jméno a popis. Obsahuje tedy pouze konfigurační informace, takže je určena indexem cbQosConfigIndex.

cbQosPolicyMapName s OID: 1.3.6.1.4.1.9.9.166.1.6.1.1.1

Objekt cbQosPolicyMapName udává jméno určité PolicyMap.

cbQosPolicyMapDesc s OID: 1.3.6.1.4.1.9.9.166.1.6.1.1.2

Objekt cbQosPolicyMapDesc udává popis určité PolicyMap.

cbQosClassMapCfg s OID: 1.3.6.1.4.1.9.9.166.1.7

cbQosCMCfgTable s OID: 1.3.6.1.4.1.9.9.166.1.7.1

Tato tabulka určuje informace ke konfiguraci ClassMap.

cbQosCMCfgEntry s OID: 1.3.6.1.4.1.9.9.166.1.7.1.1

Každý záznam v této tabulce určuje informace ke konkrétní ClassMap. Obsahuje jméno a popis. Obsahuje tedy pouze konfigurační informace, takže je určena indexem cbQosConfigIndex.

cbQosCMName s OID: 1.3.6.1.4.1.9.9.166.1.7.1.1.1

Objekt cbQosCMName udává jméno určité ClassMap.

cbQosCMDesc s OID: 1.3.6.1.4.1.9.9.166.1.7.1.1.2

Objekt cbQosCMDesc udává popis určité ClassMap.

cbQosCMInfo s OID: 1.3.6.1.4.1.9.9.166.1.7.1.1.3

Objekt cbQosCMInfo obsahuje hodnoty srovnání v daných třídách. Tento objekt obsahuje volby 1:none, 2:matchAll, 3:matchAny

cbQosMatchStmtCfg s OID: 1.3.6.1.4.1.9.9.166.1.8

cbQosMatchStmtCfgTable s OID: 1.3.6.1.4.1.9.9.166.1.8.1

Tabulka cbQosMatchStmtCfgTable určuje konfigurační informace objektu MatchStatement.

cbQosMatchStmtCfgEntry s OID: 1.3.6.1.4.1.9.9.166.1.8.1.1

Záznamy této tabulky popisují jednotlivé konkrétní objekty MatchStatement. Neobsahuje statistické informace z čehož plyne označení indexem cbQosConfigIndex. Přesné informace udávají následující objekty.

cbQosMatchStmtName s OID: 1.3.6.1.4.1.9.9.166.1.8.1.1.1

Jméno objektu MatchStatement.

cbQosMatchStmtInfo s OID: 1.3.6.1.4.1.9.9.166.1.8.1.1.2

Objekt cbQosCMInfo obsahuje hodnoty srovnání v daných třídách. Tento objekt obsahuje volby 1:none, 2:matchNot

cbQosQueueingCfg s OID: 1.3.6.1.4.1.9.9.166.1.9

cbQosQueueingCfgTable s OID: 1.3.6.1.4.1.9.9.166.1.9.1

Tato tabulka určuje pravidla řazení fronty datového toku.

cbQosQueueingCfgEntry s OID: 1.3.6.1.4.1.9.9.166.1.9.1.1

Záznamy této tabulky popisují konfigurační hodnoty řazení front. Obsahují jednotku, šířku pásma, prioritu a různá další nastavení.

cbQosQueueingCfgBandwidth s OID: 1.3.6.1.4.1.9.9.166.1.9.1.1.1

Tento objekt obsahuje nakonfigurovanou šířku pásma přidělenou určité třídě datového toku.

cbQosQueueingCfgBandwidthUnits s OID: 1.3.6.1.4.1.9.9.166.1.9.1.1.2

Obsahuje jednotky ve kterých se může počítat šířka pásma. Jedná se o tři možnosti nastavení 1:kbps, 2:percentage, 3:percentageRemaining

cbQosQueueingCfgFlowEnabled s OID: 1.3.6.1.4.1.9.9.166.1.9.1.1.3

Objekt cbQosQueueingCfgFlowEnabled obsahuje dvě možné hodnoty nastavení 1:true
2:false. Určuje zapojení algoritmu flow-based fair-queue.

