VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2020 Tomáš Opravil

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

TESTOVÁNÍ, SMĚROVÁNÍ A QOS V PRIVÁTNÍCH SÍTÍCH
TESTING, ROUTING AND QOS IN PRIVATE NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Tomáš Opravil

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. Vladislav Škorpil, CSc.

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Telekomunikační a informační systémy**

Ústav telekomunikací

Student: Tomáš Opravil ***ID:*** 195404
Ročník: 3 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Testování, směrování a QoS v privátních sítích

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s problematikou směrování a kvality služby QoS v privátních sítích a s možnostmi jejich testování. Vytvořte model jednoduché privátní sítě, navrhněte a realizujte její testování. Pomocí různého software, např. Riverbed, Wireshark, apod. testování privátní sítě simulujte. Výsledky simulací i praktických měření podrobte diskusi, porovnejte je a zhodnoťte. Vypracujte minimálně dvě laboratorní úlohy pro studenty včetně vzorových protokolů a manuálů pro učitele.

DOPORUČENÁ LITERATURA:

[1] PUŽMANOVÁ, Rita. Moderní komunikační sítě A-Z. Computer Press, Brno 2007

[2] ŠKORPIL, Vladislav. Vysokorychlostní komunikační systémy. FEKT, Brno 2014

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 8.6.2020

Vedoucí práce: doc. Ing. Vladislav Škorpil, CSc.

prof. Ing. Jiří Mišurec, CSc.
předseda rady studijního programu

Abstrakt

Diplomová práce se zabývá interakcí technologií směrování se zajištěním kvality služeb. Zaměřuje se zejména na fungování směrovacích protokolů RIP a OSPF v kombinaci s frontami FIFO, PQ a WFQ technologie DiffServ. Hlavním výstupem jsou dvě kompletní laboratorní úlohy, z nichž první se zabývá teoretickým zkoumáním dvou zmíněných technologií programem Riverbed. Navazující úloha využívá emulátoru EVE-NG a má za cíl ověřit a prohloubit nabyté teoretické poznatky v praxi.

Klíčová slova

RIP, OSPF, FIFO, PQ, WFQ, Riverbed, EVE-NG

Abstract

This thesis concerns the interaction of technologies of routing with ensuring the quality of services. It focuses on the function of routing protocols RIP and OSPF in combination with FIFO, PQ and WFQ queues of the technology DiffServ. The main output consists of two complex laboratory tasks. The first deals with theoretical research of the two mentioned above Riverbed programme technologies. The second subsequent task uses the emulator EVE-NG to aim the verification as well as deepening the acquired theoretical knowledge in practise.

Keywords

RIP, OSPF, FIFO, PQ, WFQ, Riverbed, EVE-NG

OPRAVIL, Tomáš. *Testování, směrování a QoS v privátních sítích*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2020. 100 s. Vedoucí diplomové práce byl doc. Ing. Vladislav Škorpil, CSc.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Testování, směrování a QoS v privátních sítích jsem vypracoval samostatně pod vedením vedoucího diplomové bakalářské 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 uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této diplomové bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: …… ……

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce doc. Ing. Vladislavovi Škorpilovi, CSc., za odborné vedení, konzultace, trpělivost a odbornou pomoc při zpracování mé diplomové práce. Dále děkuji panu Ing. Václavovi Oujezskému, Ph.D. za cenné rady a aktivní pomoc při řešení problémů. Nakonec děkuji své sestře Bc. Kláře Opravilové a zbytku své rodiny za obrovskou podporu během tvorby této práce.

V Brně dne: …… ……

podpis autora

Obsah

Úvod 12
1. Routing 13
1.1 Nedynamické metody směrování 13
1.1.1 Statické směrování 13
1.1.2 Náhodné směrování 14
1.1.3 Lavinové směrování 14
1.2 Dynamické metody směrování 14
1.2.1 Centralizované směrování 14
1.2.2 Izolované směrování 15
1.2.3 Distribuované směrování 15
1.3 Směrovací protokoly 15
1.3.1 RIP 16
1.3.2 EIGRP 17
1.3.3 IS-IS 18
1.3.4 OSPF 18
1.3.5 BGP 20
2. Zajištění kvality SLUŽEB – QOS 20
2.1 IntServ – Integreated Services 21
2.2 DiffServ – Differentiated Services 22
2.2.1 DSCP – DiffServ Code Point 22
2.2.2 PHB – Per Hop Behavior 22
2.3 Fronty v technologii DiffServ 23
2.3.1 FIFO 24
2.3.2 PQ 24
2.3.3 WRR 24
2.3.4 WFQ 25
3. Použité programy 25
3.1 Riverbed 25
3.2 EVE-NG 26
3.3 Wireshark 27
3.4 Iperf 27

4. Cíle laboratorních úloh 28
4.1 Návrhy sítí a metody proměření úloh 29
5. První laboratorní úloha v programu Riverbed 30
5.1.1 Sestavení sítě 30
5.1.2 Nastavení směrovacích protokolů RIP a OSPF 31
5.1.3 Nastavení provozovaných služeb 32
5.1.4 Nastavení výpadku linky 33
5.1.5 Nastavení front QoS 34
5.1.6 Spuštění simulace 35
6. Druhá laboratorní úloha v emulačním prostředí EVE-NG 35
6.1.1 Sestavení sítě 36
6.1.2 Instalace programu Iperf 37
6.1.3 Nastavení IP adres a směrovacího protokolu RIP 38
6.1.4 Nastavení front QoS 40
6.1.5 Nastavení směrovacího protokolu OSPF 43
6.1.6 Příprava programů Wireshark a Iperf 44
6.1.7 Spuštění a průběh úlohy 46
7. Výsledky první laboratorní úlohy 47
8. Výsledky druhé laboratorní úlohy 51
9. Porovnání laboratorních úloh 64
10. Závěr 65
Literatura 66
Seznam symbolů a zkratek 67
Seznam příloh 68

