VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2016 Bc. Marek Důbrava

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

ŘÍZENÍ DATOVÉHO TOKU V ISP SÍTI
DATA FLOW CONTROL IN ISP NETWORK

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Marek Důbrava**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Jaroslav Koton, Ph.D.**
SUPERVISOR

BRNO 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Marek Důbrava ***ID:*** 147421
Ročník: 2 ***Akademický rok:*** 2015/16

NÁZEV TÉMATU:

Řízení datového toku v ISP síti

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte současné možnosti řízení datového provozu a dynamického přidělování sdílených síťových prostředků v rámci přístupových až páteřních sítí. Zaměřte se na principy agregace datového provozu a řešení blížícího se stavu zahlcení až řešení vzniklého stavu zahlcení síťového prvku. Porovnejte vybrané mechanismy řešící uvedené stavy sítě a popište proces jejich činnosti z pohledu efektivity nasazení v datových sítích. Vybrané mechanismy prakticky implementujte v síti a diskutujte jejich efektivitu i s ohledem na topologii sítě.

DOPORUČENÁ LITERATURA:

[1] BRISCOE, B.: Tunnelling of Explicit Congestion Notification, Internet Engineering Task Force, RFC 6040, 2010.

[2] FLOYD, S.: Congestion Control Principles, Networking Group, RFC 2914, 2000

Termín zadání: 1.2.2016 ***Termín odevzdání:*** 25.5.2016

Vedoucí práce: doc. Ing. Jaroslav Koton, Ph.D.
Konzultant diplomové práce:

doc. Ing. Jiří Mišurec, CSc., *předseda oborové rady*

ABSTRAKT

Práce se zaměřuje na řízení datového toku v sítích poskytovatelů připojení k Internetu. Zobrazen je problém agregace a s tím související přetížení síťových prvků. V teoretické části jsou popsány standardizované metody pro řízení datového toku. U vybraných zařízení jsou diskutovány mechanismy na ochranu proti přetížení. Práce zobrazuje měření, do jaké míry se vyskytuje ECN algoritmus v praxi. V práci je teoreticky popsán program HTB, který je doplněn nově popsaným algoritmem. Nový algoritmus je následně implementován do programu a testován na reálné síti ISP.

KLÍČOVÁ SLOVA

ISP, zahlcení sítě, řízení datového toku, agregace, ECN, HTB

ABSTRACT

The thesis focuses on the control of data flow in networks of Internet service providers. The problem of aggregation and related overload network elements are shown. Theoretical section describes the standardized methods for managing data flow. The parameters associated with managing data flow are described for devices selected devices. The thesis displays measurements depicting to which extent the ECN algorithm occurs in practice. In the thesis is theoretically described HTB program, which is complemented by a newly described algorithm. The new algorithm is implemented into the program and tested on a real network ISP.

KEYWORDS

ISP, network congestion, data flow control, aggregation, ECN, HTB

DŮBRAVA, Marek *Řízení datového toku v ISP síti*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 78 s. Vedoucí práce byl doc. Ing. Jaroslav Koton, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Řízení datového toku v ISP síti“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc. Ing. Jaroslavu Kotonovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora(-ky)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Technicka 12, CZ-61600 Brno
Czechia

http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této diplomové práci byl realizovaný v laboratořích podpořených projektem Centrum senzorických, informačních a komunikačních systémů (SIX); registrační číslo CZ.1.05/2.1.00/03.0072, operačního programu Výzkum a vývoj pro inovace.

Brno ....

podpis autora(-ky)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum a vývoj
pro inovace

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Lidé se odjakživa vyznačují potřebou komunikovat s ostatními. S příchodem elektrické energie vznikla možnost komunikovat na delší vzdálenost. Vzhledem k technickému pokroku posledních století se možnost komunikovat pomocí elektrické telekomunikační infrastruktury zpřístupnila všem. Telekomunikační sítě se spojily do velkého celku nazvaného Internet, kde nejsou hranice v přenášení informací. Trendem současnosti jsou služby vyžadující tzv. širokopásmové připojení, například video konference nebo sdílení videí na sociálních sítích. Tyto služby začaly klást důraz na přenosovou rychlost. Velké množství účastníků se svými požadavky vede na nepředstavitelné objemy dat přenášených komunikační sítí. Požadavky na přenos dat prostřednictvím internetového protokolu (TCP/IP) stále narůstají. U pevného připojení k síti Internet každým rokem vzroste spotřeba dat o 23 %, u mobilních sítí je vzestup až 57 % [1].

Poskytovatelé připojení k Internetu (ISP) jsou provozovatelé těchto sítí, všichni společně tvoří celosvětovou síť Internet. Aby byly uspokojeny všechny požadavky všech uživatelů, musí být ISP sítě schopny vypořádat se s velkým objemem dat. Řešením zvyšujících se požadavků je nasazování nových technologií, které představují pro poskytovatele investiční zátěž. Proto jsou stávající sítě vytěžovány do svého maxima a blíží se stavu zahlcení. Pro dobrou výkonnost sítě je však nutné se stavům zahlcení vyhnout.

Práce se zabývá řešením problémů s řízením datového toku v ISP síti. Metody popsané níže se téměř vždy věnují tvarování datového toku. Dělí se podle principu i podle sofistikovanosti svého algoritmu. Jsou zde zobrazeny funkce řízení datového toku z více úhlů pohledů. Například bezdrátové zařízení obsahuje odlišný mechanismus proti přetížení v porovnání s kabelovým směrovačem. Některé z popsaných metod nejsou definovány v normách, ale jedná se o vlastní technologie výrobců. Jedna z metod na řízení datového provozu se nazývá ECN, vyznačuje se dobrými parametry a teprve se začíná nasazovat. Proto je v práci prakticky změřeno, jaký má podíl v reálném provozu.

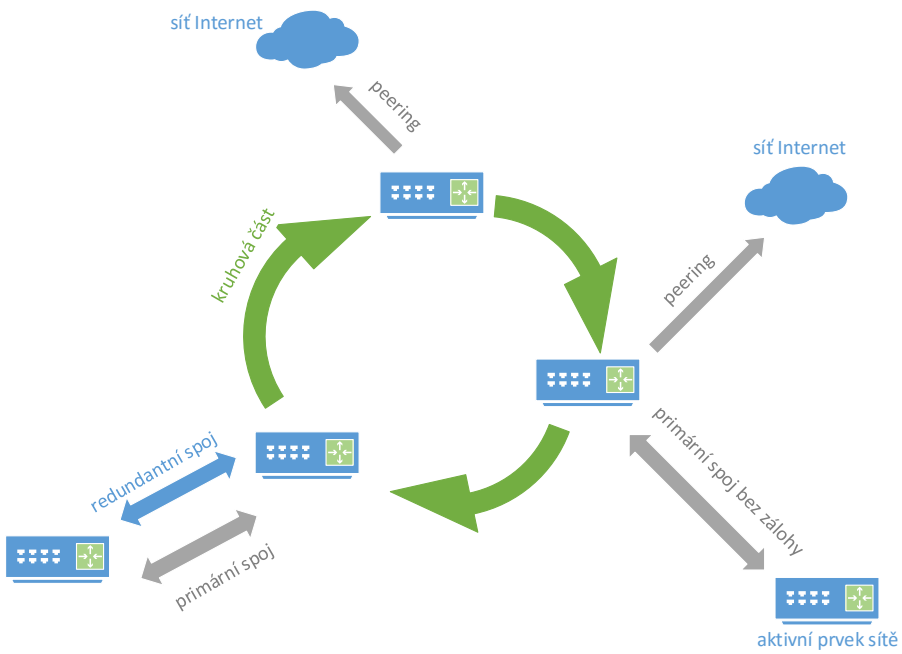
Aby mohl poskytovatel připojení dělit účastníky podle jejich service-level agreement (SLA), musí využívat některé další programy pro tvarování provozu. Příkladem je HTB, který je v práci diskutován a následně rozvinut o nový algoritmus. Oba programy společně dokáží dělit uživatele do skupin, ale i lépe chránit síť před přetížením. Nový algoritmus je implementován do reálné sítě. Následně je provedeno měření s prezentací výsledků.

1 ISP SÍŤ

Internet service provider (ISP) síť je celek vytvořený ze síťových prvků a spojů mezi nimi pod správou jediného subjektu, který má pronájem přenosové kapacity jako podnikatelský záměr. Celá síť je většinou připojena k síti Internet a vlastní autonomní systém s pevnými veřejnými Internet Protocol (IP) adresy. Kompletní síť nemusí být ve vlastnictví subjektu, ale její části mohou být najímány od dalších stran. Rozloha sítě může být na úrovni města, až státu [3], [7].

Typ sítě je velmi rozmanitý, každý poskytovatel si navrhne síť podle vlastní potřeby. V poslední době se stále častěji budují a migrují sítě na paketový přenos Ethernet, protože většina služeb je s tímto protokolem kompatibilní [7], [8]. Právě paketově orientovanou sítí se zabývá tato práce.

Topologie sítě závisí na umístění sítě a na její rozloze. Tvar sítě často tvarují terénní a právní podmínky lokality. Například kabel nebude veden přes hory, když může být vedle kolejí v údolí. Radiové spoje musí být v přímé viditelnosti. Z právního hlediska sítě ovlivňuje především vlastnické právo, kde majitel pozemku může zkomplikovat, nebo prodražit výstavbu sítě [8]. Nejčastější typ topologie je hybridní. Hybridní síť je většinou důsledkem postupného budování [7]. Jeden z možných typů této sítě je zobrazen na obr. 1.1. Základem u našeho příkladu je hlavní kruh, na



Obr. 1.1: Běžná topologie ISP sítě

který se připojují další části sítě. Jak uvedeme v kap. 1.2 důležitým parametrem je dostupnost. Proto je topologie často stavena tak, aby v případě přerušení jednoho spoje nedošlo k nefunkčnosti části sítě. Aktivní prvky mohou být zálohovány kruhem, nebo redundantním spojem [7]. Záloha v praxi znamená vyšší pořizovací náklady, proto není vždy samozřejmostí, obzvlášť u méně důležitých lokalit.

Síť je nejčastěji dělena na přístupovou a páteřní. Přístupová sít je často označována jako poslední míle a jedná se o část sítě nejblíže k zákazníkovi. V případě rodinných domů je takto označen kabel vedoucí z nejbližší ústředny do domu, nebo radiový spoj od vysílače k přijímači instalovaného na domě. Část technické realizace přístupové sítě je vždy v kontaktu s připojeným objektem. Páteřní sítí je označován zbytek sítě, převážně profesionální radiové spoje a optická infrastruktura. [3], [7]

1.1 Síť Durnet.cz

Při vypracování práce je k dispozici síť české společnosti Durnet.cz s.r.o. Síť se nachází ve Zlínském kraji. k 3. 12. 2015 má 321 klientů, převážně domácnosti, firem je méně než 5 %. Síť pokrývá celkem 13 obcí a je realizována celkem 27 síťovými uzly, někdy označovanými jako přístupový bod nebo Point of Presence (POP).

Mapa s polohou bodů a se zakreslenou topologií je zobrazena v příloze A.1. Zobrazené místo na mapě se nachází na jih od města Zlín a na sever od města Uherský Brod. Modré body na mapě zobrazují síťové uzly. Uzly jsou převážně umístěny na cizí nemovitosti a jsou realizovány anténami s radiovými modemy a směrovači provozu. Červené čáry zobrazují radiové trasy propojující uzly sítě. Propojení je realizováno převážně v pásmu 5,4 GHz radiovými zařízeními firmy Ubiquiti Networks. U více náročnějších tras jsou použity frekvence 10,5 GHz a 17 GHz se zařízeními od výrobce Alcoma a.s. Fialové čáry zobrazují propojení sítě Durnet.cz do okolních sítí. Propojení označeno číslem 1 na jihu mapy zobrazuje napojení na telekomunikační společnost Avonet s.r.o. s aktuální rychlostí 200 Mb/s. Propojení na severu označené číslem 2 směřuje na České Radiokomunikace s aktuální rychlostí 100 Mb/s.

1.2 Požadavky na ISP síť

Aby byly přenosy poskytnuté paketově orientovanou sítí kvalitní, musí být splněno několik parametrů. Nejdůležitějšími z nich jsou dostupnost, odezva, ztrátovost paketů a přenosová rychlost. Tyto parametry přímo ovlivňují kvalitu přenášených dat a není možné je zanedbat. Další parametry, které také ovlivní Quality of Experience (QoE) zákazníků jsou například Quality of Service (QoS), rovnoměrné dělení šířky pásma takzvaný Fair User Policy (FUP), proměnlivost zpoždění a další [4], [7].

1.2.1 Požadavky s ohledem na uživatele

Každý zákazník však vyžaduje specifickou kvalitu služeb. Firmy převážně kladou požadavky na garantovanou přenosovou rychlost, aby byly jejich systémy vždy obslouženy. Na dostupnost služby, aby nedošlo k zastavení provozu kvůli výpadku připojení. Další parametry jako dělení šířky pásma firemní klienti nepožadují [8].

Domácnostmi jsou často vyžadovány i parametry QoS a FUP. Uživatel laik potřebuje mít se službami dobré QoE. Pro dobré zkušenosti uživatele musíme zamezit případům, kdy jsou přednostnější služby omezeny službami na pozadí [8]. Například pokud uživatel sleduje televizi a stahuje hru, provoz televize by měl být upřednostněn.

1.3 Problematika řízení datového toku

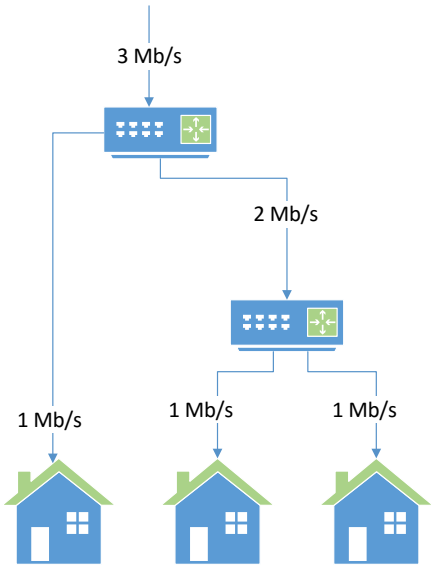
V ISP sítích je nutné použít techniku řízení datového provozu. Primárně z důvodu, aby jedna stanice svou vysokou aktivitou nezahltila celou síť [4]. Dále proto, aby klient nepřekračoval využití sítě nad rámec jeho placených služeb. Řízení toku se realizuje pomocí front, které omezují maximální přenosovou rychlost. Fronty jsou tvořeny hardwarem na aktivních prvcích, nebo mohou být softwarově nastaveny.

1.3.1 Základní příklad

Ideální realizace řízení toku v ISP síti je provedena tak, aby do každé lokality, do každého odběrného místa byla distribuována garantovaná přenosová kapacita. Garantovaná přenosová kapacita je brána jako přesně definovaná přenosová rychlost, neměnná v čase. Při stromové struktuře by se kapacity těchto míst směrem ke kořenu sčítali. V každém patře stromu by byla garantovaná rychlost spoje. Technika je zobrazena na obr. 1.2. Šipky ilustrují připojené trasy a jejich černý popisek definuje připojenou kapacitu oběma směry. Řešení se podobá síti s přepínáním okruhů. Předpokladem je komunikace klientů vždy směrem přes kořen stromu, jak je to v ISP síti nejčastější. V této topologii nemůže dojít k přetížení žádného síťového prvku. Nevýhoda této topologie je velká přenosová kapacita na páteřních spojích, která ze statistického hlediska není téměř nikdy využita. V praxi tak narážíme na extrémně velké pořizovací náklady [5].

1.3.2 Agregace

Agregace je metoda na snížení nákladů při budování sítě. Je nepravděpodobné, že všichni uživatelé budou v jednom čase využívat připojení na maximum. S tímto



Obr. 1.2: Distribuce garantované přenosové rychlosti

počítá agregace a sdílí fyzické prostředky mezi účastníky sítě [8]. Při sdílení je kapacita sítě menší, než připojená kapacita do sítě. Drtivá většina dnešních Ethernetových sítí je agregována, ať už se jedná o Local Area Network (LAN), Metropolitan Area Network (MAN) nebo Wide Area Network (WAN) sítě. Díky této metodě můžeme snížit přenosové kapacity páteřních tras podle typu účastníků několikanásobně. U domácích přípojek není garantovaná rychlost požadována a spotřeba dat je malá, proto je zde prostor pro agregaci. Pokud odběratelé vyžadují dodržení garantované rychlosti, jako například firmy, je možnost agregace omezená.

Jako příklad můžeme uvést průměrnou domácnost v síti Durnet.cz. Za období 16. 10. 2015 až 15. 11. 2015 přenesla 73,7 GB dat. Průměrná rychlost přípojky sítě Durnet.cz podle měření na serveru http://www.rychlost.cz ke dni 15. 11. 2015 je 20 Mb/s. Pokud by klient využíval přípojku naplno, přenesl by za 30 dnů

$$B = \frac{R \cdot M}{8 \cdot 1000} = \frac{20 \cdot 60 \cdot 60 \cdot 24 \cdot 30}{8 \cdot 1000} = 6480 \text{ GB} \tag{1.1}$$

dat. Kde R označuje rychlost stahování v Mb/s a M počet sekund za 30 dnů. Můžeme vypočítat procentuální vytížení přípojky průměrného uživatele.

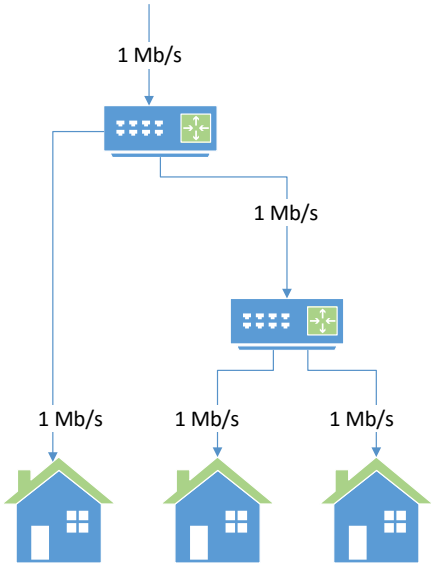
$$P = \frac{D}{B} \cdot 100 = \frac{73{,}7}{6480} \cdot 100 \doteq 1{,}14\ \% \tag{1.2}$$

D zobrazuje průměrný počet přenesených dat za 30 dnů a B zobrazuje maximální počet přenesených dat za 30 dnů. Průměrná domácnost připojená přes síť Durnet.cz

spotřebovává jenom 1,14 % maximální kapacity své přípojky. Během nečinnosti může kapacitu využívat jiný uživatel. Je zde možné uplatnit mechanismus agregace.

1.3.3 Aplikace do základního příkladu

Jako příklad můžeme použít topologii zobrazenou na obr. 1.2 diskutovanou v kap. 1.3.1. Pokud na tuto topologii aplikujeme metodu agregace razantně snížíme požadavky na síťové zdroje. Na obr. 1.3 je patrné, že přenosová rychlost v kořenu stromu klesla trojnásobně. Podmínkou této sítě je, aby v jednom okamžiku využíval připojení jen jeden uživatel. V takovém případě jsou přenosové parametry sítě stejné jako u garantované distribuce dat v základním příkladu.