cbQosQueueingCfgPriorityEnabled s OID: 1.3.6.1.4.1.9.9.166.1.9.1.1.4

Objekt cbQosQueueingCfgPriorityEnabled určuje zda je zapojená priorita pro konkrétní třídu. Obsahuje hodnoty 1:true, 2:false.

cbQosQueueingCfgAggregateQSize s OID: 1.3.6.1.4.1.9.9.166.1.9.1.1.5

Objekt cbQosQueueingCfgAggregateQSize určuje maximální velikost počtu datagramů, které mohou být drženy v jednotlivých frontách, před zahozením těchto datagramů.

cbQosQueueingCfgIndividualQSize s OID: 1.3.6.1.4.1.9.9.166.1.9.1.1.6

Objekt cbQosQueueingCfgIndividualQSize určuje maximální počet datagramů držených v jednotlivých Flow-based fair-queue frontách (spjatých s určitou třídou) před jejich zahozením.

cbQosQueueingCfgDynamicQNumber s OID: 1.3.6.1.4.1.9.9.166.1.9.1.1.7

Objekt cbQosQueueingCfgDynamicQNumber udává podporovaný počet dynamických front při použití algoritmu flow-based fair-queue.

cbQosQueueingCfgPrioBurstSize s OID: 1.3.6.1.4.1.9.9.166.1.9.1.1.8

Objekt cbQosQueueingCfgPrioBurstSize udává počet bytů povolených v jednom burstu. Funguje pouze pokud je zapnutá priorita jednotlivých front.

cbQosQueueingCfgQLimitUnits s OID: 1.3.6.1.4.1.9.9.166.1.9.1.1.9

Objekt cbQosQueueingCfgQLimitUnits udává typ jednotky objektu cbQosQueueingCfgAggregateQLimit. Můžeme si vybrat z následujících možností 1:packets, 2:cells, 3:bytes, 4:ms, 5:us.

cbQosQueueingCfgAggregateQLimit s OID: 1.3.6.1.4.1.9.9.166.1.9.1.1.10

Objekt cbQosQueueingCfgAggregateQLimit určuje maximální povolenou velikost fronty přidruženou nějaké třídě. Pokud velikost této fronty dosáhne tuto hodnotu, datagramy budou zahozeny.

cbQosQueueingCfgAggrQLimitTime s OID: 1.3.6.1.4.1.9.9.166.1.9.1.1.11

Objekt cbQosQueueingCfgAggrQLimitTime určuje maximální povolenou velikost front v určité třídě v milisekundách. Tato hodnota je pro vnitřní použití přepočítána, s použitím šířky pásma, na byty.

cbQosREDCfg s OID: 1.3.6.1.4.1.9.9.166.1.10

cbQosREDCfgTable s OID: 1.3.6.1.4.1.9.9.166.1.10.1

Tabulka cbQosREDCfgTable udává parametry konfigurace algoritmu WRED.

cbQosREDCfgEntry s OID: 1.3.6.1.4.1.9.9.166.1.10.1.1

Záznamy v této tabulce uvádějí informace o konfiguraci algoritmu WRED. Tato tabulka obsahuje konfigurační záznamy. Statistické údaje nejsou přítomné. Každá činnost algoritmu WRED je označena indexem cbQosConfigIndex.

cbQosREDCfgExponWeight s OID: 1.3.6.1.4.1.9.9.166.1.10.1.1.1

Objekt cbQosREDCfgExponWeight je faktor rozkladu pro průměrný výpočet fronty.

cbQosREDCfgMeanQsize S OID: 1.3.6.1.4.1.9.9.166.1.10.1.1.2

Objekt cbQosREDCfgMeanQsize udává průměrnou velikost fronty spočítanou algoritmem WRED.