Seznam obrázků

Obr. 5.1: Topologie laboratorní úlohy 31
Obr. 5.2: Profil VoIP 33
Obr. 5.3: Nastavení WFQ 34
Obr. 6.1: Topologie laboratorní úlohy 37
Obr. 6.2: Nastavení rozhraní směrovače R1 39
Obr. 6.3: Nastavení protokolu RIP na R1 39
Obr. 6.4: Výpis rozhraní směrovače R1 40
Obr. 6.5: Nastavení priorit pro scénáře PQ 41
Obr. 6.6: Výpis rozhraní směrovače R1 pro PQ scénáře 42
Obr. 6.7: Zrušení priorit ve scénářích PQ 42
Obr. 6.8: Výpis rozhraní směrovače R1 pro WFQ scénáře 43
Obr. 6.9: Nastavení OSPF protokolu na R1 směrovači 43
Obr. 6.10: Seznam grafů v programu Wireshark 44
Obr. 6.11: Nastavení programu Iperf na straně serveru 45
Obr. 6.12: Nastavení programu Iperf na straně klienta 46
Obr. 7.1: Zahazování IP paketů s použitím RIP protokolu 49
Obr. 7.2: Zahazování IP paketů s použitím OSPF protokolu 49
Obr. 7.3: Příjem směrovacích dat protokolu RIP na směrovači 1 50
Obr. 7.4: Příjem směrovacích dat protokolu OSPF na směrovači 1 50
Obr. 8.1: Celkový příjem dat RIP FIFO 51
Obr. 8.2: Celkový příjem dat RIP PQ 51
Obr. 8.3: Celkový příjem dat RIP WFQ 52
Obr. 8.4: Celkový příjem dat OSPF FIFO 53
Obr. 8.5: Celkový příjem dat OSPF PQ 53
Obr. 8.6: Celkový příjem dat OSPF WFQ 54
Obr. 8.7: Příjem UDP/TCP dat RIP FIFO 55
Obr. 8.8: Příjem UDP/TCP dat RIP PQ 55
Obr. 8.9: Příjem UDP/TCP dat RIP WFQ 55
Obr. 8.10: Příjem UDP/TCP dat OSPF FIFO 57
Obr. 8.11: Příjem UDP/TCP dat OSPF PQ 57

Obr. 8.12: Příjem UDP/TCP dat OSPF WFQ 58
Obr. 8.13: Příjem směrovacích dat RIP FIFO 59
Obr. 8.14: Příjem směrovacích dat RIP PQ 60
Obr. 8.15: Příjem směrovacích dat RIP WFQ 60
Obr. 8.16: Příjem směrovacích dat OSPF FIFO 62
Obr. 8.17: Příjem směrovacích dat OSPF PQ 62
Obr. 8.18: Příjem směrovacích dat OSPF WFQ 62

Seznam tabulek

Tab. 2.1: Doporučené DSCP hodnoty pro AF PHB 23
Tab. 5.1: DSCP hodnoty aplikací 32
Tab. 8.1: Výsledky programu Iperf..... 63

ÚVOD

V dnešní době roste čím dál více poptávka po komunikačních sítích, a to jak u malých začínajících firem, tak i u již existujících velkých firem, které se rozšiřují do dalších zemí a požadují dostatečně rychlou a spolehlivou komunikaci.

Je proto důležité zajistit dostatečné odborné vzdělání studentů SŠ a VŠ pro pochopení konfigurace a fungování komunikačních sítí a zajištění různých požadavků na síť. Většina studentů FEKT VUT BR má možnost prozkoumat v laboratořích buď problematiku směrování, nebo zajištění kvality služeb, ne však jejich kombinaci a vzájemnou interakci.

Z tohoto důvodu se tato bakalářská práce věnuje technologiím směrování v kombinaci se zajištěním kvality služeb a možnými problémy neoptimální konfigurace sítě. Výstupem této práce jsou dvě laboratorní úlohy, které se zabývají již zmíněnou problematikou. První laboratorní úloha se zabývá tímto tématem na teoretické bázi, zatímco druhá úloha se zaměřuje na danou problematiku v emulátoru EVE-NG, který věrněji napodobuje fyzická zařízení. Obě úlohy mají za cíl pomoci studentům pochopit spolupráci směrování se zajištěním kvality služeb.

1. ROUTING

Směrování neboli routing je technika, která propojuje jednotlivé sítě po síťové vrstvě. Tato technika v podstatě nachází nejlepší cesty skrze mezilehlé uzly k propojení uživatelů. K tomu jsou využívány různé protokoly, které mají také za úkol najít nejvhodnější cesty a v případě přerušení již používané cesty, například výpadkem mezilehlého uzlu nebo přerušení linky, najít vhodné náhradní cesty. Dříve se ke směrování používaly výhradně směrovače, v dnešní době se však už zavádějí i L3 přepínače a firewally, které jsou schopny pracovat i na síťové vrstvě a podporují směrování. Proces směrování můžeme rozdělit na proces doručování paketů, kde je možné přímé doručování paketů, pokud je zdrojová i cílová stanice ve stejné síti, a na proces předávání paketů, který pakety přeposílá po mezilehlých uzlech v řetězci cesty od zdroje k cíli skrze různé sítě.