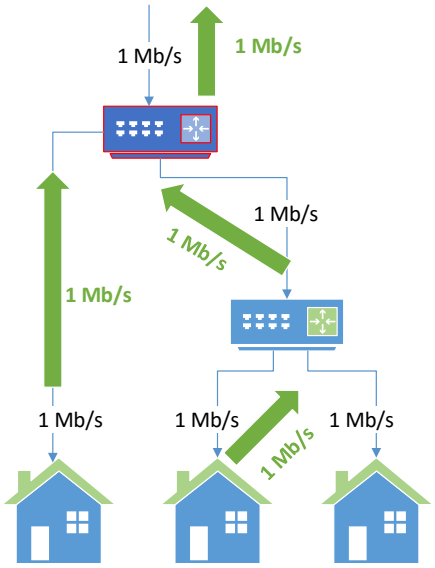
Obr. 1.3: Distribuce agregované přenosové rychlosti

1.3.4 Přetížení agregovaného prvku

Se sdílením síťových prvků však vzniká možný problém přetížení. Z pravděpodobnostního hlediska musí dojít k přetížení síťového prvku, který je agregován. Při větším sdílení síťových zdrojů je pravděpodobnost zahlcení vyšší, naopak při malé agregaci je pravděpodobnost nízká. Přetížení můžeme popsat jako jev, při kterém k síťovému prvku přichází více dat, než kolik je schopen odesílat. Reálný síťový prvek zaplní své paměti daty, které čekají na odeslání a zvýší tak odezvu sítě. Zvýšená odezva je první z nežádoucích vlastností přetížené sítě. Druhou nežádoucí vlastností

je zahazování paketů, které vznikne při úplném zaplnění pamětí. Výkonnost sítě při přetížení klesá, protože nejsou dodrženy základní parametry sítě a požadavky služeb nejsou uspokojeny [4], [6], [5].

Příkladem popíšeme přetížení v topologii z obr. 1.3 v kap. 1.3.3 Aplikace do základního příkladu. Na obr. 1.4 je zobrazen stav přetížení. k popiskům připojených kapacit černým písmem přibylo zelené písmo zobrazující aktuální datový tok. Zleva první a druhá domácnost odesílají maximální rychlostí data do sítě. Problém nastává v aktivním síťovém prvku u kořenu stromu. Tomuto prvku přicházejí data od obou účastníků rychlostí celkem 2 Mb/s, přitom jeho maximální rychlost odesílání je jen 1 Mb/s. Zvýrazněný síťový prvek nestíhá data odesílat, je přetížen.



Obr. 1.4: Přetížení v agregované síti

1.4 Zabezpečení sítě proti přetížení

Stav přetížení není akceptovatelný, ISP síť musí být postavena tak, aby k přetížení nedocházelo. Zařízení mohou podle různých normovaných metod informovat generátory provozu v síti o svém stavu zahlcení, aby omezily množství vysílaných dat. Mezi tyto metody patří Zpětný tlak, Tlumící paket, ECN a další. Níže jsou popsány nejdůležitější z nich. Nevýhodami je zátěž síťových prvků a omezená možnost řízení přetížení.

1.4.1 Zpětný tlak (Backpressure)

Tato metoda zamezuje zahazování paketů, na úkor odezvy sítě. Principem metody je informovat nejbližší síťový prvek, aby omezil odesílání dat. Data tak nejsou zahazována, ale jsou dočasně uchována ve vyrovnávacích pamětích síťových prvků. Jakmile jsou volné síťové zdroje, jsou data odeslána [11]. Problémem metody je zvyšování odezvy, neboť některá data pro služby v reálném čase nemusí být doručena včas.

Pro lepší představu můžeme přirovnat metodu k dopravnímu kolapsu ve městě, kdy vozy jsou pakety a fronty před křižovatkou reprezentují fyzickou paměť síťového prvku. Stejně tak v dopravním kolapsu jako při metodě zpětného tlaku automobily zůstávají stát před volnou křižovatkou a nepostupují k další křižovatce, protože místo před další křižovatkou je plně obsazeno. Analogicky při zpětném tlaku nepostupují pakety z jednoho síťového prvku k přetíženému, protože přetížený prvek má plné vyrovnávací paměti.

1.4.2 Tlumící paket (Source Quench)

Metoda se snaží informovat odesílatele dat o aktuálním stavu sítě. Pokud je některý prvek v síti přetížen, informuje odesílatele dat, aby zpomalil vysílání. Na rozdíl od předchozí metody nespolupracujeme s nejbližším sousedem, ale zprávu zasíláme až zařízení, které tyto data generuje do sítě. Přetížený prvek sítě informuje zdroj odesílání pomocí zprávy „Source Quench“ ze sady Internet Control Message Protocol (ICMP). Zpráva se generuje v síťovém prvku při úplném přetížení, nebo preventivně již před úplným přetížením. Při úplném přetížení se s zahazovaným paketem vytvoří informační zpráva s cílovou adresou zdroje původního paketu. Při preventivním odesílání informační zprávy se jako rozhodná hranice bere množství zaplněné paměti v síťovém prvku. Paměť ještě nemusí být plná, ale protože se plnému stavu blíží, již se generuje tlumící paket. Jako cíl zprávy je vybrán náhodný odesílatel z paměti nezpracovaných rámců v síťovém prvku [10], [9].

1.4.3 Zahazování paketů

Jedna z nejzákladnějších metod kopíruje fyzické vlastnosti sítě. Síťové zařízení nemusí podporovat žádné speciální algoritmy. Přetížené zařízení není schopno zpracovat další příchozí rámce, proto je zahazuje. Nedoručený rámec signalizuje přetížení sítě. Transmission Control Protocol (TCP) již obsahuje metody, které zajistí opětovné odeslání dat do sítě a zpomalení vysílaných dat. U ostatních protokolů je reakce na ztracený paket na vývojáři aplikace.

1.4.4 Explicit Congestion Notification (ECN)

Explicit Congestion Notification (ECN) je rozšíření Transmission Control Protocol/Internet Protocol (TCP/IP) standardu. Do češtiny je ji možné přeložit jako explicitní signalizace zahlcení. Umožňuje informovat koncové komunikující uzly o stavu zahlcení sítě. Šetrnost této metody nevyžaduje zahazování paketů, ani zvyšování odezvy sítě. Podmínkou funkčnosti algoritmu je podpora v síťových prvcích a v komunikujících uzlech. Již dnes některé zařízení a operační systémy ve výchozím nastavení podporují ECN [12], [4]. Zobrazení aktuálního stavu používání ECN je v kap. 3.

Popis algoritmu

Algoritmus zakládá na dobré znalosti aktuálního stavu sítě. Pokud se síťové prvky začínají nacházet ve stavu přetížení mohou tuto skutečnost oznámit komunikačním uzlům pomocí ECN. Komunikující stanice zpomalí rychlost odesílaných dat do sítě a nedojde k přetížení. Informace o stavu ECN se předává pomocí čtrnáctého a patnáctého bitu v hlavičce IP paketu. Bity patří do pole označeného DiffServ. Méně důležitý bit je označen jako ECT a vyšší bit jako CE [12], [4].

- 00 - Not-ECT (Not ECN-Capable Transport) Přenos ECN není podporován.
- 01 - ECT(1) (ECN-Capable Transport(1)) Řízení toku pomocí ECN je podporováno koncovými stanicemi.
- 10 - ECT(0) (ECN-Capable Transport(0)) Řízení toku pomocí ECN je podporováno koncovými stanicemi.
- 11 - CE (Congestion Experienced) Řízení toku pomocí ECN je podporováno koncovými stanicemi a síťový prvek označil paket informací o svém přetížení.

Při navazování spojení se koncové stanice domluví, zda mohou použít ECN při přenosu, výsledek uvedou do každého odesílaného paketu. Síťové prvky mohou stavem přetížení označit jenom pakety podporující ECN. To znamená jenom ty, které mají CE a ETC bity v kombinacích 10 nebo 01. Pokud je paket označen symbolem přetížení (11) musí se komunikující stanice dohodnout na zpomalení přenosu [12].

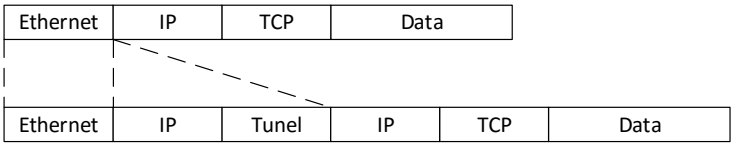
TCP řídí rychlost přenosu dat sám, proto není implementace náročná. Informace o přetížení sítě je uvedena ve třetí vrstvě TCP/IP. Pro řízení toku je nutné tuto informaci zpracovávat ve čtvrté vrstvě. Zařízení podporující ECN přenos jsou touto funkcí vybaveny a přetížení sítě je zpracováváno ve třetí vrstvě. k paketu označenému stavem přetížení (11) je nahlíženo jako k zahozenému paketu. Na označený nebo zahozený paket se reaguje zmenšením okna zahlcení na straně vysílače o polovinu [12].

Implementace do ostatních protokolů například User Datagram Protocol (UDP), Real-time Transport Protocol (RTP) a další je možná. Tyto protokoly však neobsa-

hují řízení propustnosti na čtvrté vrstvě svého modelu. Reakce na označený paket je čistě na vývojáři aplikace [12].

ECN v tunelech

Explicitní signalizace počítá s veškerým provozem a orientuje se podle hlavičky paketu. ECN patří na třetí vrstvu modelu ISO/OSI, proto všechny hlavičky zde uvedené jsou chápány jako hlavičky IP na této vrstvě. Při využití IP tunelu nastává komplikace. Každý paket procházející sítí má své záhlaví. Jak je zobrazeno v horní



Obr. 1.5: Zobrazení záhlaví IP paketu při použití tunelu

částí obr. 1.5, záhlaví je pouze jedno. Při použití tunelu se k paketu přibalí další. Paket nyní obsahuje dvě záhlaví, každé má své pole ECN. První patří zařízením, které mají mezi sebou vytvořený tunel. Druhé patří komunikujícím stanicím, které dokážou regulovat rychlost přenosu a měla by jim být doručena informace ECN. V některých případech, například pomocí funkce IPsec, mohou být data včetně původní hlavičky zašifrována. V tomto případě je čitelná pouze hlavička tunelu a není možné zapisovat do dat patřících komunikujícím zařízením. Informace o stavu sítě však musí být doručena komunikujícím zařízením. Z toho vyplývá, že pro správnou funkci explicitní signalizace musí být zavedena podpora v principech tunelování provozu [13].

V doporučení RFC 6040 jsou popsány dvě metody podpory ECN v tunelech. První metoda, nazvaná normální mód v anglickém dokumentu „normal mode“ již obsahuje podporu ECN. Podpora je zavedena tak, že tunelovací protokol při rozbalování, nebo zabalování kopíruje dva bity značící ECN. Informace o přetížení z paketu zabaleného bude zkopírována na originální paket a doručena komunikujícím uzlům. Druhý mód je nazván jako mód kompatibility „compatibility mode“, obsahuje zpětnou podporu s původní verzí bez podpory ECN [13].

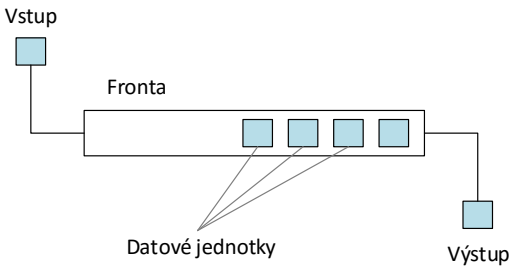
1.5 Fronty

Fronty jsou součástí každodenního života jak v telekomunikacích, tak ve skutečném světě. Frontu je potřeba realizovat před každým místem zpracování. Například při

kupování karty na tramvaj se může stát, že se kupující dostane do fronty. Podobné fronty se mohou tvořit před přepážkou na poště, v menze na jídlo a všude kde probíhají operace, které trvají nějakou dobu. V telekomunikacích je tomu obdobně frontu před zpracováním dat musí mít i síťový prvek. Fronta může být tvořena fyzicky, přímo vlastní pamětí k tomu určenou nebo pomocí softwaru vytvořena na místě, kde žádná fronta původně nebyla. Na frontu se můžou pojit další mechanizmy, které uskladněná data mění. Můžou to být například QoS a Random Early Detection (RED). V kap. 2 jsou popsána zařízení a i možnosti jejich front [4].

1.5.1 FIFO - First in first out

Je nejzákladnější a nejznámější typ. Objekt, který do fronty vstoupil dříve, je dříve

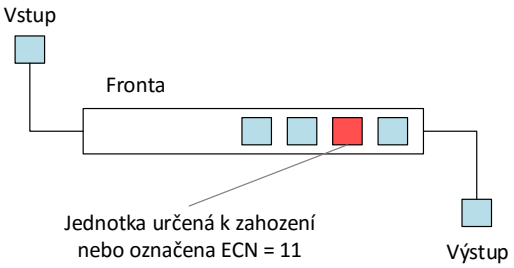


Obr. 1.6: Fronta typu FIFO

odebírán. Princip je zobrazen na obr. 1.6. Nejdůležitějším parametrem tohoto typu je velikost paměti. U fyzické realizace této fronty délka přímo odpovídá velikosti paměti. U logické (softwarové) tvorby této fronty jsou data ukládána do operační paměti síťového prvku. Operační paměť je pomalejší, ale často větší než předchozí fyzická paměť přímo k tomu určená. Při logické realizaci front můžeme délku určit podle počtu paketů (PFIFO), nebo podle délky fronty v bajtech (BFIFO). Pokud do paměti přijde více dat než je schopna uschovat, jsou zahozeny [4].

1.5.2 RED - Random early detection

Je mechanismus určený k zamezení přetížení sítě. Principem je detekovat zaplnění paměti síťového prvku a podle toho náhodně odstranit některé pakety. Zahozením některých paketů komunikující stanice detekují chybu v síti a zpomalí přenos dat. Metoda pracuje na FIFO paměti, a je přímo určená k omezování rychlosti dat [4]. Blokový diagram je zobrazen na obr. 1.7.



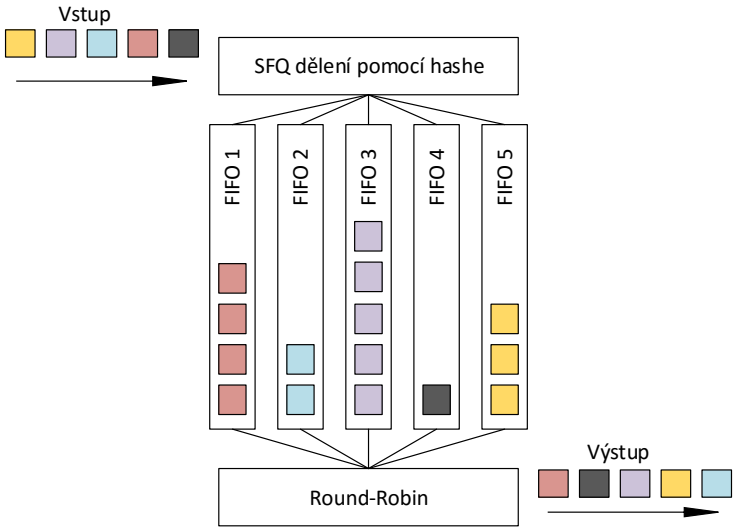
Obr. 1.7: Fronta typu RED

RED s ECN

Pokud síťový prvek podporuje ECN signalizaci, je nejčastěji implementována do RED metody. Funkce je stejná, pouze před zahozením paketu se ověří podpora ECN. Pokud paket explicitní signalizaci podporuje, je pouze označen stavem přetížení a zařazen zpět do fronty k odeslání. V opačném případě je zahozen, jako při základní funkci algoritmu RED [18], [12].

1.5.3 SFQ - Stochastic fairness queueing

Algoritmus se snaží zajistit spravedlivé dělení šířky pásma. Princip je zobrazen na obr. 1.8. Datový tok je rozdělen do takzvaných „flows“ podle hashovací funkce. Flows reálně reprezentuje fronta FIFO. Flows jsou pak rovnoměrně obsluhovány Round Robin algoritmem a předávány na výstup. Rozdělení do flows je možné přirovnat k rozdělení do skupin podle zdrojové adresy, cílové adresy, zdrojového portu a cílového portu. Výstupní pořadí paketů nemusí být stejné jako na vstupu. Cílem je rovnoměrné obsloužení všech služeb procházejících frontou [18].



Obr. 1.8: Fronta typu SFQ

2 VYBRANÁ ZAŘÍZENÍ A JEJICH ALGORITMY PROTI PŘETÍŽENÍ

Dnešní sítě obsahují mnoho zařízení od různých výrobců. Díky normám mohou tyto zařízení komunikovat mezi sebou. Pokud výrobce dodržuje doporučení Request For Comments (RFC) nebo standardy Institute of Electrical and Electronics Engineers (IEEE) je jeho zařízení kompatibilní s ostatními. Každé zařízení však nemusí podporovat všechny funkce. Například doporučení RFC jsou často opomíjena. Navíc výrobce může vytvořit vlastní funkci, která může být podporována jenom jeho zařízeními [2], [3]. V této části práce jsou popsána zařízení, ktera jsou používána v síti Durnet.cz. U zařízení jsou popsány funkce, které souvisí s řízením síťového provozu.

2.1 Mikrotik

Výrobce se sídlem v Lotyšsku se pyšní především vlastním operačním systémem umožnující velké množství síťových funkcí. Také vyvíjí a vyrábí vlastní hardware

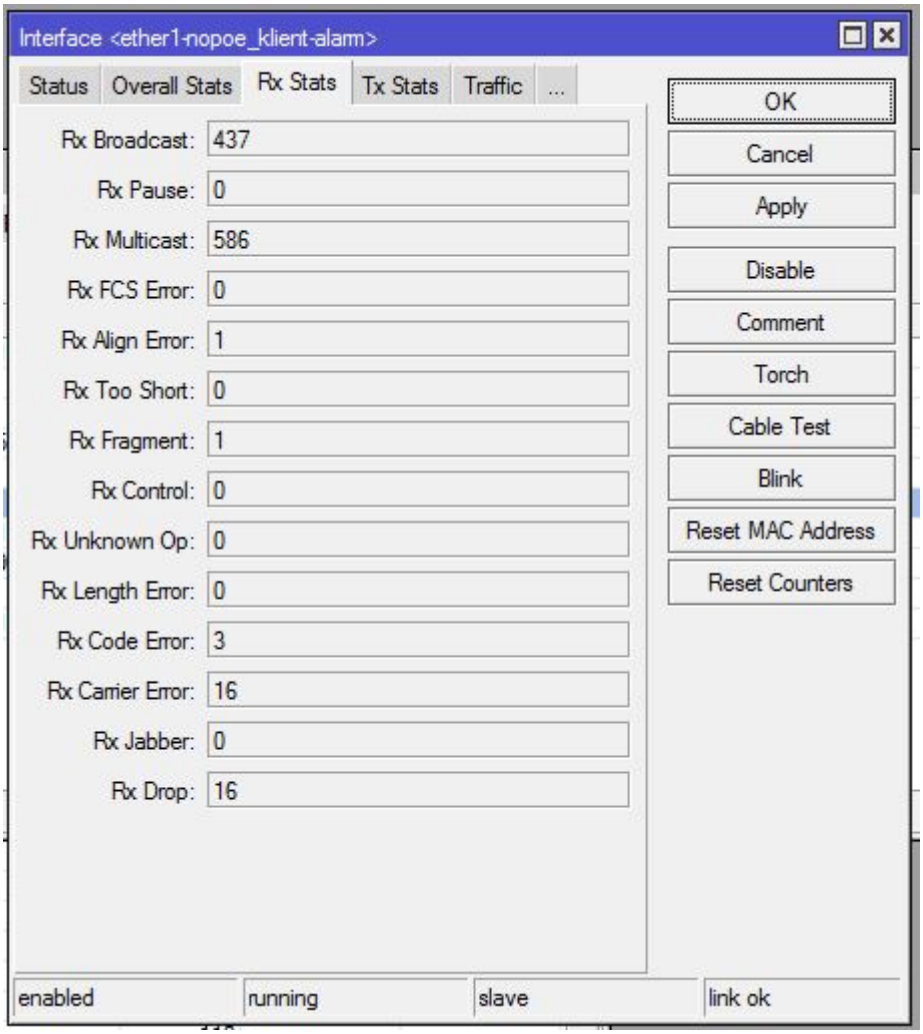
Obr. 2.1: Mikrotik RouterBoard 2011

nazvaný RouterBoard (RB), který se prodává s předinstalovaným operačním systémem RouterOS. Produkty jsou určeny pro ISP, převážně to jsou směrovače, přepínače a bezdrátové zařízení v pásmu 2,4 GHz a 5 GHz [14]. V síti Durnet.cz se od tohoto výrobce používají převážně směrovače s Ethernet porty. Největší četnost zastoupení má model RB2011, obsahuje pět portů Ethernet o rychlosti 1 Gb/s a pět o rychlosti 100 Mb/s. Směrovač je zobrazen na obr. 2.1.