cbQosREDCfgDscpPrec s OID: 1.3.6.1.4.1.9.9.166.1.10.1.1.3

Objekt cbQosREDCfgDscpPrec obsahuje třídící mechanizmus používaný algoritmem RED. Obsahuje tyto volby 1:precedence, 2:dscp, 3:discardClass, 4:l2Cos, 5:atmClp, 6:mplsExp, 7:redDefault, 8:redUserDefault.

cbQosREDCfgECNEnabled s OID: 1.3.6.1.4.1.9.9.166.1.10.1.1.4

Objekt cbQosREDCfgECNEnabled obsahuje pouze hodnoty true a false o zapnutí tohoto algoritmu signalizujícímu přetížení.

cbQosREDClassCfg s OID: 1.3.6.1.4.1.9.9.166.1.11

cbQosREDClassCfgTable s OID: 1.3.6.1.4.1.9.9.166.1.11.1

Tabulka cbQosREDClassCfgTable obsahuje informace o algoritmu WRED založené na poli IP Precedence.

cbQosREDClassCfgEntry s OID: 1.3.6.1.4.1.9.9.166.1.11.1.1

Záznamy v této tabulce popisují konfiguraci algoritmu WRED. V této tabulce jsou pouze informace o konfiguraci. Jednotlivé záznamy jsou určeny indexem cbQosConfigIndex.

cbQosREDValue s OID: 1.3.6.1.4.1.9.9.166.1.11.1.1.1

Objekt cbQosREDValue udává záznam z IP precedence nebo z IP DSCP pole.

cbQosREDCfgMinThreshold s OID: 1.3.6.1.4.1.9.9.166.1.11.1.1.2

Objekt cbQosREDCfgMinThreshold udává minimální práh počtu datagramů. Pokud průměrná šířka fronty dosáhne tohoto čísla, algoritmus WRED začne zahazovat datagramy určené polem IP Precedence.

cbQosREDCfgMaxThreshold s OID: 1.3.6.1.4.1.9.9.166.1.11.1.1.3

Objekt cbQosREDCfgMaxThreshold udává maximální hranici v počtu datagramů. Pokud průměrná šířka fronty překročí tuto hodnotu, algoritmus WRED zahodí všechny datagramy specifikované polem IP Precedence.

cbQosREDCfgPktDropProb s OID: 1.3.6.1.4.1.9.9.166.1.11.1.1.4

Objekt cbQosREDCfgPktDropProb je průměrná úroveň zlomku datagramů zahozených pokud je šířka průměrné fronty nastavena na MaxDepthThreshold.

cbQosREDClassCfgThresholdUnit s OID: 1.3.6.1.4.1.9.9.166.1.11.1.1.5

Objekt cbQosREDClassCfgThresholdUnit udává typ jednotky měřené algoritmem RED. Možné jsou tyto typy 1:packets, 2:cells, 3:bytes, 4:ms, 5:us.

cbQosREDClassCfgMinThreshold s OID: 1.3.6.1.4.1.9.9.166.1.11.1.1.6

Objekt cbQosREDClassCfgMinThreshold udává minimální prahové číslo. Při dosažení tohoto čísla průměrnou šířkou fronty začíná algoritmus WRED zahazovat datagramy v závislosti na mechanismu RED určeného objektem cbQosREDCfgDscpPrec

cbQosREDClassCfgMaxThreshold s OID: 1.3.6.1.4.1.9.9.166.1.11.1.1.7

Objekt cbQosREDClassCfgMaxThreshold udává maximální prahové číslo. Při dosažení tohoto čísla průměrnou šířkou fronty začíná algoritmus WRED zahazovat datagramy v závislosti na mechanismu RED určeného objektem cbQosREDCfgDscpPrec.

cbQosREDClassCfgMinThresholdTime sOID:
1.3.6.1.4.1.9.9.166.1.11.1.1.8

Objekt cbQosREDClassCfgMinThresholdTime určuje minimální hodnotu algoritmu WRED v milisekundách (pro interní použití přepočítáno na byty s použitím šířky pásma dostupného pro tuto třídu).