Pro směrování potřebuje síťová vrstva znát informace o topologii sítě a adresách uzlů. Většina sítí používá topologii neúplného polynomu, v těchto sítích pak pro některé uzly existuje více cest a pro některé neexistuje žádná přímá cesta. Informace o topologii jsou důležité k zajištění redundance komunikačních linek a tvorby cest pro přenos paketů. Způsoby směrování můžeme rozdělit na nedynamické a dynamické, tyto způsoby jsou popsány níže. [1]

1.1 Nedynamické metody směrování

Tyto metody patří obecně k nejjednodušším směrovacím metodám. Jinými slovy jde o nejsnadněji implementované metody, avšak jsou zatíženy mnohými problémy, jako je například zbytečné vytížení sítě či neschopnost aktivně reagovat na změny v síti. [1]

1.1.1 Statické směrování

Také tato metoda patří mezi nejjednodušší směrovací metody. Je založena na statických cestách vepsaných do směrovacích tabulek. Každý uzel má pevně definovanou výstupní linku pro konkrétního adresáta. Tento systém nemůže dynamicky reagovat na změny v topologii, jako například na výpadky zařízení nebo

přetížení sítě. Změny směrování je nutné dělat ručně na každém uzlu, což je pracné. [1]

1.1.2 Náhodné směrování

Tato metoda značně zatěžuje síť a v praxi se nepoužívá, je založena na náhodném pohybu paketů po síti s očekáváním, že pakety za určitou dobu k cíli doputují. Jedinou výhodou této metody je její jednoduchost. [1]

1.1.3 Lavinové směrování

Metoda je založena na tom, že když dorazí paket do uzlu, tak je nakopírován a vyslán na všechny linky kromě linky, ze které dorazil. Nejprve se však testují všechny sousední uzly, zda již daný paket neobdržely z jiného uzlu. Tento princip směrování je velmi jednoduchý, spolehlivý a zaručuje nejkratší cestu k cíli, ale značně zatěžuje celou síť zbytečnými kopiemi paketů. Z těchto důvodů je tato metoda směrování vhodná jen pro nalezení nejkratší cesty mezi uzly a pro multicastovou komunikaci. [1]

1.2 Dynamické metody směrování

Dynamické směrování má za hlavní úkol adaptaci na změny v topologii, například velké zatížení uzlů, výpadky prvků nebo linek. Tuto adaptaci zajišťuje tím, že ve směrovacích tabulkách pro každý cíl existuje hned několik linek. První a nejrychlejší linky se využívají jako hlavní linky a při jejich výpadku se využije záložní linky. Důležitým bodem pro fungování dynamického směrování je i sdílení technických zpráv mezi uzly, ty si díky nim posílají aktuální informace o síti a aktualizují si tak směrovací tabulky, popřípadě zjistí výpadek linky. Nevýhodami dynamických metod jsou větší nároky a složitost na výpočetní hardware uzlů. [1]

1.2.1 Centralizované směrování

Centralizované směrování je metodou dynamického směrování, kde síť obsahuje centrální bod. Tento bod shromažďuje aktuální data o stavu sítě a na jejich základě zpracovává rozhodnutí o směrování, ta pak oznamuje ostatním uzlům v síti. Hlavním

problémem takto fungující sítě je riziko, že v případě výpadku centrálního bodu zkolabuje celá sít, dále se může snížit efektivita dynamiky směrování. To záleží zejména na frekvenci vyhledávání nejvhodnějších cest v sítí, pokud je frekvence malá, tak bude déle trvat, než centrální prvek získá aktuální informace o změnách v síti a zareaguje na ně. Posledním problémem může být i větší zátěž na síť, která se generuje přenosem informací do centrálního bodu a následným rozesíláním výsledků zpět do všech uzlů v síti. [1]

1.2.2 Izolované směrování

Principem tohoto typu směrování je, že si uzly vybírají nejvhodnější cesty samy, a to pouze na základě svých získaných informací. Uzly si získávají informace pozorováním směrů, ze kterých získávají pakety od jiných uzlů a na jejich základě rozhodují o směrování paketů. Problémem je, že uzly spolu nesdílejí informace, a to celkově omezuje jejich reakce na změny v síti. [1]

1.2.3 Distribuované směrování

V současnosti je nejvíce využívána metoda distribuovaného směrování. Principem této metody je decentralizované sdílení technických informací mezi uzly bez omezení. Jinými slovy to znamená, že si jednotlivé uzly pravidelně posílají aktuální informace o stavu sítě, tím pádem mohou efektivně reagovat na změny a vypočítávat nejlepší trasy k cíli. Tato metoda podporuje myšlenku decentralizovaného internetu. [1]

1.3 Směrovací protokoly

Tyto protokoly jsou soustavy pravidel, podle kterých se řídí mezilehlé uzly v síti, které reagují na změny v topologii. Ve směrování se využívají dynamické směrovací protokoly, které se v základu dělí na dva typy.