Ochrany proti přetížení

V současnosti RouterOS neumožňuje podporu ECN. Při regulaci rychlosti toku se vždy uvažuje se zahozením paketů, nebo se zvýšením latence [14]. RouteOS podporuje metodu zpětného tlaku. Statistiky této metody jsou uvedeny v administračním

programu WinBox. Obr. 2.2 zobrazuje okno se statistikami Ethernet portu. Jedná se o statistiky přijatého provozu a je zde uvedena hodnota Rx Pause, která zobra-



Obr. 2.2: Mikrotik RouterOS statistiky Ethernet portu

zuje počet přijatých paketů zpětného tlaku. Podobné statistiky jsou uvedeny i pro odchozí provoz na obr. označený jako Tx Stats.

Mikrotik má propracovaný systém front. Na výběr jsou nejznámější FIFO, RED, SFQ fronty, ale také proprietární PCQ.

PCQ je speciální typ fronty, který se skládá z běžných FIFO front. Jedním z jeho parametrů je ukazatel nazvaný pcq-classifier. Ukazatel se nezadává konkrétní hodnotou, ale pouze se označí jméno hodnoty, podle které se pakety dělí do skupin. Ukazatelem můžou být adresy, nebo porty obsaženy v paketech, který prochází frontou. Pro každou skupinu se automaticky vytvoří zvláštní FIFO fronta [14]. Jeho použití může být například při spravedlivém dělení datového toku v domácí síti. Ukazatel by

byl v tomto případě nastavený na IP adresu a každý počítač by dostal rovnoměrnou šířku pásma.

2.2 Ubiquiti Networks

Výrobce se sídlem v Kalifornii vyniká především jednoduchými a funkčními řešeními. Zaměřuje se převážně na bezdrátové zařízení a profesionální telekomunikační zařízení [15]. Firma vyvíjí a vyrábí síťový hardware převážně se zaměřením na bez-

Obr. 2.3: Rocket M5

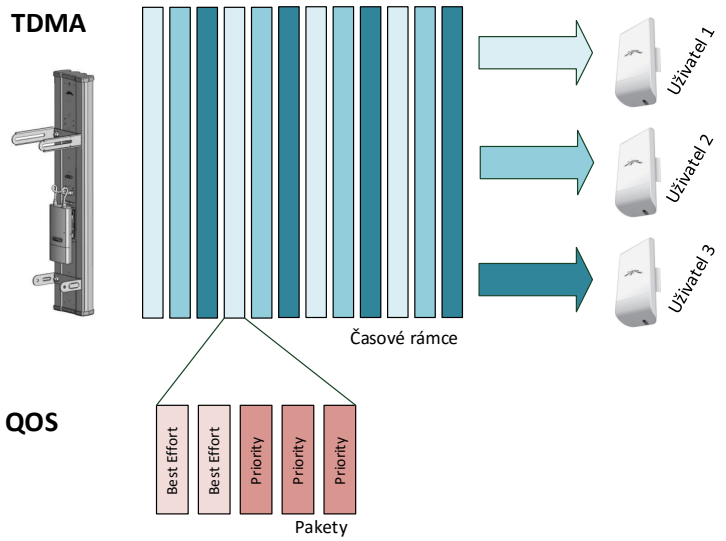
drátový přenos dat. Dalšími produkty jsou IP telefony, chytré prvky do domácnosti a další. Produkty má přehledně rozděleny do kategorií. V síti Durnet.cz je nejčetněji použita kategorie Airmax produktů. Název Airmax označuje proprietární bezdrátový protokol, který je navržen přímo pro potřeby ISP. Protokol byl vyvinut z normy 802.11, ovšem nejsou kompatibilní. Vyráběná zařízení mají v bezdrátové části implementovanou fyzickou podporu Airmax protokolu [15].

V síti Durnet.cz se nejčastěji používají jednotky označeny Rocket M5 viz obr. 2.3. Radiová jednotka má nahoře vodotěsné konektory Reverse polarity SMA connector (RSMA) pro připojení antény. Konektory jsou dva, protože jednotka umožňuje komunikaci v Multiple input, multiple output (MIMO) režimu pro zvýšení propustnosti. Dva kanály jsou nejčastěji odděleny pootočením polarizace antén o 90°, tím je zajištěn dostatečný útlum, aby bylo možné komunikovat oběma kanály současně. Výrobce připravil pro jednotku i antény se dvěma polarizacemi. Všechny potřebné

antény jsou připraveny, sektorové, všesměrové i směrové ve všech velikostech. Jednotka má univerzální držák, se kterým se může snadno připojit na kompatibilní anténu. Na čele jednotky jsou umístěny signalizační diody pro rychlé orientační nastavení signálu a odečtení aktuálního stavu zařízení. Na obrázku zespodu má jednotka RJ-45 konektor se kterým se připojuje k Ethernetu. Napájení je řešeno pomocí pasivního Power over Ethernet (POE), vodiče 4 a 5 vedou kladný pól a vodiče 7 a 8 záporný. Toto zapojení není kompatibilní s mezinárodní normou IEEE 802.3af. Jednotku obsluhuje operační systém AirOS postavený na linuxovém jádře [15].

Ochrany proti přetížení

V současnosti AirOS neumožňuje podporu ECN, [15]. Veškerá regulace rychlosti je realizována zahazováním paketů a zvyšováním odezvy.



Obr. 2.4: Airmax TDMA s QOS

Airmax obsahuje Time Division Multiple Access (TDMA) přístup k médiu, na rozdíl od IEEE 802.11, kde je použita metoda Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). Tato výhoda umožní spravedlivé obsloužení připojených bezdrátových stanic a zamezí kolizím na bezdrátovém médiu. Přístupový bod rozdělí svůj čas tak, aby udržoval kontakt s každou stanicí. Přesněji rozdělí svůj čas do timeslotů a jednotlivým stanicím pak přiřadí timeslot, při kterém spolu mohou komunikovat. U stanic je možné nastavit Airmax prioritu. Stanice s vyšší prioritou dostane vyšší podíl timeslotů a může tak komunikovat rychleji. Žádná stanice

nemůže přetížit přístupový bod tak, aby ostatní stanice nebyly obslouženy. Jedná se také o metodu ochrany proti přetížení, ovšem prozatím z úplně jiného pohledu [15].

Popisovaná jednotka podporuje také prioritizaci provozu IEEE 802.11e. Protože bezdrátová část bývá z pravidla nejpomalejší částí jednotky, je na ní aplikováno rozšíření QoS, které zamezí zpoždění, nebo zahození prioritního provozu. Prioritní provoz musí mít označení v hlavičce paketu v Type of service (TOS) poli. Toto pole je přímo určené pro značení priority provozu podle IEEE P802.1p. Prioritně označený provoz, nazývaný také jako real time, je upřednostněn před ostatním provozem a z čekací paměti určené na odeslání je odeslán jako první [15]. Tato metoda není přímo ochranou proti přetížení, je však lepší v případě problému ochránit a doručit důležitá data, než data na pozadí, která mohou být zpracována později.

Na obr. 2.4 jsou zobrazeny dva principy ochrany proti přetížení, které jsou popsány výše.

2.3 Alcoma

Alcoma a.s. je český výrobce bezdrátových spojů Point to point (PTP) vyniká vysokou kvalitou a robustním provedením zařízení [16]. Výroba je zaměřena jenom na bezdrátové spoje, jiné produkty nejsou v nabídce. Sídlo firmy je v Praze. Takzvaně

Obr. 2.5: Alcoma MP300

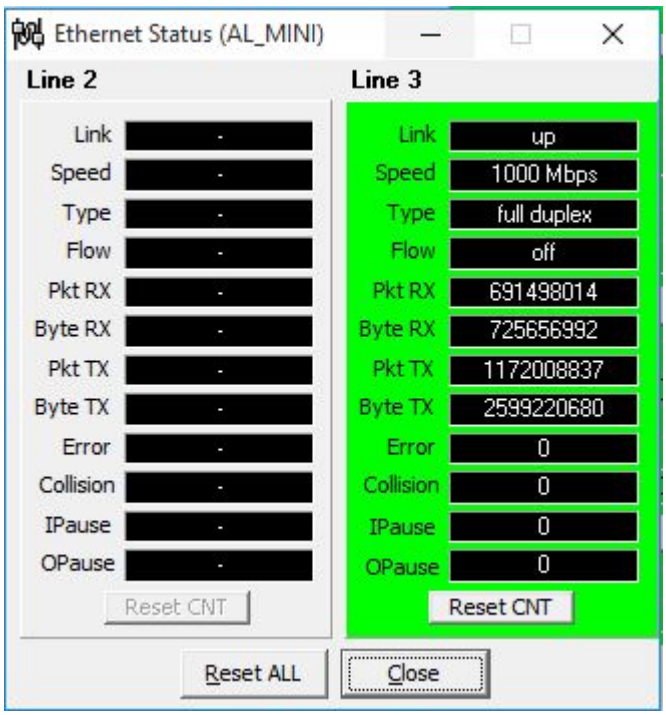
vlajková loď jsou v současnosti spoje MP600, které nabízí kapacitu oběma směry až 900 Mb/s. V Durnet.cz síti jsou nasazeny spoje kategorie MP300, jedná se o levnější

variantu. Oproti vyšší MP600 postrádá funkce jako je QoS, podporu virtuálních sítí (VLAN) a má menší maximální přenosovou rychlost a to 500 Mb/s obousměrně. Alcoma MP300 i s anténami je zobrazena na obr. 2.5. Pro administraci slouží program ASD, který nabízí kompletní dohled i správu spoje.

Informace o spojích byly čerpány z datových listů na webů výrobce a navíc konzultovány s obchodním reprezentantem panem Pavlem Tůmou.

Ochrany proti přetížení

V současnosti spoje od firmy Alcoma neumožňují podporu ECN, [16].



Obr. 2.6: ASD informace o stavu Ethernet rozhraní

Pokud je spoj přetížen, oznamuje tuto skutečnost nejbližšímu zdroji dat pomocí metody Zpětného tlaku, který je popsaný v kap. 1.4.1. Statistiky odeslaní těchto paketů jsou uvedeny v administračním rozhraní ASD. Na obr. 2.6 je zobrazeno okno statistik Ethernetového portu na spoji Alcoma MP300. Statistiky o odeslaných paketech jsou uvedeny v polích IPause a OPause ve spodní části obrázku. IPause (Input Pause) reprezentuje počet přijatých paketů. OPause (Output Pause) reprezentuje počet vygenerovaných paketů zpětného tlaku.

2.4 Linux

Operační systémy založené na Linuxu podporují ECN od verze jádra 2.4.20 a vyšší. Podpora ECN je zde hned ve dvou funkcích systému. Jedna z možností, kdy jádro operačního systému Linux podporuje ECN je ve stavu komunikujícího uzlu. Při sestavení TCP spojení se komunikující strany dohodnou na podpoře ECN. Pokud je mechanizmus podporován na obou stanicích, může být nadále povolen i ve spojení. Nastavení podpory ECN je v systémovém souboru `/proc/sys/net/ipv4/tcp_ecn`. Hodnoty souboru ovlivňují podporu ECN následovně:

- 0 - ECN není podporováno.
- 1 - Podpora ECN při odchozích i příchozích spojení.
- 2 - Podpora ECN pouze při příchozím spojení.

Ve výchozím nastavení systému Ubuntu je v současnosti hodnota 2, tedy podpora ECN jen v případě příchozího TCP spojení. Druhá možnost podpory ECN je ve funkci síťového prvku. Ve virtuální frontě Hierarchical token bucket (HTB) je možnost zapnout podporu ECN, kompatibilní pakety nebudou zahazovány, ale budou označeny ECN stavem přetížení [18].

HTB je mechanismus implementován přímo v jádru operačního systému Linux, umožňuje vytvářet virtuální fronty s různými typy. Mezi nejčastější typy front patří FIFO, HTB, RED a SFQ. Mechanizmus nemá grafické rozhraní a je kompletně ovládán z terminálu pomocí programu Tc [18].

2.5 Uživatelský počítač

Z nejznámějších operačních systémů podporují ECN všechny. Apple oznámil zavedení podpory ECN v červnu roku 2015 [20]. Microsoft podporuje ECN ve verzích Windows server 2008, Windows Vista a novějších [19].

3 VYUŽITÍ ECN V SOUČASNÉ DOBĚ

Již v teoretickém úvodu v kap. 1.4.4 je popsána podpora explicitní signalizace přetížení (ECN) ve velkém množství zařízení. V této sekci je zobrazeno, jaký podíl přenesených paketů je kompatibilní s explicitní signalizací zahlcení (ECN) v současnosti.

3.1 Metodika měření

Pro zobrazení aktuálního využití ECN v praxi je nutné pracovat s reálným provozem. Reálný provoz byl zachycen na síti Durnet.cz v období 25. 11. 2015 až 2. 12. 2015. Provoz byl zachycen na hraničním směrovači sousedícím se sítí společnosti Avonet s.r.o. Směrovač je umístěn v jižní polovině sítě viz kap. 1.1. Na severní bráně nemůže být realizováno zachytávání provozu z důvodu slabého výkonu síťového prvku.

3.1.1 Měření

Cílem bylo zachytit velké množství paketů procházející směrovačem. Pro zachytávání dat byl použit nástroj Tcpdump, je to nejznámější program pracující na platformě Linux v textovém režimu. Tímto nástrojem jdou lehce vypsat pakety procházející jakýmkoli síťovým rozhraním. Pro zachytávání provozu v Linuxu potřebuje program plné oprávnění, to je oprávnění superuživatele root. Potřebné oprávnění v teminálu umožní systémový program sudo. Zachycení a zobrazení paketů ze síťového rozhraní s názvem „mistni“ můžeme jednoduše udělat příkazem:

```
sudo tcpdump -i mistni
```

Protože směrovačem prochází v době nejvyššího zatížení až patnáct tisíc paketů za sekundu, zobrazení na monitor a ruční zpracování není možné. Data byla nejprve uložena na disk, pak zpracována. Aby Tcpdump nezobrazoval data do konzole, ale uložil je na disk, musí se použít parametr -r soubor.pcap. Při zachytávání se vytvoří soubor s názvem uvedeným za parametrem. Data, která pakety přenáší jsou při tomto měření nepotřebná. Pro měření stačí zachytit hlavičky paketů. Úspora místa je definována parametrem „-s 96“, dodatek uloží pouze prvních 96 bytů paketu. Kompletní příkaz pro zachytávání potřebných dat je:

```
sudo tcpdump -i mistni -r pakety.pcap -s 96
```

Pro lepší manipulaci bylo zachytávání zhruba po jednom dnu zastaveno a spuštěno znova s jiným názvem souboru. Vzniklo tak více souborů. Všechny tyto soubory dohromady mají velikost 363 GB, jejich obsah je možné zobrazit příkazem

```
tcpdump -r pakety.pcap -nvS
```

, kde parametr `n` zamezuje reverzní překlad IP adres na doménové jména, parametr `v` vypisuje podrobnější statistiky IP protokolu (mezi nimi i TOS), parametr `S` vypisuje absolutní čísla. Část z těchto souborů je uvedena v příloze B. Zvýrazněny jsou zde informace o poli TOS, z tohoto pole budou čerpána data pro měření.

3.1.2 Zpracování

Pro prezentaci výsledků je nutné spočítat pakety, které mají stejně nastavené bity v poli ECN. Z důvodu velkého množství zachycených dat nepřipadá v úvahu ruční zpracování. Vyčtení dat je rozděleno do dvou částí. V první části se vyseparuje hodnota Differentiated services code point (DSCP) bajtu ze zachycených paketů. V této hodnotě jsou uvedeny i dva bity signalizující ECN viz teorie v kap. 1.4.4. V druhé části jsou data za pomocí programu rozděleny do čtyř skupin podle hodnot bitů v poli ECN. Jednotlivé skupiny jsou pak sečteny a tím zobrazen podíl paketů v závislosti na hodnotě ECN bitů.

Separace pole DSCP

Pro separaci pole nepřipadá v úvahu přehledný nástroj Wireshark, ani Tshark, protože veškeré zpracovávané data jsou uchována v paměti programu. Nároky na operační paměť by byly extrémně velké. Program Tcpdump však čte data sekvenčně a historii neuchovává v paměti, proto je vhodný. Z celkových hlaviček paketů nám postačí vypreparovat pole DSCP. Vytvořený skript v terminálovém prostředí BASH v Linuxu spojuje hned tři programy.

```
tcpdump -r zachycene_pakety.pcap -nvS
| awk '/tos␣0x/␣{␣print␣\$4␣}'
| cut -c3 > pakety_vypreparovane.txt
```

První program Tcpdump sekvenčně čte soubor se zachycenými pakety a vypisuje podrobné statistiky, mezi které patří i pole DSCP. Další použitý program je Awk, jedná se o pokročilý nástroj pro práci s textovými řetězci. Jeho úkolem je zde nalezení řádku, na kterém se nachází DSCP pole a zobrazení jeho hodnoty. Výsledkem v této části je textový řetězec ve formátu například `0x0`, který zobrazuje absolutní hodnotu pole DSCP. Poslední program Cut vyfiltruje vše kromě třetího znaku řetězce. Na výstup se tak dostane pouze bajt absolutní hodnoty pole DSCP. Tento bajt je uložen do textového souboru pro další zpracování. Celkové zpracování dat trvalo na procesoru Intel Core i3 téměř dva dny.

Souborů se zachycenými pakety bylo více, proto je i více výstupních souborů ve formátu prostého textu. Pro jejich spojení byl použit příkaz:

```
cat jeden_ze_souboru.txt >> vysledny_soubor.txt
```

Příkaz Cat vypíše celý obsah souboru a přidá jej na konec souboru. Tímto způsobem byli všechny soubory spojeny do jediného.

Sečtení výsledků pomocí vlastního programu

Z připraveného textového souboru bylo zapotřebí sečíst stejné skupiny paketů podle hodnoty v bitech CE a ECT. Z dvou bitů vyplývá že skupiny budou celkem čtyři:

- CE = 0, ECT = 0 - Pakety bez podpory ECN signalizace
- CE = 0, ECT = 1 - Pakety s podporou ECN signalizace
- CE = 1, ECT = 0 - Pakety s podporou ECN signalizace
- CE = 1, ECT = 1 - Pakety s podporou ECN signalizace s označením přetížení

V textovém souboru je ovšem uložen celý bajt pole DSCP, rozdělení tak bude složitější a musí být počítáno s hodnotami přesahující velikost dvou bitů. Například pro hodnotu F, v desítkové soustavě 15, jsou oba bity CE a ECT nastaveny na logickou hodnotu 1 a musí být přičteny ke skupině s označením přetížení. Předchozí bity pole DSCP nejsou pro výsledek důležité, proto jsou ignorovány. Z důvodu velkého množství dat nepřipadá v úvahu práce s tabulkovým procesorem, například Microsoft Excel. Pro sečtení četnosti byl navržen program.