cbQosREDClassCfgMaxThresholdTime s OID: 1.3.6.1.4.1.9.9.166.1.11.1.1.9

Objekt cbQosREDClassCfgMaxThresholdTime určuje maximální hodnotu algoritmu WRED v milisekundách (pro interní použití přepočítáno na byty s použitím šířky pásma dostupného pro tuto třídu).

cbQosPoliceCfg s OID: 1.3.6.1.4.1.9.9.166.1.12

cbQosPoliceCfgTable s OID: 1.3.6.1.4.1.9.9.166.1.12.1

Tabulka cbQosPoliceCfgTable určuje konfiguraci objektu Policy Action.

cbQosPoliceCfgEntry s OID: 1.3.6.1.4.1.9.9.166.1.12.1.1

Záznamy v objektu cbQosPoliceCfgEntry jsou pro naše potřeby příliš obsáhlé. Jedná se o podrobný popis konfigurace Policy Action. Tímto výrazem je myšlena konfigurace parametrů jako jsou rychlost, velikost burstu a jiné jednání související s rychlostí datového toku. V této tabulce jsou také záznamy pouze o konfiguraci a ne o statistických údajích proto je indexována pomocí objektu cbQosConfigIndex.

cbQosTSCfg s OID: 1.3.6.1.4.1.9.9.166.1.13

cbQosTSCfgTable s OID: 1.3.6.1.4.1.9.9.166.1.13.1

Tabulka cbQosTSCfgTable obsahuje konfigurační údaje o formování datového toku.

cbQosTSCfgEntry s OID: 1.3.6.1.4.1.9.9.166.1.13.1.1

Všechny záznamy této tabulky popisují informace o konfiguraci formovače datového toku. Parametry pro konfiguraci formovače jsou rychlost, velikost burstu a různé typy formování. K nastavení se používají následující parametry cbQosTSCfgRate (1), cbQosTSCfgBurstSize (2), cbQosTSCfgExtBurstSize (3), cbQosTSCfgAdaptiveEnabled (4), cbQosTSCfgAdaptiveRate (5), cbQosTSCfgLimitType (6), cbQosTSCfgRateType (7), cbQosTSCfgPercentRateValue (8), cbQosTSCfgBurstTime (9), cbQosTSCfgExtBurstTime (10), cbQosTSCfgRate64 (11).

cbQosSetCfg s OID: 1.3.6.1.4.1.9.9.166.1.14

cbQosSetCfgTable s OID: 1.3.6.1.4.1.9.9.166.1.14.1

Tabulka cbQosSetCfgTable určuje konfiguraci značkovače.

> **cbQosSetCfgEntry s OID: 1.3.6.1.4.1.9.9.166.1.14.1.1**
>
> Každý záznam v této tabulce popisuje konfiguraci značkovače. Všechny záznamy jsou určeny objektem cbQosConfigIndex, protože tato tabulka obsahuje pouze konfigurační údaje. Nastavují se tyto parametry cbQosSetCfgFeature (1), cbQosSetCfgIpDSCPValue (2), cbQosSetCfgIpPrecedenceValue (3), cbQosSetCfgQosGroupValue (4), cbQosSetCfgL2CosValue (5), cbQosSetCfgMplsExpValue (6), cbQosSetCfgDiscardClassValue (7), cbQosSetCfgMplsExpTopMostValue (8), cbQosSetCfgSrpPriority (9), cbQosSetCfgFrFecnBecn (10).