- **Distance-vector routing protokol** – Směrovače využívající tento typ protokolu zasílají směrovací tabulky periodicky svým sousedům, v níž jsou informace o vzdálenosti do dané sítě, sousední směrovače si aktualizují svou tabulku a tu poté odesílají dál svým sousedům. Pro výpočet cest se používá buď jedna metrika, například počet skoků v RIP protokolech, nebo vícero

metrik jako například zpoždění a propustnost linky. BGP protokol využívá upraveného distance-vector routing protokolu, ten se nazývá path-vector prokocol. [3]

- **Link-state routing protokol** – S použitím tohoto protokolu si směrovače udržují v paměti celou databázi síťové topologie. Pravidelně si posílají Hello pakety, kterými si zjišťují dostupnost ostatních směrovačů. Tyto pakety obsahují informace o směrovači, který pakety vyslal. Pokud dojde k nějaké změně v síti, tak si směrovače posílají link-state advertisements protokoly, díky nimž si aktualizují směrovací tabulky. K výpočtům nejlepších cest se využívá Dijkstrova algoritmu. [2]

Dynamické protokoly se dále dají dělit podle toho, zda se využívají v rámci nebo vně autonomních systémů, to je skupina, která je pod kontrolou jedné nebo více jednotek a skládá se ze směrovačů a IP sítí. Protokoly používané uvnitř autonomních systémů se nazývají IGP – Internal Gateway Protocols a patří do nich RIPv2, EIGRP, OSPF, IS-IS protokoly. Protokoly užívané ke směrování mezi autonomními systémy se označují jako EGP – External Gateway Protocol a v současnosti patří do této skupiny BGP protokol. [2]

1.3.1 RIP

Routing Information Protocol je dynamický směrovací protokol a spadá pod třídu Distance Vector protokolu, který pracuje s metrikou počítání skoků k cíli a pravidelným rozesíláním směrovacích tabulek sousedním směrovačům. Maximální počet skoků je 15 a nejlepší cesta je označena jako ta s nejmenším počtem skoků. Pokud je cílová síť vzdálená 16 a více skoků skrze směrovače, je označena za nedosažitelnou. Tento protokol určuje pouze jednu cestu k cíli, a tedy jasnou nevýhodou je nemožnost rozložení zátěže na více cest. RIP protokol existuje ve verzích RIPv1, RIPv2 a RIPng/RIPv6. Rozdíl mezi verzemi RIPv1 a RIPv2 protokolu je ten, že druhá verze podporuje třídní adresování, tedy proměnné délky masek podsítí, což má za následek efektivnější sumarizaci a méně adres ve směrovacích tabulkách v jednotlivých směrovačích. RIPv2 také podporuje autentizaci za pomoci MD5 šifrování. [3]

RIP protokol využívá pro své fungování 4 časovače:

- **Update Timer** – je časovač, po jehož vypršení směrovač posílá všem svým sousedním směrovačům celou svou směrovací tabulku. (defaultně 30 vteřin)
- **Invalid Timer** – časovač odpočítává dobu platnosti cesty (řádku) ve směrovací tabulce. Pokud před vypršením časovače obdrží směrovač aktualizaci cesty tak je časovač resetován na původní hodnotu, v případě vypršení časovače je cestě přiřazena metrika 16 a přesouvá se do stavu hold-down. (defaultně 180 vteřin)
- **Hold-down Timer** – odpočítává dobu, kdy jsou ještě informace o cestě přidrženy, ale již ji nejde obnovit ani v případě obdržení nových aktualizací o dosažitelnosti cíle. Tento časovač slouží k poskytnutí dostatku času k přizpůsobení se změnám v síti. (defaultně 180 vteřin)
- **Flush Timer** – slouží k vymazání záznamů o cestě a běží současně s Invalid Timer časovačem. Po 60 vteřinách od označení cesty za neplatnou dojde k jejímu vymazání. (defaultně 240 vteřin)

1.3.2 EIGRP

Enhanced Interior Gateway Routing Protocol je proprietární Cisco protokol jedná se o rozšíření IGRP protokolu z kategorie IGP a patří do skupiny distance-vector routing protokol, ale má i některé vlastnosti ze skupiny link-state routing protokol. Směrovače s tímto protokolem navazují a udržují vztahy paketem Hello, který je posílán periodicky v rozmezí 5-60 vteřin. Při navazování sousedství si směrovače po paketech Hello posílají pakety Update, kterými si navzájem aktualizují směrovací tabulky. Ukončení aktualizací směrovacích tabulek se potvrzuje ACK paketem. Při ztracení spojení mezi sousedy a po vypršení Hold Timeru, což je časovač určující, jak dlouho bude směrovač čekat, než vymaže sousední směrovač ze své tabulky, směrovač vyšle zprávu Goodbye, což je Hello paket, a tím oznámí změnu topologie ostatním směrovačům v síti. Protokol spolu s informacemi o cestě posílá i masku sítě a proto se jedná o classless protokol, ale lze jej přenastavit na chování classfull protokolu. Podporuje technologie IPv4, IPv6,

AppleTalk, dále pak podporuje autentizaci a sumarizaci cest a je podporován pouze firmou Cisco. [4]

1.3.3 IS-IS

Intermediate System to Intermediate System protokol používá stejně jako OSPF protokol k výpočtu cesty Dijsktrův algoritmus. Tento protokol se však od OSPF protokolu liší ve směrování oblastí (areas) a v jejich směru. Směrovače s IS-IS protokoly jsou rozdělovány do úrovní. Směrovače s úrovní 1 pracují pouze uvnitř oblastí a směrovače úrovně 2 pracují jen mezi oblastmi. Úrovně 1-2 pak spojují obě úrovně. [5]