Z důvodu rychlosti zpracování byl program napsán v jazyku C++ a zkompilován přímo pro platformu Linux. Program nemá grafické rozhraní a veškerá práce s ním probíhá v terminálu. Zdrojový kód je uveden v příloze C.1. Funkce programu se dá rozdělit do čtyř částí. V první části se připraví proměnné ke zpracování, to znamená zpřístupnění souboru pro čtení a vytvoření proměnných pro zápis četnosti. Druhá část je reprezentována cyklem, který postupně čte řádky souboru a předává je třetí části. Cyklus se opakuje, dokud nejsou přečteny všechny řádky. Třetí část rozhodne, do které kategorie zaznamenaný paket patří a přičte dané kategorii množství. Čtvrtá část vypíše četnosti. V programu je vynechána diakritika pro lepší kompatibilitu.

Pro spuštění musí být program přeložen do strojového kódu, takzvaně zkompilován. Přeložení programu je realizováno programem `g++`:

```
g++ work.cpp -o work
```

Komplilátor G++ je volně dostupný a jeho výsledkem je spustitelný soubor v tomto případě s názvem `work`.

Po spuštění vytvořeného programu dostaneme konzolový výpis:

```
marek@marek-VirtualBox:~\$ ./work
Pocet celkovy: 3877785508
Pocet 00: 3844638477
Pocet 01: 30547070
```

```
Pocet 10: 1164619
Pocet 11: 1435342
```

V prvním řádku uvedeného textu je spuštěn vytvořený program, tečka a lomítko před názvem programu je syntaxe Linuxu potřebná pro spuštění. Práce programu trvala téměř čtvrt hodiny, pak byli vypsány výsledky.

Z těchto výsledků můžeme dopočítat přehlednější čísla pro zobrazení. Například celkový počet paketů podporujících ECN. Do této skupiny patří všechny pakety, které nemají CE a ECT bity nulové:

$$\begin{aligned} ECN &= Pocet01 + Pocet10 + Pocet11 = \\ &= 30547070 + 1164619 + 1435342 = \\ &= 33147031 = 33,147\ M \end{aligned} \tag{3.1}$$

Výpočet poměru paketů s dopředou signalizací zahlcení ku paketům bez signalizace zahlcení.

$$X = \frac{ECN}{Celkem} \cdot 100 = \frac{33147031}{3877785508} \cdot 100 = 0,85479\ \% \tag{3.2}$$

Kde ECN označuje celkový počet paketů s podporou ECN a $Celkem$ označuje celkový počet paketů.

3.2 Zobrazení výsledků

Během týdenního měření bylo zachyceno téměř 3,9 miliard paketů. Tyto pakety byly analyzovány pro zobrazení četnosti výskytu ECN. Výsledky měření jsou zobrazeny v tab. 3.1

Název	Hodnota
Celkový počet zachycených paketů	3,878 G
Počet paketů podporující ECN	33,147 M
Počet paketů označených stavem přetížení	1,435 M
Poměr paketů s ECN ku paketům bez ECN	0,854 %

Tab. 3.1: Využití ECN v současné době

Celkový počet zachycených paketů

Je absolutní hodnota počtu paketů, které byly zachyceny za měřené období od 25. 11. 2015 do 2. 12. 2015 v síti Durnet.cz. Přesněji uvedeno v kap. 3.1.

Počet paketů podporující ECN

Je absolutní hodnota počtu paketů, které jsou označeny podporou ECN signalizace. Logické spojení, do kterých patří tyto pakety jsou sjednány s podporou ECN signalizace. Bity CE a ECT musí být v nenulové, to je v kombinacích 01, 10 a 11.

Počet paketů označených stavem přetížení

Je absolutní hodnota počtu paketů, které podporují ECN signalizaci a jsou označeny stavem přetížení. Informace o stavech přetížení jsou uvedeny v kap. 1.4.4 popisující ECN.

Poměr paketů s ECN ku paketům bez ECN

Poměr zobrazuje, kolik paketů přenesených během měření je kompatibilní s explicitní signalizací ECN. Z celkového počtu zachycených paketů během měření je pouze 0,854 % kompatibilních s explicitní signalizací. Zbylých 99.146 % neumožňuje signalizaci ECN.

4 SOUČASNÉ SLABINY SÍTĚ PROTI PŘETÍŽENÍ

V této části je podle teoretických informací sestavena síť s mechanizmy zabraňujícími přetížením. Síť by měla být schopna vysoké agregace síťových zdrojů. Při sestavení jsou využity mechanismy uvedeny v doporučení RFC, ale také mechanismy vytvořené výrobci zařízení. Současně mohou být k řešení použity mechanismy řízení provozu frontami.

4.1 Další mechanismy řízení provozu

ISP musí poskytovat služby tak, aby se účastníci sítě navzájem co nejméně omezovali. Z toho vyplývá, že samotné mechanismy řízení datového provozu a ochrany proti přetížení nemusí stačit. Tyto mechanismy nepřihlížejí na rovnoměrné obsloužení uživatelů. Směrovače a přepínače často nemají ani hardwarové vybavení na to, aby přenášená data třídily rovnoměrně do skupin klientů. Většina ochran nemá ani možnost znalosti topologie, která je důležitá k rovnoměrnému rozdělení prostředků.

4.1.1 Selhání mechanismů směrovače Mikrotik

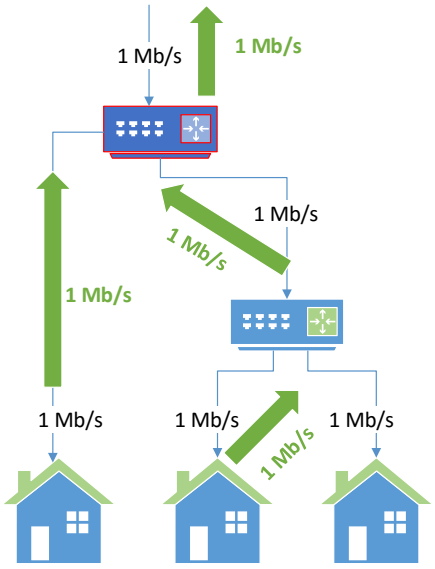
V případě nasazení směrovače RB 2011 do sdílené sítě mohou vzniknout komplikace související s přetížením i v případě, že směrovač má funkční výše popsané algoritmy na ochranu proti přetížení. Pro příklad můžeme převzít obrázek z kap. 1.3.4. Obr. 4.1 zobrazuje přetížený síťový prvek, který se snaží pomocí svých dostupných metod přetížení vyřešit. Pokud uvažujeme, že tento prvek bude RouterBoard 2011 obdobně jako v síti Durnet.cz, jsou jeho metody, které může použít zobrazeny výše.

První z metod je uskladnění do paměti zařízení. Ve výchozím nastavení má OS Mikrotik nastaveny fronty FIFO u všech rozhraní. To znamená, že nadbytečná data se snaží dočasně uložit do své paměti. Zaplněné paměti zvyšují odezvu sítě.

Druhá metoda je přirozené zahození nadbytečných paketů, které není možné uložit v paměti. V takovém případě vzniká v síti náhodné zahazování paketů. Zahazování paketů se aplikuje na celý provoz, takže i na uživatele, kteří nevyužívají připojení na maximum a požadují spolehlivé doručení dat.

Metoda zpětného tlaku je použita až tehdy, kdy dojde k zaplnění příchozí paměti rozhraní. Všechna data přijatá rozhraním jsou uložena do paměti a následně zpracovávána procesorem, nebo hardwarovým přepínačem. V případě správné funkce jsou pakety odebírány a zpracovávány několikanásobně rychleji, než jak mohou být přijímány. V případě zatížení procesoru jinou funkcí může dojít k hromadění paketů

v příchozí paměti rozhraní a tím ke generování paketů zpětného tlaku.



Obr. 4.1: Přetížení v agregované síti s směrovačem Mikrotik

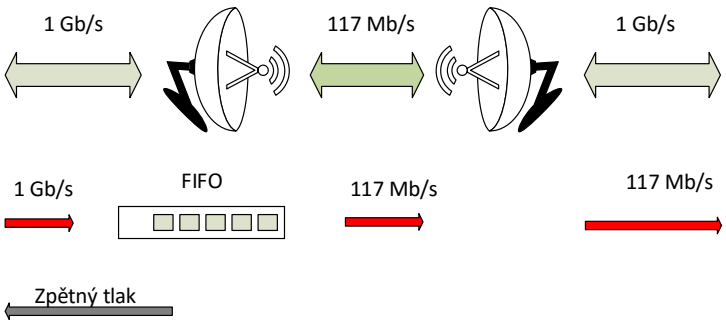
Dopad na ostatní uživatele a na služby

Při těchto metodách je odezva sítě vysoká a náhodně se objevuje ztrátovost paketů. Poslední domácnost, která připojení nevyužívá doplatila na zatížení vysokou odezvou. Při tomto stavu nemůže ani jeden uživatel používat služby běžící v reálném čase. Interaktivní služby budou omezeny a například prohlížení webového obsahu bude mít dlouhou odezvu. Navíc data u kterých se vyžaduje spolehlivé doručení, mohou být zahozena. Podle teoretických znalostí RouterBoard ve výchozím nastavení není imunní proti přetížení sítě.

4.1.2 Selhání mechanismů spoje Alcoma

Spoje Alcoma MP300 mají mechanismy proti přetížení popsány výše v kap. 2.3. Příkladná topologie je uvedena na obr. 4.2. Na obrázku jsou robrazeny dvě jednotky Alcoma, které vytváří transparentní bridge pro přenos paketového provozu. Zelené šipky zobrazují kapacitu připojení. Jednoty jsou po kabelu připojeny k směrovači rychlostí 1 Gb/s. Radiová rychlost jednotek je 117 Mb/s. Radiová rychlost závisí na licenci spoje, kvalitě signálu a síle signálu. V případě zhoršeného počasí se může

přenosová rychlost rádiové části snížit. Červené šipky zobrazují reálný provoz procházející spojem. Je zřetelné, že k radiovému spoji přichází větší množství dat, než jaké je schopen přenést. Zařízení začíná ukládat data do vyrovnávací paměti a tím



Obr. 4.2: Přetížení spoje Alcoma

zvýší odezvu spoje. Paměť je zobrazena jako FIFO fronta. V případě zaplnění fronty začíná Alcoma zahazovat nadbytečný provoz. Zpětný tak stejně jako u výrobků Mikrotik v tomto případe není použit.

Dopad na ostatní uživatele a na služby

Ani spoj Alcoma není dobře vybaven k řešení stavu přetížení. V případě přetížení se veškerému provozu zvýší odezva a některá data mohou být zahozená. Není definována žádná spravedlivost zahazování s ohledem na uživatele sítě. Proto může většinu kapacity obsadit jeden uživatel bez ohledu na ostatní. Například neregulovaným tokem UDP dat. Uživatelské vytížení spoje by se mělo pohybovat pod hranicí maxima. Ovšem u radiového spoje může být maximální rychlost proměnlivá v závislosti na počasí a rušení radiové frekvence.

4.2 Nedostatečné řešení uvedených metod

V ISP síti musí platit striktní pravidla pro přenos dat tak, aby každý uživatel mohl využívat jen zaplacené služby. Uživatelé by se měli v síti minimálně ovlivňovat převážně u kritického parametru jako je odezva. Šířka pásma musí být mezi účastníky taktéž rovnoměrně rozdělena. Metody popsané v teoretickém úvodu pracují pouze okrajově s rovnoměrným dělením síťových prostředků. Spíše jsou orientovány na krátkodobé přetížení a na potlačení následků. Pro komplexní správu sítě je nutné

uvažovat se sítí jako s celkem a na okrajích sítě definovat parametry příchozího provozu tak, aby data nezahltila vlastní síť, popřípadě vytvořit aktivní správu front, která bude sledovat dění v síti a omezovat nadbytečný provoz.

Teoretické metody v první části práce vůbec neuvažují se službami klientů připojených do sítě. Není zde žádná možnost nastavení tarifu, priority, maximální, nebo minimální rychlosti. Tyto parametry jsou klíčové v celkovém pohledu na síť, jedná se o hlavní parametry každé sítě. Proto jsou nezbytné další kroky k řízení provozu v ISP síti.

5 SLA UŽIVATELŮ SÍTĚ

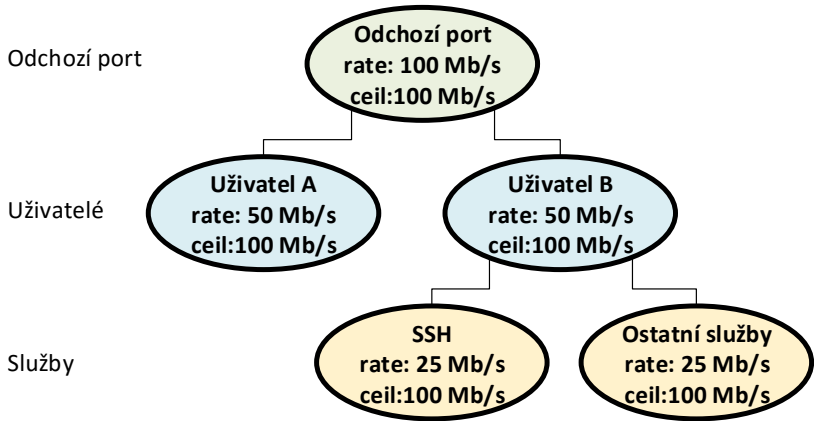
Z pohledu trhu jsou na síti nejdůležitějšími parametry SLA a cena. SLA představuje podmínky sjednané mezi klientem a poskytovatelem služby, například přenosovou rychlost, dostupnost a další. U domácí klientely se můžeme setkat i s označením tarif. Různí klienti sítě požadují různé SLA. Je na poskytovateli připojení, aby dokázal na svou síť připojit co největší počet odběratelů, ale přitom dodržel sjednané parametry. Z toho vyplývá, že v síti musí být algoritmy, které dokážou rozlišit klienty a postarat se o jejich požadavky dle SLA. Řízení provozu probíhá většinou omezením provozu uživatelů s nízkou přenosovou rychlostí. Tím se zajistí, aby nepřečerpávali své služby a nevyčerpali všechny sdílené zdroje [21].

5.1 Algoritmy řídící provoz

Omezení přenosových rychlostí se realizuje většinou programem v síťovém prvku. Reprezentanty těchto programů mohou být `HTB`, `CBQ`, které jsou obsaženy v jádře operačního systému Linux a ovládají se přes příkaz `tc`. S podobnými metodami se setkáme i u síťových prvků výrobce Ubiquiti a Mikrotik [14], [15], [18], [22].

HTB je program umožňující správu přenosové rychlosti procházející zařízením a sdílení rychlosti. Sdílením rychlosti se rozumí rozdělení přenosové rychlosti mezi uživatele v definovaném poměru, tato metoda se velmi často používá u domácí klientely s agregovanou službou. Nastavení programu a vyčítání informací se provádí pouze přes příkazový řádek Linuxu. Řízení provozu se aplikuje vždy na odchozím síťovém portu. Při nastavování programu je nejprve nutné nastavit na odchozí port výchozí třídu a až následně za ni přidávat třídy další [22]. Celé nastavení tak tvoří stromovou strukturu příklad této struktury je uveden na obr. 5.1. Každý oválný objekt se nazývá třída a můžeme si ho představit jako složkový systém v průzkumníku souborů na počítači. Aby byl program funkční, musí se procházející pakety označit a zařadit do jednotlivých tříd. Toto označení může být provedeno ve firewallu pomocí programu `iptables`, nebo pomocí programu `tc filter`. Program `iptables` je více podrobný a umožňuje lepší filtrování provozu. `tc filter` je daleko rychlejší, protože při filtraci používá hašovací tabulky. V každé třídě je možno definovat pravidla, jak se bude zacházet s provozem. Nejčastěji je zde vytvořena některá z front popsána v kap. 1.5. Při nastavení programu je nutné zadat dvě rychlosti, jedna z nich je označena jako `rate` a druhá jako `ceil`. Rychlost `rate` reprezentuje garantovanou rychlost, kterou může třída za jakýchkoli podmínek použít. Hodnota `ceil` je maximální přenosová rychlost, kterou je možno sdílet mezi potomky třídy. Tyto rychlosti musí být z pravidla nižší než reálné rychlosti přenosové soustavy. Tvarování

provozu musí probíhat vždy v nejpomalejším bodě přenosové soustavy. V případě,že



Obr. 5.1: Příklad HTB struktury

jsou tyto rychlosti vyšší, než přenosová rychlost soustavy, nebude řízení provozu pracovat správně [22].

Níže jsou zobrazeny příkazy, kterými se `HTB` nastavuje.

```
# root trida
tc qdisc add dev mistni root handle 1:0 htb default 0

# pod trida
tc class add dev mistni parent 1:0 classid 1:2 htb rate\
 197000kbit ceil 197000kbit prio 1
```

5.2 Vlastnosti HTB

`HTB` dokáže dělit rychlost do tříd podle uživatelů a zacházet s provozem podle SLA. Struktura by měla být sestavena tak, aby každý uživatel spadal do své třídy. Tím je zajištěno, že uživatel spadající do své třídy nemůže vyčerpat prostředky jiného uživatele, který je v jiné třídě. Dobře sestavená struktura `HTB` již dokáže zamezit vzájemnému ovlivňování uživatelů a zlepšit tak odolnost proti přetížení sítě. V případě vysoké agregace, kdy je zaplněna celá síťová kapacita, `HTB` dělí rychlost rovnoměrně, nebo podle definované priority. Není tedy možné, aby vysoká agregace způsobila kolaps celého telekomunikačního systému, dojde pouze ke zpomalení přenosové rychlosti uživatele [22].

Při konfiguraci musí být přesně definovány rychlosti `rate` a `ceil`, se kterými má program `HTB` pracovat. Mohou nastat případy, při kterých tyto rychlosti nejsou známy. Například při

- sdílení kapacity spoje,
- přenášení provozu přes pomalejší spoj.

Spojem nemusí protékat pouze provoz, který protéká strukturou `HTB`. V případě, že nemůžeme definovat rychlost tohoto provozu, nemůžeme definovat ani zbylou přenosovou rychlost, kterou potřebujeme znát na nastavení řízení provozu sítě. V dnešních sítích je dynamika provozu běžná a není výjimkou, aby provoz procházel našim spojem, a mířil do jiného zařízení, které není za strukturou `HTB`. Například k jinému serveru, nebo k jinému výchozímu bodu sítě.

Druhý bod obsahuje méně časté události v síti. Například pokles přenosové rychlosti bezdrátového spoje z důvodu vnějších vlivů. Nebo změnu v dynamickém směrování z důvodu výpadku části sítě, nebo jenom její údržby. Při těchto událostech také není možné definovat přenosovou rychlost na nastavení struktury `HTB`.

Výše popsané stavy vždy nějakým způsobem sníží přenosovou kapacitu soustavy. Při nižší kapacitě je vyšší pravděpodobnost, že se síť přetíží. Proto funkce struktury `HTB` a její dělení toku je důležité v tento okamžik udržet funkční. Problémem jsou pevně definované rychlosti při nastavení řízení provozu, protože při stavech popsaných v této kapitole není možné tuto rychlost definovat a nastavit ručně. Špatně nastavené rychlosti v programu `HTB` způsobí jeho nefunkčnost.