Tabulka cbQosSetCfgTable v objektu cbQosSetCfg je poslední objekt ve větvi Cisco MIB databáze , kde můžeme najít podobné vlastnosti jako u větve MIB databáze organizace IETF. V dalších tabulkách jsou pouze sbírané statistické údaje, které už námi probírané principy neobsahují. Některé si uvedeme. Jedná se např. o cbQosClassMapStats (15), cbQosMatchStmtStats (16), cbQosPoliceStats (17), cbQosQueueingStats (18), cbQosTSStats (19), cbQosREDClassStats (20). [11]

6.3 Porovnání vlastností architektur MIB databází společnosti Cisco a organizace IETF

Při fyzickém srovnávání se obě větve zdají téměř totožné. Všechny části se nazývají objekty. Podle místa jim můžeme dávat i jiná pojmenování jako tabulka, třída, atribut, záznam. Hlavní objekt v obecném tvaru architektury IETF je tabulka. Pod tabulkou máme objekt, kterému říkame záznam obsahující konečné objekty s konkrétní konfigurací nebo hodnotou. U společnosti Cisco je situace velmi podobná s tím rozdílem, že tabulky jsou chráněny ještě hierarchicky nadřízeným objektem. Architektura MIB databáze organizace IETF je jednodušší a logičtější. Musí se přízpůsobit širšímu okruhu požadavků a tak nemůže mít speciální vlastnosti těžko použitelné širšímu okruhu populace. Oproti tomu je část Cisco MIB databáze přizpůsobena přímo na míru síťovým zařízením společnosti Cisco. Největší rozdíl je v přístupu k jednotlivým databázím a v účelu ke kterému se využívají. MIB databáze organizace IETF se používá hlavně na konfiguraci a tudíž je zde možnost zápisu, bez kterého

by konfigurace síťových zařízení nebyla možná. Oproti tomu je hlavním účelem Cisco MIB databáze získávat statistiky z datových toků v síťových zařízením. Což vysvětluje i přístup pouze pro čtení do většiny objektů Cisco MIB databáze.

7 Závěr

Cílem této práce bylo poznat složitý mechanizmus Diffserv navržený pro řízení kvality služeb (QoS) v sítích a seznámit se s MIB (Management Information Base) databázi pomocí které je Diffserv mechanizmus ovládán a konfigurován.

V práci jsme se postupně seznámili s protokolem SNMP (Simple Network Management Protocol). Tento asynchronní protokol je založen na jednání manažera a agenta. Manažer je software nainstalovaný na stanici síťového managementu (Network Management Station) a sbírá data ze síťových zařízení, která spravuje. Agent je software nainstalovaný na síťovém zařízení a posílá data do Manažera. Agent si data ukládá do MIB databáze. MIB databáze je složena z objektů, které popisují chování síťového zařízení. Struktura objektů a tím i MIB databáze je, pro lepší přehlednost, stromová.

Probrali jsme ztrátu paketů, zpoždění a kolísání zpoždění, což jsou hlavní parametry QoS. Podrobně jsme si ukázali hlavičku IP datagramu a pole ToS (Type of Service) hlavičky IP datagramu, které bylo použito na mechanizmus Diffserv. Popsali jsme si tři možné techniky při zacházení s datagramy. Byly to Standartní chování (Per Hop Behaviour), Zajištěné předávání (Assured Forwarding), Urychlené předávání (Expedited Forwarding). Probrali jsme podrobně architekturu Differencovaných služeb. Jednotlivé části byli Třídič (Classifier), Měřidlo (Meter), Značkovač (Marker), Formovač (Shaper) a Zahazovač (Dropper).

Jako poslední jsme probrali strukturu PIB (Policy Information Base). Skládá se ze dvou hlavních částí. První je místo prosazování zásad PEP (Policy Enformceement Point) a druhá je místo stanovení zásad PDP (Policy Decision Popint). PEP je umístěn přímo na síťových zařízeních a vykonává akce určené PDP, komunikují pomocí protokolu COPS (Common Open Policy Service).