1.3.4 OSPF

Open Shortest Path First je protokol typu Link State a směrovač s využitím tohoto protokolu získává informace o celé topologii sítě za pomoci skupiny paketů. Nejvhodnější cesta se určuje na základě topologie a metriky linek mezi jednotlivými uzly, která se skládá například ze zpoždění, zabezpečení a šířky pásma. Výsledná hodnota se pak označuje jako „cena spoje“ a nejvýhodnější cesta je ta s nejmenší hodnotou. Tento protokol využívá Dijkstrova algoritmu k odvození nejvhodnější cesty a podporuje dělení sítí na subsítě. Rozdělením na subsítě (oblasti) lze snížit počet zpráv, které si směrovače posílají a zmenšit tak celkové zatížení sítě. Směrovače znají topologii jen o dané oblasti, ve které se nacházejí a informace o okolních sítích nebo subsítích jim poskytují hraniční směrovače, které propojují oblasti mezi sebou. Hraniční směrovače shrnují informace o sítích, v nichž se nachází a posílají je do sousedních sítí. Výhodou tohoto protokolu oproti RIPv2 protokolu je hlavně skutečnost, že při změně v síti směrovač neposílá celou směrovací tabulku, což značně snižuje zatížení v síti a jeho reakce na změny v topologii je rychlejší. Směrovač si kontroluje dostupnost sousedních směrovačů, a pokud dojde ke změně v síti, tak ihned zasílá informaci jen o dané změně hierarchicky všem směrovačům v síti. Směrovače si přeposílají aktualizace vždy při změně anebo každých 30 minut v případě, že nenastane žádná změna. U tohoto protokolu není omezen počet skoků a cenu spoje lze nastavit manuálně, a tak upřednostnit pomalejší cestu v případě nízkých požadavků na přenos. Protokol

podporuje proměnné délky masky podsítí a druhá verze podporuje autentizaci za pomoci šifrování MD5. [3]

OSPF pro své fungování využívá pět druhů paketů:

- **Hello** – slouží k navázání a udržení spojení se sousedním směrovačem a využívá dvou časovačů. Hello interval udržuje spojení a je posílán každých 10 vteřin. Dead interval je zpravidla čtyřnásobek Hello intervalu, tedy 40 vteřin a pokud směrovač nepřijme žádný Hello paket od souseda do vypršení tohoto intervalu, spojení mezi směrovači zanikne.
- **Database Description** – slouží k přeposílání informací o topologii sítě.
- **Link State Request** – žádá o zaslání chybějící položky po obdržení paketu DBD.
- **Link State Update** – je odpověď na LSR paket s chybějící položkou.
- **Link State Acknowledgment** – slouží jako potvrzení o přijetí paketu LSU.

Při navazování nového spojení a výměně směrových informací mezi směrovači procházejí směrovače sedmi stavy:

- **Down** – je stav před obdržení Hello paketu, kdy ještě směrovač neví o existenci souseda.
- **Init** – je stav po obdržení Hello paketu, stáje ještě bez zařízení obousměrné komunikace.
- **Two-Way** – označuje stav, kdy je zřízena obousměrná komunikace a určuje se pověřený směrovač a záložní pověřený směrovač. Všechny směrovače pak při změně v topologii komunikují přes pověřený směrovač.
- **Exstart** – ustanovuje směrovače Master a Slave podle velikosti sekvenčního čísla v paketu DBD. Komunikaci začíná směrovač Master.
- **Exchange** – výměna DBD paketů mezi směrovači.
- **Loading** – výměna paketů LSR, LSU a LSA.
- **Full** – je stav plné synchronizace mezi směrovači.

1.3.5 BGP

Border Gateway Protocol je protokol využívaný k směrování mezi sousedními autonomními systémy. Patří mezi dynamické směrovací protokoly typu EPG a, jak již bylo zmíněno, patří mezi path-vector prokocol. To znamená, že informace směrovacích tabulek obsahují kromě ceny cesty i všechny předešlé skoky. Protokol je aplikační, pracuje nad TCP protokolem na portu 179, podporuje filtrování cest, beztřídní mezidoménové směrování a sdružování směrovacích cest. [6]

2. ZAJIŠTĚNÍ KVALITY SLUŽEB – QOS

Technologie QoS je skupina podpůrných funkcí, které mají zajistit upřednostnění přenosu dat pro náročnější služby před nenáročnými službami. Obecně se technologií QoS podporují služby v reálném čase, jako jsou například hlasové služby, live streaming či videokonference, nicméně je nutné zajistit i funkčnost nenáročných služeb (spolehlivý přenos dat, emailové služby, chat, atd). QoS tedy musí efektivně rozdělovat síťové prostředky pro služby real-time i obecné služby, k tomu je nutné provádění klasifikaci a značení datového provozu. Tato klasifikace probíhá na hranici domény, tedy na hraničním přepojovacím prvku, popřípadě s příslušnou dohodou lze přebírat klasifikaci datových toků od poskytovatele. Jelikož je klasifikace náročná na výpočetní výkon prostředků, je vhodné vybírat způsob klasifikace, která nezatěžuje hraniční uzel. Nejvhodnější je tedy, aby prvek pracoval s informacemi na vrstvě, na které pracuje obvykle, tedy přepínač, aby pracoval s informacemi na linkové vrstvě a směrovač na vrstvě síťové, případně na nižších vrstvách. Z toho vyplývá, že je efektivnější třídit datový tok podle síťové adresy, hardwarové adresy nebo příslušného fyzického portu na přepojovacím prvku než třízení dle obsahu dat v aplikační úrovni. Upřednostnění různých služeb lze použít i pro přepojovací prvky uvnitř sítě. Vložením informací do hlaviček datových jednotek o klasifikaci se umožní prioritní obsluhování služeb uvnitř sítě za použití jednoho z frontových mechanismů. V těchto mechanismech se třídí datové jednotky podle priority do front, ze kterých se jim přidělují dostupné a požadované síťové prostředky. Pro plnou podporu QoS je nutno implementovat tyto funkce po celé cestě datového toku, to znamená podporu QoS už od koncového

zařízení přes přepojovací prvky, směrovače, firewally i přepínače, až k cílovému koncovému zařízení. [9]