6 NOVÝ ALGORITMUS NA ŘÍZENÍ PROVOZU

Přínosem této práce je zprovoznění telekomunikačního systému, která bude imunní vůči přetížení. Síťové prvky se proti přetížení brání jednoduchými způsoby, které mají své nedostatky popsány v kap. 4.2. Tyto nedostatky mohou být odstraněni použitím programu `HTB`, ale musíme znát přenosové parametry sítě. Pokud je neznáme, nemůžeme program nastavit. Řešení tohoto nedostatku je popsáno v této kapitole, nový algoritmus nastavuje rychlosti `HTB` dynamicky. Výhodou je nejenom jednodušší nastavení, ale hlavně větší možnost sdílení kapacity spojů a lepší reakce na změny přenosové kapacity sítě. Idea nového algoritmu pochází z principů řízení provozu TCP protokolu, který reguluje počet vyslaných dat do sítě podle naměřeného stavu sítě.

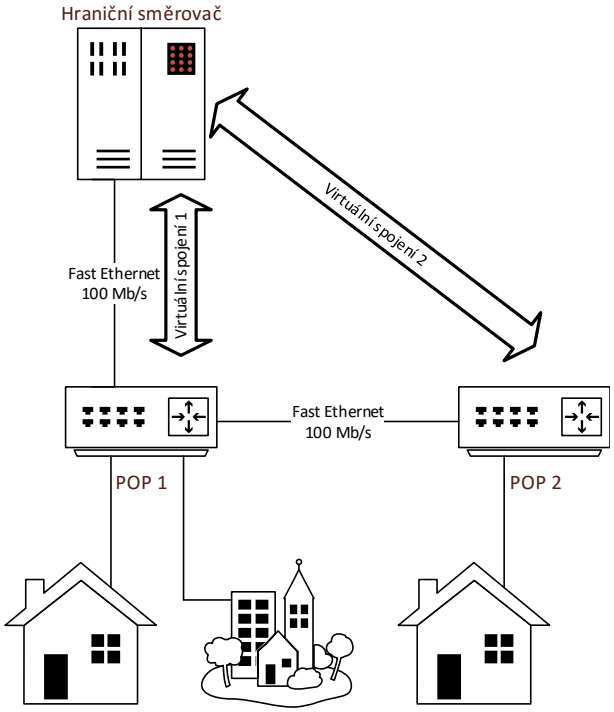
6.1 Princip nového algoritmu

Základní funkcí nového algoritmu je monitorovat stav sítě v reálném čase a při náznaku přetížení zpomalit datový tok v síti tak, aby ke stavu přetížení síťových prvků nedošlo.

Topologie

Pro vyšší výpočetní náročnost není možné algoritmus nasadit na všechny prvky sítě, ale pouze na výkonnější. Hraniční směrovače s dostatečnou výpočetní kapacitou jsou běžně nasazovány na kraje autonomních systémů. Proto i algoritmus uvažuje s nasazením na tyto místa. Je samozřejmě možné síť jednoho autonomního systému rozdělit na více menších oblastí a na hranicích nasadit tento algoritmus. Uvnitř takovéto oblasti se nachází POP. Každé aktivní místo je reprezentováno třídou v programu HTB a podtřídami uživatelů, kteří jsou připojeni do tohoto místa sítě. Toto spojení umožňuje řídit maximální rychlost, která je distribuována do POP pomocí změny rychlosti v programu HTB. Algoritmus uvažuje, že je každý POP připojený přímo k hraničnímu směrovači, to zajisté nemusí být pravidlem. Pokud není POP přímo připojen k hlavní bráně, ale je připojený přes jiný POP, nový algoritmus uvažuje virtuální spojení, mezi sebou a každým POP. Obr. 6.1 zobrazuje příklad možné topologie. Nahoře je uveden hraniční směrovač, který je připojen do zobrazené sítě a do sítě Internet. Na tomto zařízení je spuštěn náš algoritmus. Dále jsou na obrázku připojeny dva přepínače, reprezentující POP. Tyto přepínače jsou propojeny mezi sebou technologií Fast Ethernet, POP 1 je touto technologií připojen i k hraničnímu směrovači. k přepínačům jsou připojeny domácnosti, které reprezentují uživatele sítě.

Velké šipky pojmenované Virtuální spojení zobrazují uvažování algoritmu nad topologií sítě. Je zde patrné, že POP 2 je připojen přes POP 1, ale algoritmus uvažuje



Obr. 6.1: Topologie pro nasazení algoritmu

přímé virtuální spojení mezi svým hostitelem a POP 2. Toto je jedna z nejdůležitějších podstat algoritmu. Díky tomuto uvažování není potřeba brát v potaz topologii sítě, přepínání routingu, kruhovou zálohu a další věci, které znamenají zásah do cesty paketu.

Monitorování sítě

Při přicházejícím stavu přetížení a při plném přetížení jsou změněny základní parametry sítě viz kap. 1.3.4. Převážně odezva sítě nabývá vyšších hodnot a ztrátovost paketů je vyšší. Nový algoritmus monitoruje tyto parametry a využívá je k zjištění aktuálního stavu sítě. Monitorování je realizováno pomocí dotazu ICMP Echo Request, běžně nazývaného jako Ping. Tento dotaz je průměrně každých 300 ms zasílán na každý POP.

Změřená odezva je dále zpracovávána do dvou hodnot. První hodnota je nazvaná jako odezva jmenovitá. Do tohoto čísla se uloží nejnižší naměřená hodnota odezvy. Algoritmus počítá s tím, že v tomto čísle je uložena odezva nezatížené sítě. Druhá

hodnota nazvaná odezva maximální, ukládá se do ní nejvyšší odezva sítě. I při normálních podmínkách v nepřetížené síti se díky shluku dat, nebo ztrátě informace při přenosu může stát, že se požadavek nového algoritmu ztratí, nebo zpozdí. Proto je nejprve vypočítána hodnota průměrná:

$$oPrum = \frac{oPrum \cdot 3 + oNamerena}{4} \tag{6.1}$$

kde `oPrum` reprezentuje průměrnou odezvu a `oNamerena` je současná hodnota odezvy, právě naměřená. Zprůměrování zajistí, aby nahodilá chyba nezpomalila provoz nepřetížené sítě. Pokud je však opožděných odpovědí více, jsou promítnuty do průměrné odezvy. Odezva je měřena v rozsahu 0 ms až 50 ms. Pokud se požadavek nevrátí včas, nebo je ztracen úplně uvažuje algoritmus s maximální hodnotou odezvy. Průměrování času opožděných odpovědí zpomaluje reakční dobu na stav přetížení, je ovšem nutné. Tato hodnota musí být vždy co nejvíce aktuální, proto se neustále snižuje. Tím je způsobeno, že jsou do maximální odezvy promítnuty jen opravdové stavy přetížení a že se po skončení přetížení hodnota přiblíží jmenovité odezvě. Maximální odezva je pak z průměrné odezvy vypočítána vztahem:

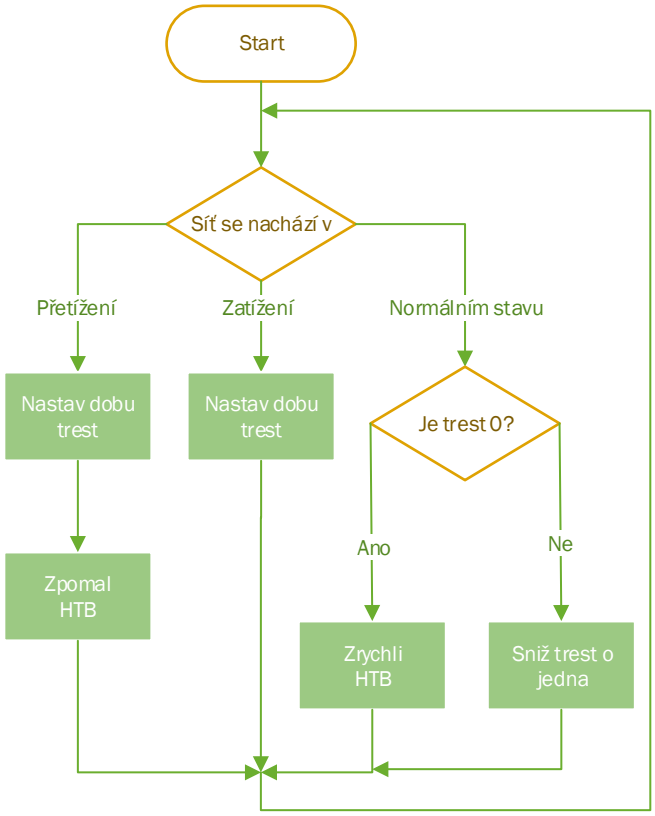
$$oMax = oPrum - oJmenovita \cdot 1.1 \tag{6.2}$$

kde `oMax` je výsledná maximální hodnota pro práci algoritmu a `oJmenovita` je jmenovitá odezva. Ve výpočtu se odečítá o 10 % odezvy jmenovité více. Je to z důvodu kolísání zpoždění, které je v reálné síti úměrné velikosti jmenovité odezvy. Tyto vzorce vznikly analyticky, pro rozpoznání stavu sítě jsou dostačující. Citlivost nového algoritmu na změny v síti je možné nastavit v konfiguračním souboru.

Dynamické nastavení rychlosti

Nový algoritmus je propojen se strukturou `HTB` a může nastavovat rychlosti `rate` a `ceil`. Princip tohoto programu je více popsaný v kap. 5.1. Algoritmus rozeznává tři stavy sítě: přetížení, zatížení a normální stav. V případě detekování přetížení jsou rychlosti třídy HTB reprezentující tento POP sníženy tak, aby nadále k přetížení sítě nedocházelo. Hodnota trest je nastavena na čas z konfiguračního souboru. Při stavu zatížení je odezva sítě mírně zvýšena na běžnou úroveň a nastavená rychlost v programu `HTB` se nemění. Hodnota trest je opět nastavena na definovanou hodnotu. Algoritmus uvažuje, že jsou rychlosti v tento stav nastavený správně. Pokud jsou parametry sítě bez náznaku zatížení, nebo přetížení, začne se odečítat hodnota trest. Odečítání způsobí prodlení před tím, než se začnou rychlosti zase zvyšovat. Tím je předcházeno neustálému měnění rychlostí. V případě, že se doba trest snížila až na nulu, začnou se rychlosti pomalu zvyšovat až do hodnoty nastavené administrátorem. Tím je zajištěno opětovné zvýšení povolené přenosové rychlosti. Snižování

rychlosti v případě detekce přetížení je velmi rychlé, v řádu několika sekund. Zvyšování na počáteční úroveň je naopak pozvolné. Vývojový diagram tohoto rozhodování je uveden na obr. 6.2. Nový algoritmus tak doplňuje nevýhody programu `HTB`. Tento algoritmus je spuštěn paralelně pro každý POP a řízení rychlosti tak probíhá pro každé virtuální spojení.



Obr. 6.2: Vývojový diagram algoritmu

Protože přes jednu fyzickou cestu může probíhat více virtuálních spojení, je nutné zajistit, aby si mezi sebou dělili rychlost rovnoměrně. To je zajištěno tím, že rychlejší virtuální spojení má přísnější politiku snižování rychlosti, než pomalejší virtuální spojení. Díky tomuto řešení se rychlost fyzického spojení rovnoměrně rozdělí mezi virtuální spojení.

6.2 Realizace nového algoritmu

Aby bylo možné algoritmus nějakým způsobem spustit je realizován v programovacím jazyku C++. Zdrojový text byl následně zkompilován pro Linux. Celý vývoj probíhal na operačním systému Debian ve vývojovém prostředí KDevelop. Tento

programovací jazyk byl zvolen z důvodu velké podpory, univerzálnosti a nízké náročnosti na výpočetní výkon při běhu programu. Veškeré programové vybavení spadá do sekce s otevřeným zdrojovým kódem, které je možno zdarma užívat. To znamená, že úprava tohoto programu je dostupnější, protože jsou programy volně ke stažení. Kód je velmi dlouhý (1780 řádků), proto nemá smysl uvádět jej do tištěné práce. Celý program je uveden v elektronické příloze této práce.

6.2.1 Funkce programu

Základem programu je nový algoritmus, který je již popsaný v kap. 6.1. Aby byl program lépe implementovatelný a bylo ho možné lépe ovládat, obsahuje i další funkce, které jsou popsány v této kapitole.

Ověření práv uživatele

Program pracuje s příkazy na řízení algoritmů, které jsou zavedeny v jádře operačního systému, například `iptables`, nebo `tc`. Aby se nestalo, že administrátor bude hledat chybu v oprávněních programu, je zde jednoduše zavedena podmínka, která nedovolí spustit program bez oprávnění superuživatele. V případě špatných oprávnění program vrátí chybu.

```
marek@debian:/gw ./gw
Je nutne byt root!
```

Automatické spojení HTB, iptables a nového algoritmu

Aby byla implementace a nastavení programu co nejjednodušší, je program sestaven tak, aby spojil co nejvíce programů. Pro nastavení celého systému stačí uvést informace o POP a uživatelích sítě do konfiguračních souborů programu. Program si při prvním zapnutí označení provoz klientů pomocí programu `iptables` a sestaví strukturu `HTB`. Do programu `HTB` již není možné zasahovat. Do firewallu je možné zasahovat a dokonce jsou zde předem připravené soubory pro pravidla administrátora. Všechny konfigurační soubory jsou popsány v další kap. 6.2.2.

Ověření vstupních dat

Protože se program nastavuje pouze přes konfigurační soubory, je vhodné zajistit ověření vstupních souborů a vstupního textu. Ověření vstupních souborů je jednoduché a jednoznačné, soubor musí existovat, jinak program skončí chybou.

```
root@debian:/gw# ./gw
Durnet.cz GW 2016
```

```
ERROR: neexistuje soubor /gw/gw.conf
```

Ověření vstupních dat je již daleko složitější, je zde vyšší riziko, že se vyskytne chyba. V takovém případě program může skončit chybou, ale také nemusí. Například při zadání chybné veřejné IP adresy klienta program automaticky opraví adresu za správnou. Rozsah správných adres je uveden v hlavním konfiguračním souboru. V souboru definujícím POP je automaticky odebírána IP adresa, která patří zároveň i klientovi. Tato ochrana je z toho důvodu, že u klienta nemůžeme garantovat dobré parametry spojení. Pokud klient zatíží svůj stroj, může na dotazy nového algoritmu reagovat opožděně a tím zkreslit měření. I přes tyto opatření existuje spousta možných chyb v konfiguračním souboru, které nejsou opatřeny a je nutné, aby administrátor vyplnil soubory správně.

Úspora výkonu

Popisovaný systém je dost náročný na výpočetní výkon a to především ze dvou funkcí. První funkce je monitorování sítě, kdy v krátkém čase odchází velké množství paketů na různé aktivní prvky sítě. Například v síti Durnet.cz je těchto aktivních míst ke dni 9. 5. 2016 přesně 30 [21]. Algoritmus testuje dostupnost sítě v průměru každých 300 ms. To znamená, že za sekundu je odesláno průměrně 100 dotazů ICMP echo request. Tento počet zatěžuje procesor i telekomunikační systém. Program `HTB` je taktéž velmi náročný na výpočetní výkon procesoru. Jeho úloha je neustále přepočítávat rychlosti uživatelů a rovnoměrně mezi ně dělit síťové prostředky.

Úspora výkonu uvažuje, že v jednom čase nemusí být všichni uživatelé sítě aktivní. Proto sleduje provoz uživatelů a rozděluje je na aktivní a neaktivní. Pokud je uživatel neaktivní, je jeho třída v programu `HTB` smazána. Pokud není na POP ani jeden klient aktivní, vypne se i monitorování virtuální cesty k tomuto POP a nový algoritmus. V případě, že některý z uživatelů začne přenášet data, během pár sekund se označí jako aktivní i s nadřazeným POP. Tato funkce se zapíná a vypíná v konfiguračním souboru, proměnná je nazvána jako `ECONOMYHTB`. Úspora výkonu je odvinutá od poměru neaktivních a aktivních uživatelů.

QoS

Nový program zajisté implementuje i QoS. Na rozlišení provozu je používáno pole TOS dle doporučení RFC4594. Z výpočetního hlediska není možné obsluhovat všechny definované fronty v tomto doporučení. Z toho důvodu se rozlišují jen dva typy provozů: prioritní a normální. Na doporučení je pak navázáno tak, že best effort (TOS = 0) a Low-Priority data (TOS = 2) je označen jako normální a ostatní hodnoty jako prioritní. Označení TOS na paketech, které přichází ze sítě Internet je ve výchozím

nastavení přijímáno. Administrátor však může nastavit vlastní pravidla v konfiguračních souborech firewallu. Příkaz

```
iptables -t mangle -i gateway -j DSCP --set-dscp-class CS0
```

označí veškerý přicházející provoz jako normální, dle doporučení RFC4594 jako best effort. V těchto souborech může upřednostnit i další provoz, například příkazy

```
iptables -t mangle -i gateway -s 123.123.123.123 -j DSCP\
  --set-dscp-class CS4
iptables -t mangle -i mistni -d 123.123.123.123 -j DSCP\
  --set-dscp-class CS4
```

upřednostní provoz z adresy a na adresu 123.123.123.123. V síti Durnet.cz jsou takto označeny servery internetové televize [21]. Výsledkem je, že provoz televize není omezen stahováním velkého objemu dat. Tímto je zajištěna dobrá QoE uživatelů sítě.

Nemá cenu uvádět veškerý provoz klienta jako prioritní, protože označení v poli TOS nedokáže ovlivnit dělení provozu mezi uživateli sítě. Pracuje pouze v třídě uživatele a pouze s provozem uživatele. Tímto je také zajištěna bezpečnost a rovnoměrné rozdělení provozu mezi uživatele i případě podpory QoS. Nerovnoměrné rozdělení rychlostí mezi uživatele podle SLA je uvedeno níže v kap. 6.2.2 popisující konfiguraci programu.

6.2.2 Nastavení programu

Nastavení programu je realizováno pouze pomocí konfiguračních souborů v textovém formátu. Všechny soubory jsou umístěny ve složce `/gw/` v kořenu operačního systému. Struktura souborů je následující:

```
root@salas:/# find /gw/*
/gw/aps.conf
/gw/gw
/gw/firewall
/gw/firewall/pre.conf
/gw/firewall/dynamic.conf
/gw/firewall/post.conf
/gw/gw.conf
/gw/klienti.conf
/gw/commands.txt
root@salas:/gw#
```

Hlavní konfigurační soubor `/gw/gw.conf` obsahuje základní nastavení programu. Jeho obsah je uveden v příloze D. Jsou zde definovány například síťové rozhraní, maximální rychlost průtoku dat, použité IP adresy a další.

Konfigurační soubor `/gw/aps.conf` definuje seznam míst POP uvnitř oblasti. Ke každému zde uvedenému řádku je vytvořena jedna instance nového algoritmu. Struktura souboru je pevně definovaná. První řádek obsahuje názvy hodnot. Další řádky obsahují identifikační údaje každého POP. Je zde definované unikátní ID, volitelný název bez diakritiky, maximální povolená rychlost ve směru stahování a odesílání. Na uvedené IP adresy nový algoritmus zasílá ICMP požadavky a tak zjišťuje zatížení virtuálního spojení ke každému POP.

```
id;jmeno;maxDown;MaxUp;ip1,ip2,ip3,...,ipx
29;sklenar;110;50;172.20.13.242,172.20.13.241,172.20.13.240
27;gala;90;30;172.20.11.247,172.20.11.248,172.20.11.246
26;breznice;110;50;172.20.11.242,172.20.11.241,172.20.11.240
```

V Síti Durnet.cz je tento soubor plněn z administrační databáze, která tyto informace obsahuje.