Ukázali jsme si dvě části struktury MIB databáze související s kvalitou služeb. Jedna větev MIB databáze byla od organizace IETF a druhá větev od společnosti Cisco, která je světovým lídrem v síťových komunikacích. Obě větve fungují na stejných principech. Jsou hierarchické a kromě jména mohou být vyjádřeny i čísly, které jsou odděleny tečkami a nazývají se OID. Při fyzickém srovnávání se obě větve zdají téměř totožné. Všechny části se nazývají objekty. Podle místa jim můžeme dávat i jiná pojmenování jako tabulka, třída, atribut, záznam. Největší rozdíl je mezi větvemi MIB databáze v přístupu k jednotlivým databázím a v účelu ke kterému se využívají. MIB databáze organizace IETF se používá hlavně na konfiguraci a tudíž je zde možnost zápisu, bez kterého by konfigurace síťových zařízení nebyla možná.

Oproti tomu je hlavním účelem Cisco MIB databáze získávat statistiky z datových toků v síťových zařízením. Což vysvětluje i přístup pouze pro čtení do většiny objektů Cisco MIB databáze.

Struktura větve MIB databáze organizace IETF je jednodušší a logičtější než privátní struktura společnosti Cisco. Tento jev souvisí s faktem, že větev organizace IETF je určena pro použití celé společnosti. Naproti tomu struktura společnosti Cisco vychází z mechanizmů a principů organizace IETF, ale celá větev je navržena pro její síťová zařízení přímo na míru, tak aby umožňovala co nejvyšší výkon a velmi široký rozsah služeb. Vrámci boje o zákazníka je společnost nucena nabízet stále větší výkon a kvalitu na komerčním trhu. Oproti tomu organizace IETF nemusí bojovat s tímto tlakem a soustředí se spíš na budování nových koncepcí a přístupů k dané problematice bez ohledu na zisk.

8 Přílohy

8.1 Seznam tabulek

8.2 Seznam obrázků

8.3 Použitá literatura

[0] CASE, J., McCLOGHRIE, K., ROSE, M., WALDBUSSER, S. Management Information Base for version 2 of the Simple Network Management Protocol (SNMPv2) RFC 1450, Internet Engineering Task Force, 1993

[1] K. NICHOLS , S. BLAKE, F. BAKER, D. BLACK, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 HeadersRFC 2474,
Internet Engineering Task Force, 1998

[2] S. BLAKE, D. BLACK, M. CARLSON, E. DAVIES, Z. WANG, W. WEISS, An Architecture for Differentiated Services RFC 2475,
Internet Engineering Task Force, 1998

[3] J. HEINANEN, F. BAKER, W. WEISS, J. WROCLAWSKI, Assured Forwarding PHB Group, RFC 2597,
Internet Engineering Task Force, 1999

[4] B. DAVIE, A. CHARNY, J.C.R. BENNETT, K. BENSON, J.Y. LE BOUDEC, W. COURTNEY, S. DAVARI, V. FIROIU, D. STILIADIS, An Expedited Forwarding PHB (Per-Hop Behavior) RFC 3246
Internet Engineering Task Force, 2002

[5] D. GROSSMAN, New Terminology and Clarifications for Diffserv RFC 3260
Internet Engineering Task Force, 2002

[6]CHAN, K., SAHITA, R., McCLOGHRIE, K. Differentiated Services Quality of Service Policy Information Base, RFC 3317,
Internet Engineering Task Force, 2003

[7] PUŽMANOVÁ R., TCP/IP v kostce, ISBN 80-7232-236-2, Praha 2004

[8] VYTVOŘENO SLUŽBOU ESTRÁNKY.CZ , © 2005 - 2007 eStránky.cz
http://www.internet.estranky.cz/

[9] TOBY J. VELTE, ANTHONY T. VELTE, Síťové technologie Cisco Velký Průvodce, IBSN 80-7226-857-0, 2003

[10] ING. KAROL MOLNÁR, PH.D., Použití MIB databáze pro Diffserv mechanizmus

[11] FUNG, A. Cisco Class-Based QoS MIB, Cisco Systems Inc.,
http://tools.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&mibName=CISCO-CLASS-BASED-QOS-MIB-CAPABILITY, 2001