Mezi důležité parametry pro službu QoS patří:

- **Jitter** – proměnlivost zpoždění je negativní parametr zejména pro služby pracující v reálném čase. Zpoždění se dynamicky mění podle aktuálního zatížení a stavu sítě. Pro eliminaci tohoto parametru se na přijímací straně využívá Jitter Bufferu, který slouží jako zpožďovací paměť a mění tak zpoždění z dynamického na konstantní. [8]

- **Zpoždění** – je parametr, který sčítá celkově všechna zpoždění počínaje kódováním zdroje, paketizačním zpožděním a propagačním zpožděním až po Jitter Buffer zpoždění. Propagačním zpožděním se myslí potřebný čas k přenosu dat po síti od zdroje k cíli. Paketizační zpoždění je čas strávený umísťováním bitů do paketů, například do hlaviček. Kódováním zdroje se myslí A/D a D/A převod hlasu u zdroje a cíle. [8]

- **Ztrátovost paketů** – určuje množství paketů, které buď dorazily pozdě a u služeb pracujících v reálném čase se nedají už použít, nebo které vůbec nedorazily. Jde taktéž o negativní parametr, který se však u služeb v reálném čase dá do jisté míry tolerovat. Ztrátovost paketů může například zhoršit kvalitu hlasu či videa anebo u TCP služeb může zapříčinit nutnost opětovného odeslání dat. [8]

2.1 IntServ – Integreated Services

Služba IntServ pracuje hlavně v síťové vrstvě, a tedy je využívána směrovači a dalšími prvky pracujícími na této vrstvě. Tato služba rezervuje zdroje v síťových prvcích po celé cestě mezi zdrojem a příjemcem dat. Rezervace zdrojů v síti obsahuje nalezení vhodné cesty skrze uzly v síti a alokaci zdrojů pro každý směr přenosu od příjemce. U každého směrovače v cestě se zjišťuje požadavek oprávnění

a dostatek zdrojů pro uskutečnění relace. K tomu se používají různé protokoly jako například RSVP a COPS, které využívá firma CISCO, dále ještě YESSIR protokol.[7]

2.2 DiffServ – Differentiated Services

V této službě je teoreticky možné rozdělit tok do 64 tříd a v současnosti je často používána v podnikových sítích, ale i globálně díky své nenáročné implementaci. Síť s využitím této služby třídí datové toky jen na hraničních uzlech. V těchto uzlech síťové prvky třídí datové toky stejných druhů do jednotlivých tříd a to úpravou DSCP pole. Nerozlišují tak například jednotlivé hovory VoIP, ale upřednostní všechny VoIP hovory jako skupinu. Prvky uvnitř sítě se řídí podle PHB (Per Hop Behavior). Služba DiffServ je implementována například ve standardech RFC 247, RFC 2475 a RFC 2598. [8]

2.2.1 DSCP – DiffServ Code Point

Ke značkování IP paketů se používají pole v jejich záhlaví. U IPv4 protokolu jsou to Type of Service pole a u IPv6 zase pole Traffic Classes. V těchto 8bitových polích se využívá 6 bitů pro DSCP. Obecně paket s vyšší binární hodnotou DSCP má i vyšší prioritu. [8]

2.2.2 PHB – Per Hop Behavior

PHB slouží k identifikaci chování paketu v síti a dělí se na dvě skupiny. První skupinou je Expedited Forwarding PHB neboli urychlené doručování a druhou skupinou je Assured Forwarding PHB, tedy zaručené doručování. [8]

EF PHB zaručuje jistou šířku pásma a je vhodná pro služby v reálném čase, jako je například přenos videa či hlasu. V této skupině mají pakety nízké zpoždění, malý jitter a malou ztrátovost. Hodnota DSCP pro tuto skupinu je 101110. [8]

Skupina AF PHB je spíše určena pro spolehlivý přenos dat, a tedy pro TCP, jelikož jitter a zpoždění nejsou v této skupině důležité oproti ztrátovosti paketů.

V této skupině jsou čtyři třídy AF1 až AF4, které jsou rozděleny ještě na další tři podtřídy. Tyto podtřídy slouží pro priority zahození a rozlišují se na nízkou, střední a vysokou prioritu. Třídy AF mají přiděleny různé šířky pásma, velikosti bufferu a dalších síťových prostředků. V těchto třídách můžou být použity metody prevence proti zahlcení. Jedná se o metodu Random Early Detection a Weighted Random Early Detection. Při použití jedné z metod je vyšší pravděpodobnost zahození u paketů s vyšší prioritou zahození v případě hrozícího zahlcení. [8]

Tab. 2.1: Doporučené DSCP hodnoty pro AF PHB

Třída PHB	Podtřída PHB	Priorita zahození	DSCP
EF			101110
AF4	AF41	Malá	100010
	AF42	Střední	100100
	AF43	Velká	100110
AF3	AF31	Malá	011010
	AF32	Střední	011100
	AF33	Velká	011110
AF2	AF21	Malá	010010
	AF22	Střední	010100
	AF23	Velká	010110
AF1	AF11	Malá	001010
	AF12	Střední	001100
	AF13	Velká	001110
Best-effort			000000

2.3 Fronty v technologii DiffServ

Ke zpracování a rozdělení již klasifikovaných paketů slouží ve směrovačích mechanismy vyrovnávací paměti neboli fronty s určitou obslužnou logikou.