Soubor `/gw/klienti.conf` je obdobný, ovšem obsahuje informace o uživatelích sítě.

```
id;idAp;jmeno;ip;ipVerejna;maxDown;maxUp;netmap;priorityIps;\
  garantSpeed
1004;23;marek-dubrava;172.20.2.31;79.170.248.140;60;25;1;\
  162.249.73.95,195.28.94.66;0
1006;6;radek-jurik;172.20.1.6;79.170.252.201;20;5;1;;0
7067;28;firma-s-r-o;79.170.252.194;79.170.252.194;25;25;1;;1
```

Nově je v souboru `idAp`, které reprezentuje identifikační číslo POP, ke kterém je klient připojen. Dále jsou zde uvedeny jeho IP adresy, jedna z nich reprezentuje privátní IP adresu v síti Durnet.cz a druhá veřejnou IP adresu, pod kterou má klient vystupovat. V takovémto případě je zde realizován překlad adres, který sdílí veřejnou IP adresu, mezi více klientů. V případě kdy chce mít klient v síti veřejnou IP adresu, jsou tyto adresy stejné a v konfiguračním souboru firewallu se musí přidat výjimka, nebo popřípadě vypnout funkci překladu adres. Jsou zde také uvedeny maximální rychlosti klienta, které mohou být dynamickou změnou rychlosti sníženy. S rychlostmi také souvisí poslední položka `garantSpeed`, která udává vyšší prioritu. Vyšší prioritu je zde možno zapnout nastavením na hodnotu `1`. Provoz prioritního klienta je upřednostněn před ostatními. V případě, že prioritní klient svou kapacitu nevyužívá, mohou ji využívat ostatní klienti, tím je zajištěna maximální možnost agregace síťových prvků. Upřednostněny jsou většinou firemní přípojky s garantovanou kapacitou. Dalším důležitým parametrem je hodnota `priorityIps`, jedná se

o QoS. Adresy uvedeny v tomto poli jsou v rámci provozu uživatele upřednostněny před ostatním provozem. Provoz z těchto adres je nejprve označen vyšší prioritou podle RFC4594 v poli TOS příkazem:

```
iptables -t mangle -A K1004 -s 162.249.73.95 -j DSCP\
  --set-dscp-class CS4
```

dále pak spadá do prioritní třídy, kde je upřednostněn před ostatním provozem uživatele.

Ve složce `/gw/firewall` jsou umístěny konfigurační soubory firewallu. Soubor `/gw/firewall/pre.conf` je aplikován před spuštěním nového algoritmu, ten zapíše do firewallu vlastní označení klientů a pak program spustí další dva soubory ze složky. V těchto souborech jsou umístěny příkazy například na překlad adres, na blokování klientů, kteří nemají oprávnění komunikovat v síti, také jsou zde uvedeny příkazy zajišťující bezpečnost zařízení a další.

Soubor `/gw/gw` je binární soubor vlastního programu. Soubor `/gw/commands.txt` je výstup příkazů, které jsou zasílány do jádra, standardně se do něj nic nezapisuje, slouží pouze pro odladění funkce programu. Většinou obsahuje příkazy do `iptables` a `tc`.

6.2.3 Výstup programu

Aby měl administrátor přehled nad stavem algoritmu, jsou informace zpracovávané programem zapisovány do souborů. Jelikož program zapisuje informace každých pět sekund, mohl by zatížit, nebo poškodit disk zařízení. Z toho důvodu jsou soubory zapisovány do operační paměti počítače. V Linuxu se takové úložiště nazývají jako `tmpfs`. Jejich přípojné cesty můžeme jednoduše zjistit pomocí příkazu:

```
root@salas:/# df -hl | grep tmpfs
tmpfs                   385M  776K  385M   1\% /run
```

Dva soubory byly uloženy na umístění `/run/gw/aps.out` a `/run/gw/klienti.out`. Jeden z nich zobrazuje stav POP a druhý zobrazuje stav klientů. Protože je nepohodlné připojovat se k směrovači pomocí terminálu a sledovat textové soubory o několika stovkách řádků, byl vytvořen další program v programovacím jazyku Hypertext Preprocessor (PHP), který tyto soubory vyčítá a odesílá administrátorovi pomocí Hypertext Transfer Protocol (HTTP) protokolu do jeho webového prohlížeče. Výpočetní výkon tohoto programu a webového serveru `apache2` je ve srovnání s hlavním programem minimální. Program v jazyku PHP i se soubory určujícími styl jsou přiloženy v elektronické příloze této práce. Protože se data neustále mění, je celá stránka jednou za pět sekund obnovena, tím je zajištěno, že administrátor uvidí vždy aktuální data.

Program má dvě stránky, na jedné zobrazuje shrnuté informace POP, na druhé zobrazuje informace o klientech. Kromě nadpisu a drobných informacích o stavu systému je hlavní dominantou vždy tabulka zobrazující stav.

informace o POP

V příloze E.1 jsou zobrazeny informace o virtuálních spojeních na POP. Z tabulky jsou graficky odebrány některé řádky, aby byla zmenšena. Rozdělení tabulky je zobrazeno červenou čarou. Barevné rozlišení pomáhá administrátorovi rychleji se orientovat v tabulce, která se neustále mění. Na první pohled je patrný rozdíl mezi aktivními a neaktivními místy. V případě detekce zatížení, nebo přetížení v síti se hodnota odezvy podbarví do červené.

Identifikační číslo je stejné jako v konfiguračním souboru a je použito jako jednoznačná identifikace POP. Ve výpisu je použito také jako odkaz, ve kterém se přechází na stránku s výpisem stavu klientů. Odezva jmenovitá a maximální je stejná jako v popisu algoritmu v kap. 6.1. Rychlost stahování označená ceil down reprezentuje parametr `ceil` a rychlost označená rate down parametr `rate` programu `HTB`. Hodnoty s označením up jsou použity pro rychlost odesílání. V posledním sloupci je zobrazena rychlost stahování každého virtuálního spojení. Poslední řádek tabulky uvádí součty. Součet rychlostí označených `rate` nemůže přesáhnout rychlosti linky, jinak nebude program správně fungovat.

Informace o klientech

V příloze F.1 je zobrazen výstup programu zobrazující informace o klientech na POP s identifikačním číslem 19. Jsou zde uvedeny rychlosti podobné jako v přehledu POP, ovšem tady se jedná přímo o rychlosti uživatele sítě. Aktuální rychlost je zobrazena pouze ve směru stahování.

6.2.4 Návod na zprovoznění nového programu

Pro zprovoznění programu je vhodný operační systém Debian, nebo Ubuntu Server LTS [18]. Počítač, na kterém program poběží musí mít dvě síťové karty, jednou bude připojen do místní sítě a druhou do sítě Internet. Operační systém Linux nemá ve výchozím nastavení zapnuto přeposílání paketů. Aby systémem protékal datový provoz, musíme tuto funkci povolit následujícím příkazem.

```
sysctl -w net.ipv4.ip_forward=1
```

Po restartu zařízení je nutné tento příkaz zadat znova, proto je vhodné jej spouštět po startu systému, například zadáním příkazu do programu `/etc/rc.local`.

Program je binární soubor, ke své funkci však využívá další programy. Pro funkčnost programu je nezbytné mít v systému programy `iptables`, `tc`, `ulimit`, `fping`, `apache2` a `php5`. První tři programy jsou běžnou součástí jádra systému Linux. Programy `fping`, `apache2` a `php5` nejsou součástí standardní instalace a musí se doinstalovat.

```
sudo apt-get install fping apache2 php5
```

Příkaz nainstaluje chybějící programy z repozitářů. `Fping` se používá na zjištění odezvy sítě. Webový server je použit na zobrazení stavu programu.

Dalším krokem je vytvoření adresáře programu příkazem `sudo mkdir /gw/`. Do tohoto adresáře je nutné zkopírovat soubory z přílohy této práce. Tímto jsou soubory programu připraveny ke konfiguraci. Program na výstup do webového prostředí je nutné zkopírovat do adresáře webového serveru, nejčastěji `/var/www/html`. Tento program již nepotřebuje žádné další nastavení.

Dalším krokem je nastavení statických adres na síťových rozhraních. Operační systém Linux má připravený konfigurační soubor `/etc/network/interfaces`. Změny provedené v tomto souboru se aplikují až po restartu systému.

Nyní je nutné provést konfiguraci programu v hlavním konfiguračním souboru `/gw/gw.conf`. Veškeré proměnné jsou již připraveny, proto stačí jen měnit jejich hodnoty, nebo měnit zakomentování řádku znakem `#`. Jako první je doporučeno zapnout výstup do textového souboru `COMMANDTOTXT=yes`, všechny příkazy tak budou postupně uloženy do souboru a administrátor může kontrolovat jejich správnost, nebo je všechny z tohoto souboru spustit a sledovat průběh. Dále je nutné nastavit správná jména síťových rozhraní, rychlost linky a veřejné IP adresy. Proměnná `PUBLICIPS` obsahuje všechny veřejné adresy, se kterými bude směrovač pracovat, nebo které může mít směrovač na svém síťovém rozhraní. Proměnná `PUBLICSHAREDIPS` obsahuje pole IP adres, na které budou překládány klienti funkcí Network address translation (NAT). Tyto adresy musí být přímo na rozhraní sestavovaného směrovače, tudíž i v souboru `/etc/network/interfaces`. Další nastavení je již volitelné a dají se jím ovlivnit převážně funkce nového algoritmu. Dalším krokem je naplnění souborů klientů a míst POP. Také nastavení statických pravidel ve složce `/gw/firewall/` je nutné zkontrolovat. Nastavení firewallu je již plně na administrátorovi, takže nebude dále rozvíjeno. Nyní je možné program spustit příkazem `/gw/gw`.

7 PRAKTICKÉ NASAZENÍ

V této práci je teoreticky popsán nový algoritmus, který odstraňuje nevýhody programu `HTB`. Aby bylo možné tento algoritmus spustit, je implementován do programu, viz kap. 6. Tato kapitola zobrazuje výsledky z reálné činnosti programu při stavu přetížení v síti. Při měření je kladen důraz na základní parametry sítě, které úzce souvisí s QoE zákazníků.

7.1 Topologie pro měření

V příloze G.1 je zobrazena topologie sítě se zeleně zvýrazněnou trasou. Tato trasa, vedoucí přes více míst reprezentuje jedno virtuální spojení nového algoritmu tzn. i jednu mateřskou třídu v programu `HTB`. Na severním konci této trasy je připojen hraniční směrovač s možností spuštění nového algoritmu. Na jižní straně jsou připojeni čtyři zákazníci, první bude generovat provoz a ostatní budou použiti na sledování dostupnosti sítě. Parametry sítě jsou měřeny z hraničního směrovače pomocí skriptů uvedených níže. Na uvedené trase, která je zvýrazněna v příloze, nebyl v době měření žádný jiný provoz. Spoje jsou určeny pro zálohu v případě výpadku. Přesnější výpis použitých technologií je v tab. 7.1. Radiové spoje i přepínače a směrovače po zvýrazněné trase jsou nastaveny tak, aby měly všechny standardizované funkce na ochranu proti přetížení zapnuty. Výše zobrazené spoje propojují zařízení RouterBoard 2011. Podrobnější výpis, jaké algoritmy tyto zařízení obsahují je v kap. 4.

Tab. 7.1: Tabulka spojů

Lokalita 1	Lokalita 2	vzdálenost	Typ zařízení	Anténa
Salaš	Kelníky	6414 m	Rocket M5	650 mm 30 dBi
Kelníky	Velký Ořechov	1882 m	PowerBeam M5	300 mm 22 dBi
Velký Ořechov	Dobrkovice	1813 m	NanoBridge M5	400 mm 25 dBi

7.2 Generování přetížení

Cílem je pozorovat vlastnosti sítě v závislosti na funkci nového algoritmu. Zatížení sítě bude generovat jeden ze zákazníků na jižní straně sítě stahováním souborů ze serveru HTTP. Server je umístěn na hraničním směrovači, na kterém je možnost spustit nový algoritmus. Klient je standardní stolní počítač připojený do sítě s operačním systémem Ubuntu (Linux). Server je hraniční směrovač s operačním

systémem Linux Ubuntu server LTS a webovým serverem `apache2`. Pro měření je vytvořen soubor s náhodným obsahem o velikosti 10 MB. Pro vytvoření souboru byl použit program pro bitové kopírování `dd`.

```
dd if=/dev/urandom of=10M bs=1M count=10
```

Program čte náhodně generovaná data ze zdroje `if=/dev/urandom` a ukládá je do souboru `10M`. Velikost je nastavena parametry `bs=1M count=10`, které znamenají, že se do souboru desetkrát zkopíruje blok o velikosti 1 MB. Zákazník si od serveru vyžádá soubory pomocí skriptu v jazyce `BASH` (Linuxový terminál).

```
wget 172.20.12.240/dp/10M -O /dev/null > /dev/null 2>&1 &
wget 172.20.12.240/dp/10M -O /dev/null > /dev/null 2>&1 &
wget 172.20.12.240/dp/10M -O /dev/null > /dev/null 2>&1 &
wget 172.20.12.240/dp/10M -O /dev/null > /dev/null 2>&1 &
wget 172.20.12.240/dp/10M -O /dev/null > /dev/null 2>&1 &
```

Příkaz `wget` stáhne soubor pomocí protokolu HTTP z adresy `172.20.12.240/dp/10M`. Parametr `-O /dev/null` způsobí zahození stažených dat a přesměrování výstupu `> /dev/null 2>&1` zakáže vypisování jakýchkoli informací na terminál. Aby se na stažení prvního souboru nečekalo a hned se začal stahovat další, je na konci řádku znak `&`. Soubor, který obsahuje tyto příkazy, spuštěním vyvolá současně pět stahování souboru `10M` ze serveru (hraničního směrovače). Stahování více souborů současně umožní nasimulovat přetížení sítě.

7.3 Zachycování parametrů sítě

Jevy vzniklé při stahování souborů jsou zachycovány skriptem spuštěným na hraničním směrovači. Skript zachytává odezvu, aktuální přenosovou rychlost a dynamicky nastavenou rychlost nového algoritmu v závislosti na čase. Struktura měřícího programu je vytvořena tak, aby běželo jedno hlavní vlákno a to dále spouštělo v definovaných časech měření parametrů sítě. Hlavní vlákno tak každých 500 ms spustí měření všech uvedených parametrů. Tento hlavní skript je uveden v příloze H. Skript spouští další pod-procesy, které vyčítají měřenou hodnotu a ukládají ji do výstupního souboru. Jako v případě stahování souborů je u těchto pod-procesů na konci řádku parametr `&`, který zajistí, aby se nečekalo na dokončení procesu, ale okamžitě se pokračovalo spuštěním další úlohy.

Odezva se měří ze směrovače na tři klienty, kteří nestahují soubory ze serveru. Tato odezva tedy reprezentuje odezvu páteřní sítě od jádra sítě (hraničního směrovače) k zákazníkovi. Zhoršení odezvy negativně ovlivní služby v reálném čase všech

zákazníků, kteří jsou na tomto POP připojeni. Výsledky ze tří klientů jsou zprůměrovány do jediného, aby byly hodnoty co nejpřesnější a výsledný graf přehledný. Skript měřící odezvu je uveden v příloze I. Pro jednoduchost práce s textem byl zvolen programovací jazyk PHP. Skript využívá program `fping` a pomocí PHP funkcí se z výsledku vybírá pouze číslo odezvy. Toto číslo se pak vypíše na standardní výstup terminálu, který je parametrem např. `» $SODEZVA1` v hlavním programu přesměrován do výstupního souboru.

Reálná přenosová rychlost virtuálního spojení je vyčítána z programu `iptables`. Program spuštěný na směrovači vytvoří pro každého klienta vlastní `chain` a do něho směřuje jeho provoz. Poté tedy stačí vyčítat množství dat přímo u klienta, který stahuje soubory. Rozdílem předchozích hodnot můžeme zjistit, kolik dat bylo za 500 ms přeneseno. Program v jazyku PHP vyčítající tyto data je uveden v příloze J. Program vyčítá data pro směr odesílání i přijímání. V programu je použit příkaz

```
iptables -v -t mangle -S PREROUTING | grep K4\
    | head -1 > rychlost1.log.txt
```

který vyčítá data z programu `iptables`, konkrétně z tabulky `mangle` a chainu `PREROUTING`. Parametr `-v` zajišťuje podrobnější výpis statistik, mezi statistikami je pak i počet přenesených bajtů. Druhý program grep vybere řádek našeho klienta, který má označení K4. Poslední program `head` vybere pouze první řádek z výpisu, protože statistiky obsahovaly i další informace. Zbytek programu v PHP již text upraví tak, aby byla uložena jen hodnota přenesených bajtů.

Dynamicky nastavená rychlost nového algoritmu je vyčítána přímo z výstupu programu ze souboru `/run/gw/aps.out`. Program, který filtruje text a zobrazuje pouze požadovanou rychlost, je zobrazen v příloze K.

```
cat /run/gw/aps.out | grep POPid
```

Příkaz na získání dat ze souboru do měřicího programu obsahuje příkaz `cat`, který vypíše obsah souboru a program `grep`, který zobrazí pouze řádek, na kterém se nachází sledované virtuální spojení. Další část programu text přefiltruje a vypíše pouze rychlost `ceil down`.

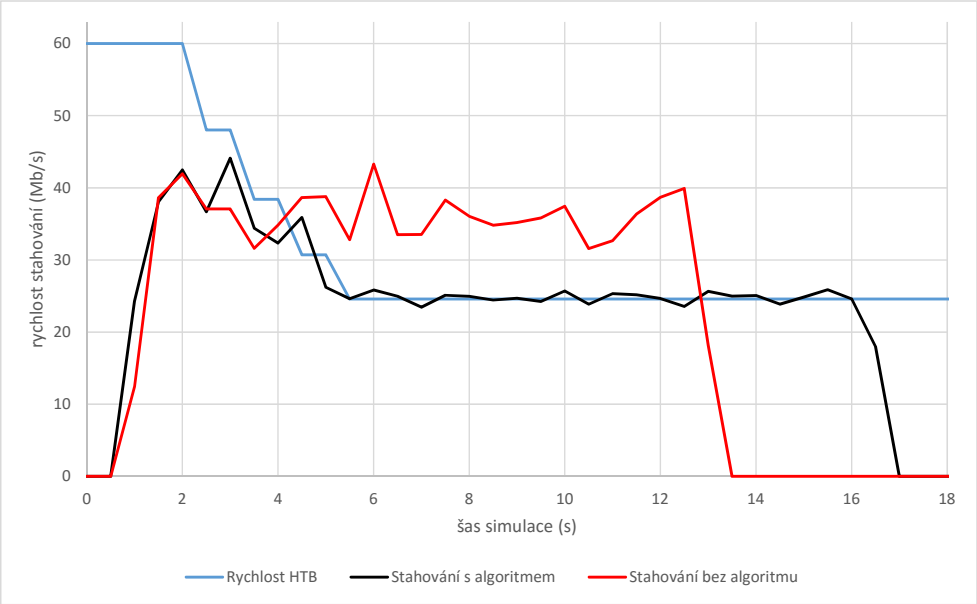
Po spuštění popsaného měření se vytvoří několik textových souborů, které obsahují naměřená data. Tato data poté stačí zpracovat v některém z tabulkových procesorů, např. v Libre Office Calc, nebo Microsoft Office Excel. Aktuální přenosová rychlost není uvedena přímo, proto je nutné její výpočet. V datech je uvedena absolutní hodnota přenesených bajtů, která je odečítána každých 500 ms. Reálnou přenosovou rychlost tak můžeme vypočítat vztahem

$$R = \frac{(X_{i-1} - X_i) \cdot 2 \cdot 8}{10^6} \text{ [Mb/s]} \tag{7.1}$$

kde X_i označuje naměřenou hodnotu, X_{i-1} označuje předchozí naměřenou hodnotu a R označuje reálnou přenosovou rychlost v Mb/s. Soubor, který byl použit pro upravení dat a vytvoření grafů je uložen v elektronické příloze.