Nejčastější druhy front v DiffServ jsou například fronty FIFO, WFQ nebo PQ, tyto fronty jsou dále popsány níže.

2.3.1 FIFO

Jedná se o nejjednodušší mechanismus fronty bez jakéhokoliv algoritmu řízení. Pakety jsou obsluhovány tak, jak přijdou, podle čehož nese tento mechanismus i své jméno „First In First Out“. Výhoda tohoto mechanismu je jeho jednoduchost, je zapotřebí jen jedna vyrovnávací paměť, která odesílá data ve stejném pořadí, ve kterém přišly.

Hlavní nevýhodou je taktéž jednoduchost, kvůli absenci algoritmu řízení tento mechanismus nerozlišuje druhy dat, třídy provozu, a v případě zaplnění front je se všemi druhy dat zacházeno stejně. Tento mechanismus je vhodný pro službu Best-effort, pro provoz bez podpory QoS. [8]

2.3.2 PQ

Mechanismus front Priority Queuing řeší neduhy fronty FIFO, ale zároveň si zachovává jednoduchost, ovšem i tento mechanismus má své nevýhody. PQ mechanismus třídí klasifikované pakety do front podle priorit. Fronty s pakety s vyšší prioritou jsou obsluhovány do doby vyprázdnění fronty, pak mechanismus začne obsluhovat frontu s nižší prioritou. Zde může vzniknout problém „uvíznutí“ paketů ve frontách s nižší prioritou, pokud fronta s vyšší prioritou je stále doplňována dalšími pakety. Tento mechanismus je vhodný pro služby založené na UDP protokolu, jako například stream video či videohovor. Mechanismus PQ lze rozdělit například do dvou front, kde první fronta bude pro služby pracující v reálném čase a druhá fronta bude obsluhovat služby založené na TCP protokolu a budou tedy obsluhovány službou Best-effort. Při této implementaci je však nutné počítat s rizikem, že se pakety ve druhé frontě zpozdí a vyšší vrstvy je budou považovat za ztracené, což zapříčiní znovu odeslání těchto paketů od zdroje. Hrozí tak vyšší a zbytečné zatížení sítě. [8]

2.3.3 WRR

Weighted Round Robin je mechanismus front s váženou cyklickou obsluhou, která je schopná zajistit různým třídám různě velkou část dostupné šířky pásma. Tento mechanismus pracuje tak, že využívá dvouúrovňové cyklické plánování,

někdy je označován jako řízení front podle třídy (class-based queuing). První úroveň představuje výběr tříd od 1 po *n* a druhou úroveň představuje výběr fronty ze třídy. Jednotlivým frontám jsou přidělovány váhy a tím se zajišťuje proporcionální počet obsloužených paketů z každé fronty za cyklus. Procentuálním rozdělením času tráveného obsluhou třídy se pak zajišťuje poměrové rozdělení šířky pásma mezi jednotlivé třídy. [8]

2.3.4 WFQ

Frontový mechanismus Weighted Fair Queuing je mechanismus s váženou spravedlivou obsluhou. Je založen na skupině front se stejnou prioritou, které jsou průběžně obsluhovány a každá z nich má svou určitou váhovou hodnotu. Podle těchto hodnot je pak frontám přidělována část kapacity výstupní linky, v případě, že fronta zrovna nevyužívá svou část kapacity linky, může být tato část kapacity přidělena jiné frontě. Součet všech váhových hodnot pak odpovídá celé šířce pásma. Mechanismus za pomoci teoretického modelu, který lze označit jako váženou bitovou obsluhu, vypočítává pro každý příchozí paket čas, kdy nejpozději musí být obsloužen. Podle tohoto času je pak paket přidělen do určité fronty s určitou garancí kapacity linky. [8]

3. POUŽITÉ PROGRAMY

Tato kapitola se zabývá popisem použitých programů v laboratorních úlohách. Nejprve je pojednáváno o programu Riverbed, jenž byl použit pro tvorbu první laboratorní úlohy, dalším programem je emulační prostředí EVE-NG ve kterém byla vytvořena druhá úloha, dále následují pomocné programy Wireshark a Iperf, které byly použity ve druhé laboratorní úloze pro zatížení a analýzu sítě.

3.1 Riverbed

Program Riverbed Modeler Academic Edition 17.5 vytvořený firmou Riverbed je simulační prostředí sloužící pro návrh, simulaci a analýzu síťových mechanismů, protokolů a technologií. Je dostupný pro školy i studenty, kteří si zde mohou vyzkoušet navrhnout vlastní sítě s různými síťovými technologiemi a analyzovat je

v různých podmínkách, a tím také pochopit fungování těchto technologií a navrhnuté sítě. Tento program je vhodný jak pro školní účely, tak i pro analýzu již existujících sítí v různých podmínkách, které mohou a nemusí nastat a předvídat tak jejich chování v reálném životě.

Program disponuje rozsáhlou databází síťových prvků aktivních i pasivních, různých technologií a mnoha možnostmi jejich analýzy, dále pak program podporuje simulace s určitým zrychlením a je vhodný i pro testování simulovaných sítí v delším časovém horizontu.