7.4 Výsledky

Naměřená zpracovaná data jsou zobrazena do dvou grafů. Každý graf porovnává parametry sítě bez a s novým algoritmem. Graf 7.1 zobrazuje porovnání přenosových rychlostí při nasazení algoritmu. Červený průběh zobrazuje přenosovou rychlost virtuálního spojení bez nového algoritmu. Modrý průběh zobrazuje dynamicky nastavenou přenosovou rychlost nového algoritmu. Černý průběh zobrazuje reálnou rychlost toku dat s implementovaným novým algoritmem.

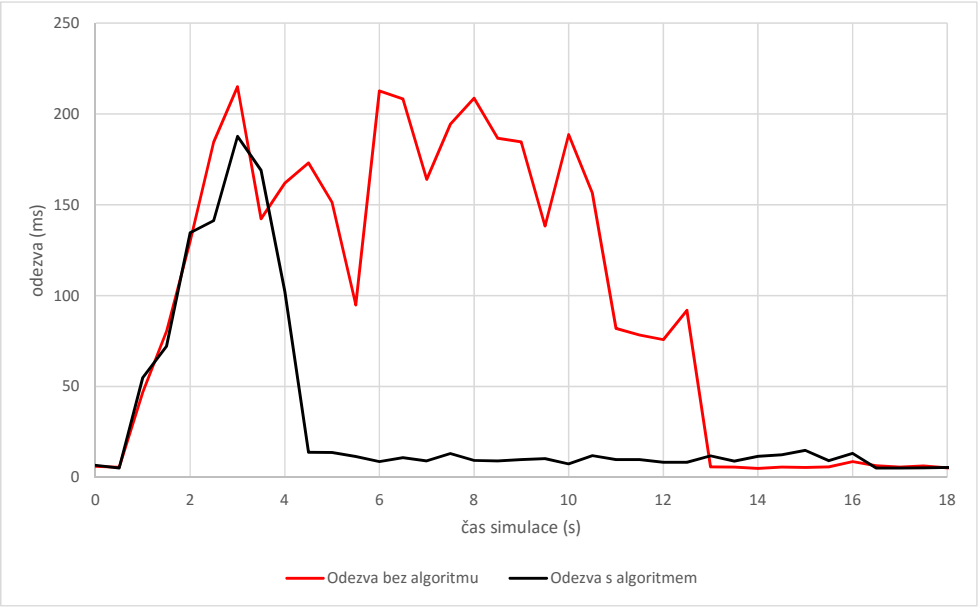


Obr. 7.1: Porovnání rychlosti stahování při nasazení nového algoritmu

Graf 7.2 porovnává odezvu virtuálního spojení při nasazení algoritmu. Červený průběh zobrazuje odezvu bez nového algoritmu, tj. s vypnutým programem. Černý průběh zobrazuje odezvu s nasazeným novým algoritmem, tj. se spuštěným programem na hraničním směrovači.

Z grafů vyplývá, že při spuštění stahování se páteřní sít přetížila a zvýšila svoji odezvu nad hodnotu 150 ms. Podle doporučení ITU G.114 by zpoždění pro Voice over Internet Protocol (VoIP) nemělo přesáhnout 150 ms [4]. Samotná zařízení se standardizovanými algoritmy se se zatížením nevypořádala. Síť se tedy nachází v přetíženém stavu. Nový algoritmus tento stav rozpoznal pomocí vlastního měření a začal

postupně snižovat maximální dosažitelnou rychlost v programu `HTB`. Program `HTB` začal postupně snižovat rychlost virtuálního spojení a tím i rychlost stahujícího zá-



Obr. 7.2: Porovnání odezvy při nasazení nového algoritmu

kazníka. Toto umělé snížení rychlosti způsobilo uvolnění kapacit spojů a síť se tak dostala ze stavu přetížení. Stav sítě se promítl do poklesu doby odezvy. Hodnota zpoždění v závislosti na zatížení roste exponenciálně [4]. Proto je velký rozdíl mezi přetíženou sítí, zatíženou sítí a nezatíženou sítí. Dle měření je odezva přetížené sítě v průměru 172 ms, zatížené sítě 11 ms a nezatížené sítě 6 ms. Reakce algoritmu na tyto hodnoty je možné upravit v konfiguračním souboru. Je tedy možné pro sítě jiného charakteru nastavit jinou průměrnou odezvu v zatížení.

Po návratu na nezatížený stav sítě, algoritmus začne dynamicky nastavenou rychlost v programu `HTB` zase zvyšovat. Rychlost tohoto růstu je možné plně ovlivnit v konfiguračních souborech. Výsledný průběh by pak zobrazoval vzrůst rychlosti o definovaný krok v definovaném čase. Je plně na administrátorovi, aby uzpůsobil rychlost růstu dle požadavků na síť.

8 ZÁVĚR

V teoretickém úvodu byla představena síť ISP i s popisem reálné sítě Durnet.cz. Po představení principu řízení datového toku v síti je zobrazen klasický případ přetížení síťového prvku. Tento problém řeší popsané metody, které spolu s teorií front reprezentují téměř všechny možnosti určené k modelování datového toku.

Další část je zaměřena na zařízení, které jsou nasazeny v síti Durnet.cz. Výrobci zařízení jsou krátce představeni. U každého modelu jsou vypsány jeho funkce související s řízením datového toku. Hlavním účelem bylo zobrazení podpory ECN v síťových prvcích. Jelikož je ECN podporováno pouze ve virtuální frontě v operačním systému Linux, nebylo nutné ani vytváření přehledu. Na druhou stranu se ukázalo, že výrobci přichází s vlastními funkcemi určenými k řízení datového toku. U klientských zařízení byla zobrazena plná podpora ECN na nových operačních systémech Windows a OS X (Apple). První měření zobrazuje jak velký podíl paketů v síti je kompatibilní s ECN protokolem. Měření probíhalo na síti Durnet.cz jeden týden a zachycené poznatky jsou přehledně zobrazeny na konci kapitoly. V teoretické části je popsána vysoká podpora v klientských zařízeních, ovšem měření zobrazuje opak. Pouze 0,854 % paketů je kompatibilních s explicitní signalizací přetížení.

Další kapitola diskutuje problémy standardizovaných metod a zobrazuje proč tyto metody nejsou na řízení provozu v ISP síti dostačující. Jejich velká slabina je především v nemožnosti rozeznat zákazníka sítě a pracovat s jeho daty podle SLA. Proto je nutné v síti nasadit i další metody.

Problém velké agregace a podpory SLA řeší program HTB. Ten dokáže dělit provoz podle definovaných parametrů. Při teoretickém zpracování popisu HTB byl prezentován nedostatek, který znemožní nasazení programu v případě, kdy není možné přesně definovat přenosovou rychlost spoje. Dynamické nastavení rychlosti do konfigurace HTB řeší nový algoritmus. Pomocí sledování stavu sítě se omezuje maximální provoz sítí tak, aby se síť nenacházela ve stavu přetížení. Algoritmus je realizován programem, který je v elektronické příloze této práce. Implementace algoritmu neomezuje topologie sítě, naopak dynamicky reaguje na její změny.

Nový program spolu s standardizovanými metodami proti přetížení byl nasazen do ISP sítě Durnet.cz. Síť byla testována na odolnost proti přetížení. Výsledky tohoto testování jsou reprezentovány na konci práce. Vyplývá z nich, že standardizované metody jsou nedostačující, protože s nimi byla odezva páteřní sítě při měření vyšší, než 150 ms. Tato doba je vyšší, než v doporučení ITU G.114, tedy síť by nebyla schopná přenášet VoIP. Při zapnutí nového algoritmu je síť chráněna lépe. Po detekci přetížení nový algoritmus nastavil HTB tak, aby byla rychlost přenášených dat omezena. Po omezení rychlosti se síť dostala ze stavu přetížení, její odezva klesla na průměrnou hodnotu 10 ms.

LITERATURA

[1] CISCO SYSTEMS, INC. Cisco Visual Networking Index: Forecast and Methodology, 2014-2019 White Paper. *White Paper* [online]. 2015, 2015-11-16, 2015: 1-14 [cit. 2015-11-16]. Dostupné z: `http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html`

[2] TRULOVE, James. *Sítě LAN: hardware, instalace a zapojení.* 1. vyd. Praha: Grada, 2009, 384 s. Profesionál. ISBN 978-80-247-2098-2.

[3] KÁLLAY, Fedor a Peter PENIAK. *Počítačové sítě a jejich aplikace: LAN / MAN / WAN.* 2. aktualiz. vyd. Praha: Grada, 2003, 356 s. ISBN 80-247-0545-1.

[4] KOTON, Jaroslav. *Moderní síťové technologie.* Brno, 2014. Skripta. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ.

[5] GRIMM, Christian a SCHLÜCHTERMANN GEORG. *IP traffic theory and performance.* Heidelberg: Springer, 2008, xiv, 48 s. signals and communication technology. ISBN 978-3-540-70603-8.

[6] RAMAMURTHY, Byrav, George N ROUSKAS a Krishna M SIVALINGAM. *Next-generation internet: architectures and protocols.* New York: Cambridge University Press, 2011, xxiii, 409 p. ISBN 0521113687.

[7] SHINDER, Debra Littlejohn. *Počítačové sítě: nepostradatelná prírucka k pochopení sítové teorie, implementace a vnitrních funkcí [sic].* Praha: SoftPress, c2003, 752 s. Cisco systems. ISBN 80-86497-55-0.

[8] BERKOWITZ, Howard C. *Building service provider networks.* New York, N.Y.: Wiley, c2002, xxi, 570 p. ISBN 0471099228.

[9] Doporučení RFC 6633: *Deprecation of ICMP Source Quench Messages* Únor 2012, dostupné online: `http://tools.ietf.org/search/rfc6633`, cit. 2015-11-20.

[10] Doporučení RFC 792: *INTERNET CONTROL MESSAGE PROTOCOL* Září 1981, dostupné online: `http://tools.ietf.org/search/rfc792`, cit. 2015-11-20.

[11] Doporučení RFC 2285: *Benchmarking Terminology for LAN Switching Devices* Únor 1998, dostupné online: `http://tools.ietf.org/search/rfc2285`, cit. 2015-11-20.

[12] Doporučení RFC 3168: *The Addition of Explicit Congestion Notification* Září 2001, dostupné online: `http://tools.ietf.org/search/rfc3168`, cit. 2015-11-20.

[13] Doporučení RFC 6040: *Tunnelling of Explicit Congestion Notification* Listopad 2010, dostupné online: `http://tools.ietf.org/search/rfc6040`, cit. 2015-11-20.

[14] *MikroTik Routers and Wireless* [online]. Lotyšsko: MikroTik, 1996 [cit. 2015-11-29]. Dostupné z: `www.mikrotik.com`

[15] *Ubiquiti Networks* [online]. Kalifornie: Ubiquiti Networks, 2005 [cit. 2015-11-29]. Dostupné z: `www.ubnt.com`

[16] *ALCOMA a.s.* [online]. Česká republika: ALCOMA a.s., 1993 [cit. 2015-11-29]. Dostupné z: `www.alcoma.cz`

[17] *Cisco Systems, Inc.* [online]. San Jose: Cisco Corporate Overview, 1984 [cit. 2015-11-29]. Dostupné z: `www.cisco.com`

[18] *Canonical Ltd.* [online]. United Kingdom: Canonical | The company behind Ubuntu, 2004 [cit. 2015-11-29]. Dostupné z: `www.ubuntu.com`

[19] *Microsoft Corporation* [online]. Redmond: Oficiální domovská stránka Microsoft, 1975 [cit. 2015-11-29]. Dostupné z: `www.microsoft.com`

[20] *Apple Inc.* [online]. Kalifornie: Apple, 1976 [cit. 2015-11-29]. Dostupné z: `www.apple.com`

[21] *Durnet.cz připojení k internetu* [online]. Dobrkovice, 2016 [cit. 2016-03-08]. Dostupné z: http://durnet.cz/

[22] *HTB Linux queuing discipline manual - user guide* [online]. 2002 [cit. 2016-04-27]. Dostupné z: http://luxik.cdi.cz/ devik/qos/htb/manual/userg.htm

[23] GEOFF HUSTON. *ISP survival guide: strategies for running a competitive ISP.* [Nachdr.]. New York [u.a.]: Wiley, 1999. ISBN 0471314994.

[24] HECKMANN, Oliver. *The competitive Internet service provider: network architecture, interconnection, traffic engineering and network design.* Hoboken, NJ: J. Wiley, c2006, xxvii, 370 p. ISBN 9780470012932.

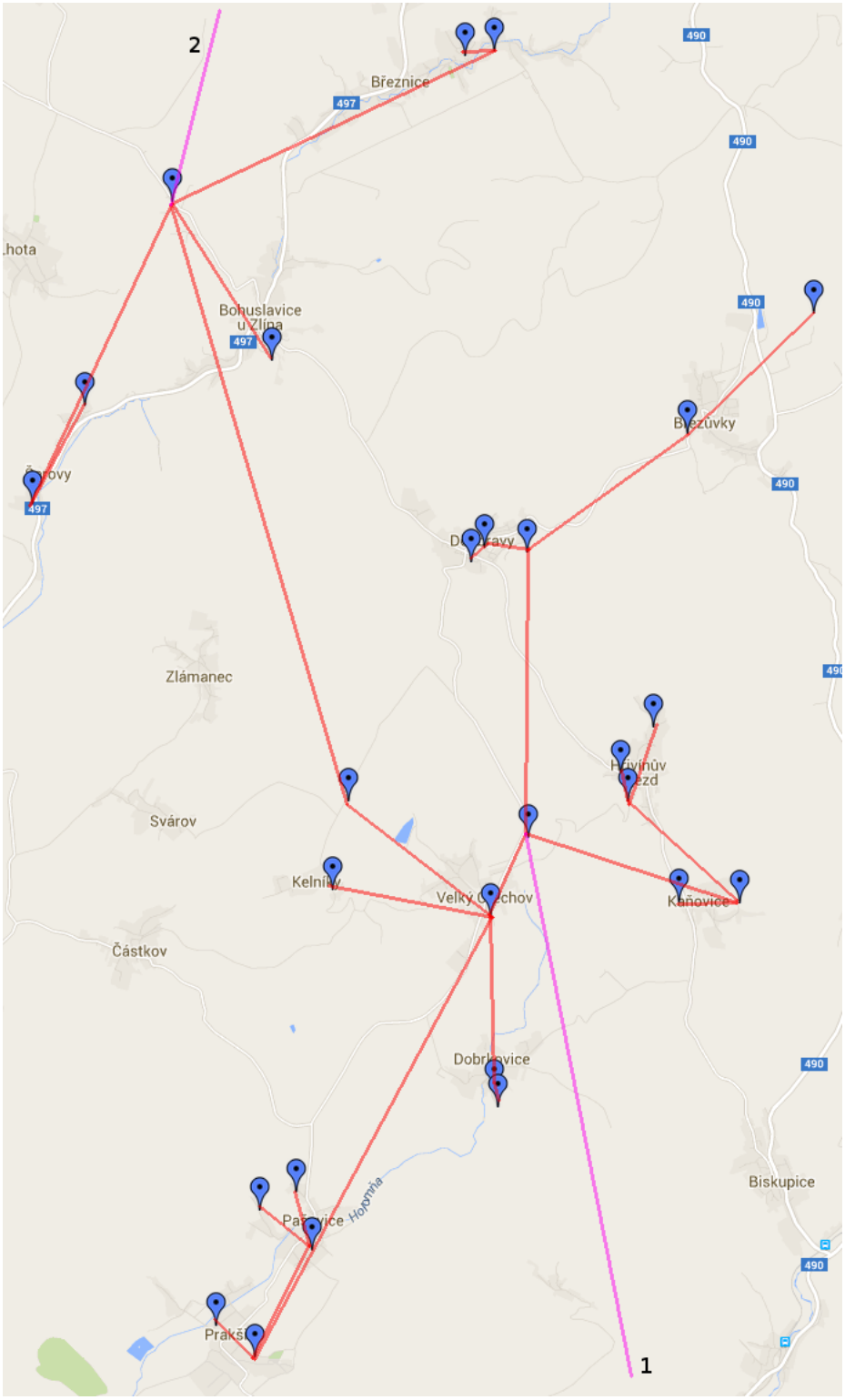
SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ISP	Internet service provider
TCP/IP	Transmission Control Protocol/Internet Protocol
IP	Internet Protocol
DSP	číslicové zpracování signálů – Digital Signal Processing
QoE	Quality of Experience
QoS	Quality of Service
FUP	Fair User Policy
LAN	Local Area Network
MAN	Metropolitan Area Network
WAN	Wide Area Network
ICMP	Internet Control Message Protocol
WAN	Wide Area Network
WAN	Wide Area Network
WAN	Wide Area Network
ECN	Explicit Congestion Notification
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
RTP	Real-time Transport Protocol
OS	Operační systém
DSCP	Differentiated services code point
RSMA	Reverse polarity SMA connector
MIMO	Multiple input, multiple output
POE	Power over Ethernet
TDMA	Time Division Multiple Access

CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
TOS	Type of service
PTP	Point to point
HTB	Hierarchical token bucket
RED	Random Early Detection
SFQ	Stochastic Fairness Queueing
FIFO	First In First Out
RFC	Request For Comments
IEEE	Institute of Electrical and Electronics Engineers
ISO/OSI	International Organization for Standardization/Open Systems Interconnection
SLA	service-level agreement
POP	Point of Presence
HTTP	Hypertext Transfer Protocol
PHP	Hypertext Preprocessor
NAT	Network address translation
VoIP	Voice over Internet Protocol

SEZNAM PŘÍLOH

A TOPOLOGIE SÍTĚ



Obr. A.1: Topologie sítě Durnet.cz

B ZACHYCENÉ PAKETY

```
11:24:08.978381 IP (tos 0x0, ttl 47, id 30103, offset 0,\
  flags [DF], proto TCP (6), length 1420)
  77.174.142.232.51413 > 172.20.7.37.65167: Flags [.],\
  seq 4052087614:4052088982, ack 3613522785, win 331,\
  options [nop,nop,TS val 904718887 ecr 1187928787],\
  length 1368

11:24:08.978490 IP (tos 0x0, ttl 103, id 24160, offset 0,\
  flags [none], proto UDP (17), length 1466)
  76.169.245.109.39789 > 172.20.9.110.52805: UDP,\
  length 1438

11:24:08.978621 IP (tos 0x0, ttl 47, id 30104, offset 0,\
  flags [DF], proto TCP (6), length 1420)
  77.174.142.232.51413 > 172.20.7.37.65167: Flags [.],\
  seq 4052088982:4052090350, ack 3613522785, win 331,\
  options [nop,nop,TS val 904718887 ecr 1187928787],\
  length 1368

11:24:08.978656 IP (tos 0x0, ttl 54, id 48755, offset 0,\
  flags [DF], proto TCP (6), length 6940)
  77.75.75.9.80 > 172.20.9.168.3287: Flags [.],\
  seq 1433985589:1433992489, ack 3119118834, win 66,\
  length 6900

11:24:08.978761 IP (tos 0x3,CE, ttl 51, id 0, offset 0,\
  flags [DF], proto UDP (17), length 313)
  109.90.134.12.6881 > 172.20.3.30.6881: UDP, length 285

11:24:08.978761 IP (tos 0x0, ttl 61, id 40603, offset 0,\
  flags [none], proto UDP (17), length 48)
  172.20.7.37.45685 > 81.82.159.237.60866: UDP, length 20
```

C PROGRAM NA ZOBRAZENÍ ČETNOSTI

```
#include <fstream>
#include <iostream>
using namespace std;

int main()
{
	std::ifstream file("vsechno_separovane.txt");	// otevření souboru
	std::string str;
	long int pocet_celkovy = 0;			// vytvoření proměnných
	long int pocet_00 = 0;
	long int pocet_01 = 0;
	long int pocet_10 = 0;
	long int pocet_11 = 0;

	while (std::getline(file, str))			// čtení souboru po řádcích
	{
		switch((int)(str[0])){			// převod z hexadecimální na
			case 48: pocet_00++; break;	// ASCII, uložený v jako integer
			case 49: pocet_01++; break;
			case 50: pocet_10++; break;
			case 51: pocet_11++; break;	// zařazení paketu do kategorie
			case 52: pocet_00++; break;	// podle ECN pole, přičtení
			case 53: pocet_01++; break;
			case 54: pocet_10++; break;
			case 55: pocet_11++; break;
			case 56: pocet_00++; break;
			case 57: pocet_01++; break;
			case 97: pocet_10++; break;
			case 98: pocet_11++; break;
			case 99: pocet_00++; break;
			case 100: pocet_01++; break;
			case 101: pocet_10++; break;
			case 102: pocet_11++; break;
		}
	}
	pocet_celkovy = pocet_00 + pocet_01 + pocet_10 + pocet_11;
	cout << "Pocet celkovy: " << pocet_celkovy << "\n";
	cout << "Pocet 00: " << pocet_00 << "\n";
	cout << "Pocet 01: " << pocet_01 << "\n";	// vypsání výsledků
	cout << "Pocet 10: " << pocet_10 << "\n";
	cout << "Pocet 11: " << pocet_11 << "\n";

	file.close();
}
```

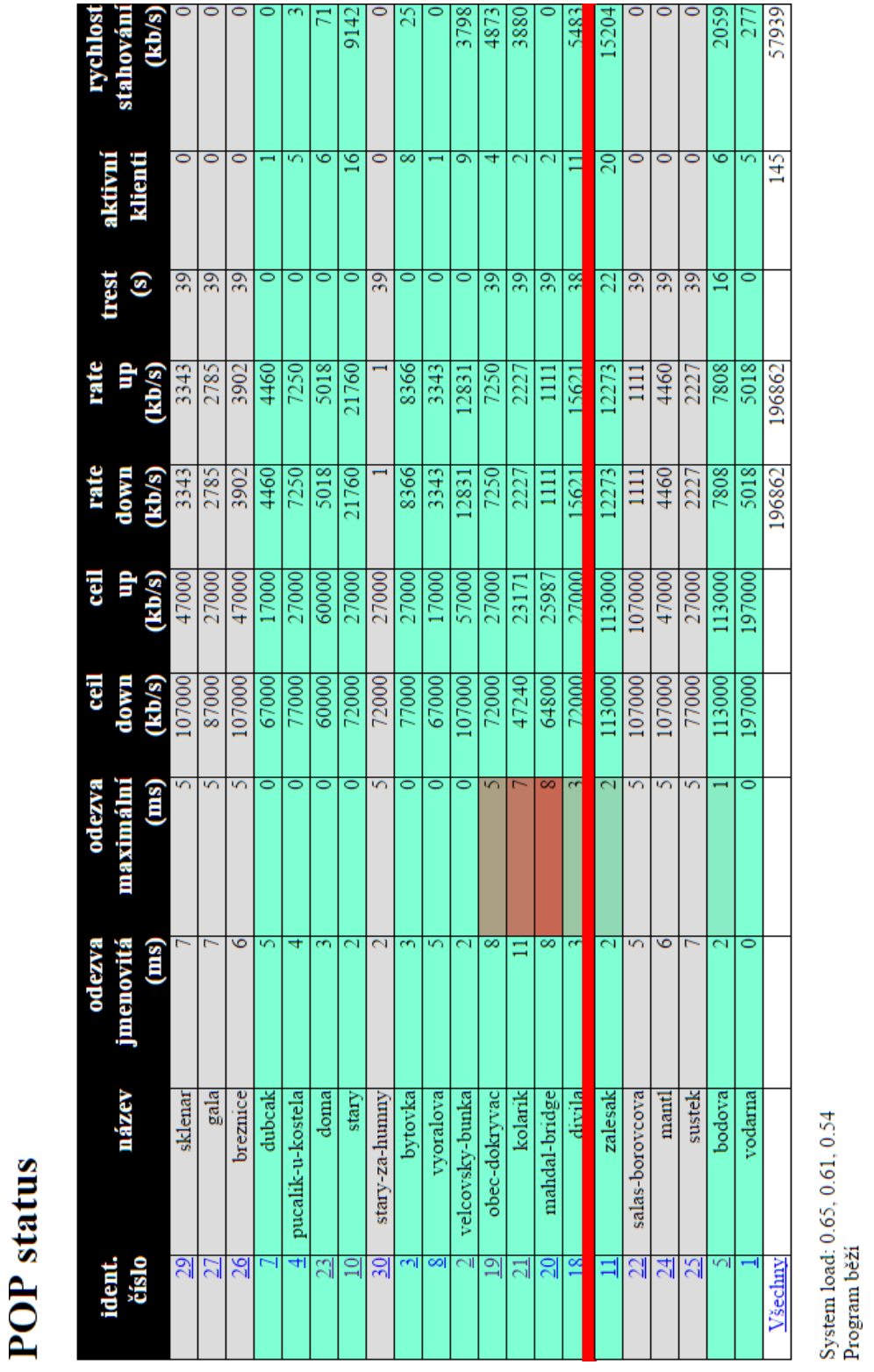
Obr. C.1: Program na sečtení paketů podle CE a ECT bitů

D OBSAH ZÁKLADNÍHO KONFIGURAČNÍHO SOUBORU

```
# hlavni konfiguracni soubor
# Durnet.cz GW

# prikazy uklada do souboru
# pokud vse funguje, vypnout
COMMANDTOTXT=no
#COMMANDTOTXT=yes



# zakaze prime zadavani prikazu
# pouze pro lazeni - simulace, nic se v iptables nezmeni
SIMULATIONONLY=no
#SIMULATIONONLY=yes

# Garantovane parametry paterni linky (Mb/s)
ROOTLINEUP=100
ROOTLINEDOWN=100

# definice interface
# interface smerem do venkovni site/peering
IFACEUP=gateway
# interface do mistni site
IFACEDOWN=mistni



# neaktivni klienti jsou vyhozeni z HTB (uspora vykonu)
#ECONOMYHTB=no
ECONOMYHTB=yes

# cas necinnosti klienta, po kterem se uvede do\
  neaktivniho stavu (s * 5)
ECONOMYTIMEOUT=120

# seznam sdilenych IP addres mezi klienty (= ip\
  adresa routeru)
```

```
# na tyto IP adresy je mozne netmapovat klienty z mistni\
  site
PUBLICIPS=84.244.73.34,84.244.73.35,84.244.73.36,\
  84.244.73.37,84.244.73.38
PUBLICSHAREDIPS=84.244.73.34,84.244.73.35,84.244.73.36

# hranice maximalni odezvy, po ktere se zacne AP\
  zpomalovat (ms)
GWMAX=13

# hranice maximalni odezvy, po ktere se zacne ap zvysovat\
  (ms)
GWMIN=3

# procentualni skok, o ktery se spomali AP, pii prekroceni\
  maximalni hodnoty (\%)
STEPDOWN=20

# skok, o ktery se zrychli AP, pri srovnani odezvy (Mb/s)
STEPUP=1000

# cas v sekundach, jak dlouho se ceka, nez se zacne AP\
  zrychlovat (s)
UPTIME=40
```

E TABULKA STAV POP



POP status

ident. číslo	název	odezva jmenovitá (ms)	odezva maximální (ms)	ceil down (kb/s)	ceil up (kb/s)	rate down (kb/s)	rate up (kb/s)	trest (s)	aktivní klienti	rychlost stahování (kb/s)
29	sklenar	7	5	107000	47000	3343	3343	39	0	0
27	gala	7	5	87000	27000	2785	2785	39	0	0
26	breznice	6	5	107000	47000	3902	3902	39	0	0
7	dubcak	5	0	67000	17000	4460	4460	0	1	0
4	pucalik-u-kostela	4	0	77000	27000	7250	7250	0	5	3
23	doma	3	0	60000	60000	5018	5018	0	6	71
10	stary	2	0	72000	27000	21760	21760	0	16	9142
30	stary-za-hummy	2	5	72000	27000	1	1	39	0	0
3	bytovka	3	0	77000	27000	8366	8366	0	8	25
8	vyoralova	5	0	67000	17000	3343	3343	0	1	0
2	velcovsky-bunka	2	0	107000	57000	12831	12831	0	9	3798
19	obec-dokryvac	8	5	72000	27000	7250	7250	39	4	4873
21	kolarik	11	7	47240	23171	2227	2227	39	2	3880
20	mahdal-bridge	8	8	64800	25987	1111	1111	39	2	0
18	divila	3	3	72000	27000	15621	15621	38	11	5483
11	zalesak	2	2	113000	113000	12273	12273	22	20	15204
22	salas-borovcova	5	5	107000	107000	1111	1111	39	0	0
24	mantl	6	5	107000	47000	4460	4460	39	0	0
25	susek	7	5	77000	27000	2227	2227	39	0	0
5	bodova	2	1	113000	113000	7808	7808	16	6	2059
1	vodarna	0	0	197000	197000	5018	5018	0	5	277
Všechny						196862	196862		145	57939

System load: 0.65, 0.61, 0.54
Program běží

Obr. E.1: Výstup programu, zobrazení stavu POP

F TABULKA STAV KLIENTI

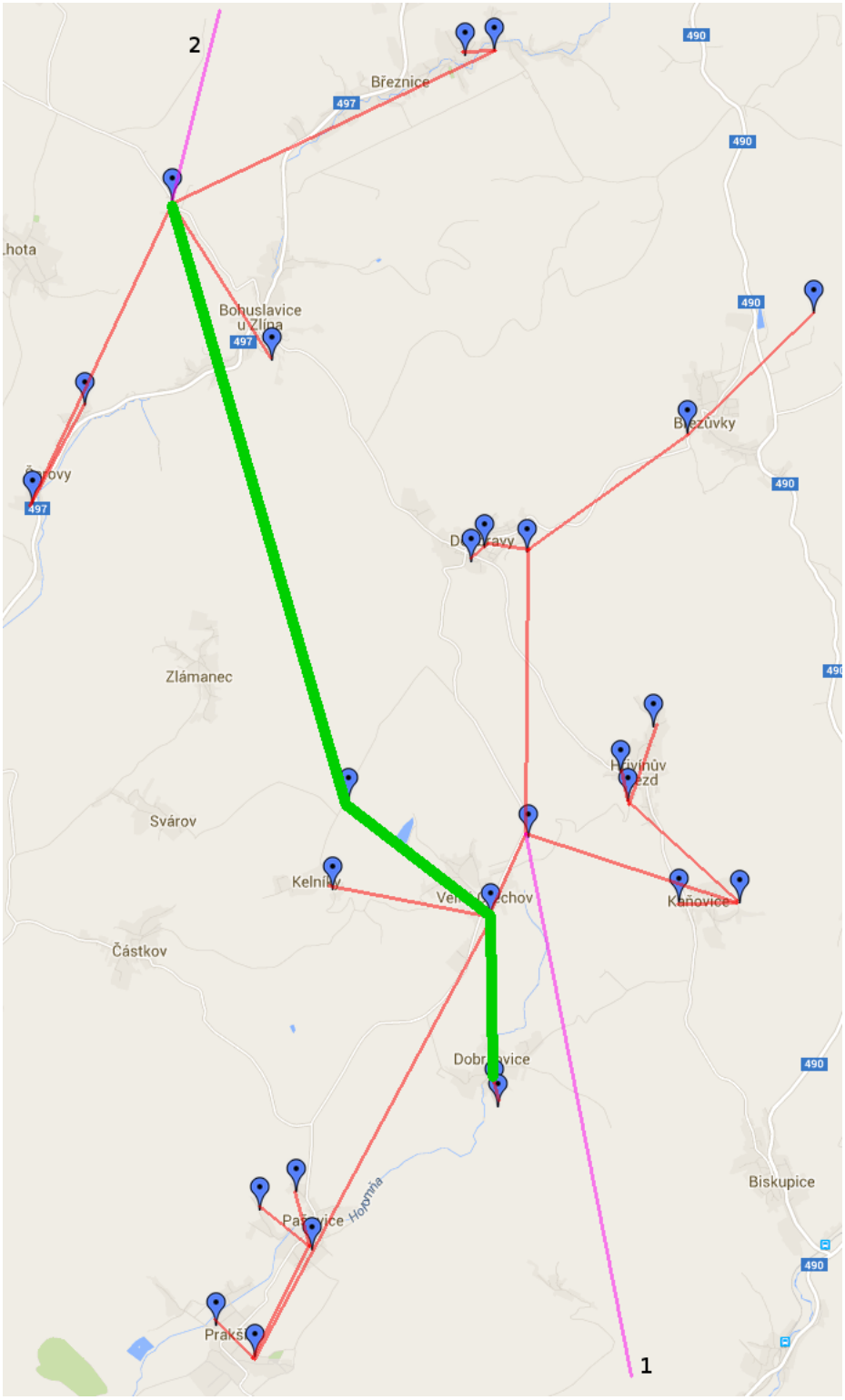
Klienti na POP 19

ident. číslo	jméno	ceil down (kb/s)	ceil up (kb/s)	rate down (kb/s)	rate up (kb/s)	rychlost stahování (kb/s)
7009	lenka-kudlacova	20000	5000	556	556	0
7013	jana-sestakova	20000	5000	556	556	0
7014	bohumil-vitek	20000	5000	556	556	1
7015	petr-gregor	20000	5000	556	556	4837
7024	vladimir-sustek	20000	5000	556	556	0
7026	miroslav-chmela	20000	5000	556	556	0
7027	vlastimil-kadlcak	10000	5000	556	556	0
7035	josef-lekes	10000	5000	556	556	0
7054	jaroslav-mostek	5000	1000	556	556	0
7056	martina-caputova	5000	1000	556	556	0
7064	daniel-valerian	5000	1000	556	556	0
7065	vaclav-gregor	20000	5000	556	556	15
7077	vera-olejnikova	10000	5000	556	556	0
				7228	7228	4853

System load: 0.87, 0.52, 0.5
Program běží

Obr. F.1: Výstup programu, zobrazení stavu uživatelů sítě

G MĚŘENÍ SÍTĚ



Obr. G.1: Topologie pro měření sítě

H PROGRAM PRO MĚŘENÍ SÍTĚ

```
#!/bin/bash

#definice vystupnich souboru
SODEZVA1="vysledky/odezva1.txt"
SODEZVA2="vysledky/odezva2.txt"
SODEZVA3="vysledky/odezva3.txt"
SHTB="vysledky/htb.txt"
SRYCHLOSTD="vysledky/rychlostD.txt"
SRYCHLOSTU="vysledky/rychlostU.txt"

#vymazani vystupnich souboru
echo "odezva1" > $SODEZVA1
echo "odezva2" > $SODEZVA2
echo "odezva3" > $SODEZVA3
echo "stavhtb" > $SHTB
echo "rychlostD" > $SRYCHLOSTD
echo "rychlostU" > $SRYCHLOSTU

var=0;

#provedeni 60 mereni po 500ms
while [ "60" -gt "$var" ]
do
  echo "mereni cislo: $var"
  let "var=var+1"
  php odezva.php 172.20.2.27 >> $SODEZVA1 &
  php odezva.php 172.20.2.249 >> $SODEZVA2 &
  php odezva.php 172.20.2.248 >> $SODEZVA3 &
  php mereniHTB.php >> $SHTB &
  php rychlost.php &
  sleep 0.5
done
```

I PROGRAM PRO MĚŘENÍ ODEZVY

```
<?php
function ping($ip) {
    $prikaz = "fping␣" . $ip . "␣-c1␣-t1000␣2>&1";
    $vysledek = shell_exec($prikaz);
    $odezvaPole = explode("/", $vysledek);
    $odezva = $odezvaPole[sizeof($odezvaPole) - 1];
    $odezva = str_replace("␣", "", $odezva);
    $odezva = str_replace("\n", "", $odezva);
    if ($odezva == "100%") {
        return "timeout";      // nedostupne
    }
    return $odezva;
}
echo ping($argv[1]);
```

J PROGRAM PRO MĚŘENÍ RYCHLOSTI

```
<?php
function refreshData() {
    $d1 = "iptables␣-v␣-t␣mangle␣-S␣PREROUTING␣|␣grep␣K4\
␣|␣head␣-1␣>␣rychlost1.log.txt";
    exec($d1);
    $d2 = "iptables␣-v␣-t␣mangle␣-S␣POSTROUTING␣|␣grep␣K4\
␣|␣head␣-1␣>␣rychlost2.log.txt";
    exec($d2);
}
function saveData($hodnota, $cesta) {
    $myfile = fopen($cesta, "a") or die("Error!");
    fwrite($myfile, $hodnota."\n");
    fclose($myfile);
}
function readData() {
    $myfile = fopen("rychlost1.log.txt", "r") or die("Error!");
    $r1Surove = fread($myfile,filesize("rychlost1.log.txt"));
    fclose($myfile);
    $myfile2 = fopen("rychlost2.log.txt", "r") or die("Error!");
    $r2Surove = fread($myfile2,filesize("rychlost2.log.txt"));
    fclose($myfile2);
    $r1 = explode("␣",$r1Surove)[8];  // up
    $r2 = explode("␣",$r2Surove)[8];  //down
    saveData($r1,"vysledky/rychlostU.txt");
    saveData($r2,"rychlostD.txt");
}
function start() {
    refreshData();
    readData();
}
start();
```

K PROGRAM PRO MĚŘENÍ STAVU NOVÉHO ALGORITMU

```
<?php
function vycti() {
    $c = "cat␣/run/gw/aps.out␣|␣grep␣POPid";
    $vysledek = shell_exec($c);
    $pole = explode("␣",$vysledek);
    $pole2 = array();
    foreach ($pole as $item) {
        if (strlen($item) > 1) {
            $pole2[] = $item;
        }
    }
  $hodnota = (int) $pole2[4];
    echo $hodnota;
    echo PHP_EOL;
}
vycti();
```

L OBSAH ELEKTRONICKÉ PŘÍLOHY

K práci je přiložen nepřepisovatelný disk DVD s následujícími soubory.

```
ProgramHlavni.zip
ProgramZobrazeni.zip
ProjektCpp.zip
TabulkaMereni.zip
RizeniDatovehoTokuvISPSiti.pdf
```

Soubor `ProgramHlavni.zip` je komprimovaný archiv ve formátu `zip`, který obsahuje konfigurační soubory a program. Z těchto souborů je možné nakonfigurovat a spustit program vytvořený v této práci a tak vytvořit instanci nového algoritmu, více informací je uvedeno v kap. 6.2.4.

Soubor `ProgramZobrazeni.zip` obsahuje program určený pro webový server, který umožňuje přehledné zobrazení stavu programu v reálném čase.

Soubor `ProjektCpp.zip` obsahuje kompletní projekt tj. zdrojový kód programu i zkompilovaný program. Tyto soubory jsou potřebné pro případnou modifikaci programu.

Soubor `TabulkaMereni.zip` obsahuje soubor se zdrojovými daty, výpočty a grafy z měření v kap. 7.

Soubor `RizeniDatovehoTokuvISPSiti.pdf` obsahuje elektronickou verzi této práce.