3.2 EVE-NG

Emulační virtuální prostředí nové generace „Emulated Virtual Enviroment – Next Generation“ poskytuje ve webovém prohlížeči grafické laboratorní prostředí pro konfiguraci a simulaci různých síťových zařízeních jako jsou například Cisco směrovače a přepínače, zařízení mikrotik, virtuální servery či virtuální počítače. Tato zařízení lze poté konfigurovat například přes Putty. Díky webovému prostředí a pomoci IP adresy se lze připojit k emulátoru z více zařízení což umožňuje například práci z domu skrze vzdálenou plochu. EVE-NG může být buďto nainstalován přímo na fyzický hardware, který mu poskytuje nejvíce výkonu, nebo může fungovat jako virtuální zařízení skrze VMware Workstation.

K dispozici má EVE-NG tři uživatelské verze, první verze „Comunity“ je zdarma a byla použita v této bakalářské práci. Druhá verze emulátoru „Professional“ obsahuje snadnější export a import konfigurací do lokálního PC, možnost většího počtu uzlů atd. Třetí verze je určena pro studijní účely a podporuje vytvoření role studentů, kteří mohou nezávisle na sobě pracovat na laboratořích, nese jméno „Learning Center“. [11]

3.3 Wireshark

Program Wireshark je free open-source paketový *sniffer* a protokolový analyzátor sloužící pro analýzu a ladění problémů v počítačových sítích, monitorování datových toků, analýzy jednotlivých paketů a mnoho dalšího. Původní název programu byl Etheral ale v květnu roku 2006 byl přejmenován kvůli problémům s ochrannými známkami. Program lze nainstalovat do běžných OS jako je například Windows, Linux, Mac atd. Využívá se jak ve školách pro výukové účely, tak v laboratořích pro monitorování různých druhů sítí a jejich chování.

Ve Wiresharku lze nastavit řadiče síťového rozhraní do promiskuitního režimu, který umožňuje zachycení veškerého provozu na daném rozhraní, multicast i broadcast provoz. Dále program dokáže pečlivě rozebrat strukturu zapouzdřeného paketu a rozlišit informace jednotlivých protokolů. Zachycené pakety lze analyzovat přímo z živé sítě nebo z uložené relace, k tomu Wireshark využívá knihovnu *pcap*.

Pakety lze procházet pomocí GUI, ve kterém je možno vygenerovat grafy různých druhů paketů, použít filtry ke snadnějšímu třízení dat a obsahuje mnoho dalších analyzátorských funkcí. Existuje taktéž verze programu pro příkazový řádek jménem TShark. [10]

3.4 Iperf

Program Iperf je běžně používaný nástroj pro testování sítě, který umožňuje vytvářet datové toky TCP a UDP, měřit propustnost sítí, jitter a ztrátovost UDP paketů. Iperf umožňuje uživateli nastavit různé parametry, které lze použít pro testování sítě, nebo pro optimalizaci a ladění sítě. Program má funkcionalitu klienta a serveru a umožňuje měřit propustnost mezi dvěma konci, to buď jednosměrně, nebo obousměrně. Jedná se o software Open-Source, který lze spouštět na různých platformách, včetně systémů Linux, UNIX a Windows. [12]

4. CÍLE LABORATORNÍCH ÚLOH

Cílem první laboratorní úlohy bude srovnání QoS frontových mechanik v kombinaci směrovacích protokolů RIP a OSPF na jednoduché síti se třemi směrovači a skupinou koncových zařízení. V síti bude simulován provoz služeb operujících v reálním čase se službami FTP a HTTP, současně bude simulován výpadek jedné linky mezi směrovači a bude analyzováno chování sítě. Budou zjištěny reakce směrovacích protokolů při výpadku linky a množství zahozených paketů po obnovení spojení při použití různých frontových mechanik obsluhy paketů.

Výstupem této laboratorní úlohy budou grafy zobrazující zahazování IP paketů v síti ze scénářů pro frontové mechanismy FIFO, PQ a WFQ s použitím nejprve směrovacího protokolu RIP, poté OSPF. Tyto grafy budou součástí vypracovaného protokolu spolu se zodpovězenými otázkami v závěru protokolu.

Cílem druhé laboratorní úlohy bude podobně jako již v první úloze srovnání QoS frontových mechanik FIFO, PQ a WFQ spolu se směrovacími protokoly RIP a OSFP, rozdílem bude vytvoření podobné sítě s emulovanými směrovači a virtuálními stroji Linux v prostředí EVE-NG. V tomto emulačním prostředí by se prvky sítě měly chovat reálněji než v simulátoru Riverber, proto zde bude cílem potvrzení či vyvrácení teoretických poznatků z první laboratorní úlohy a případně i získání nových poznatků.

Výsledkem druhé laboratorní úlohy budou data z UDP a TCP testů mezi virtuálními stroji a grafy zobrazující celkový provoz, TCP/UDP provoz a směrovací data přijata na koncovém směrovači jednoho virtuálního stroje. Síti se bude skládat ze scénářů pro frontové mechanismy FIFO, PQ, WFQ, podobně jako v první úloze s využitím směrovacího protokolu RIP, poté protokolu OSPF. Tyto grafy budou součástí vypracovaného protokolu spolu v závěru protokolu.

Laboratorní úlohy studentům poskytnou nové, nejprve teoretické a pak i praktické, pohledy na společné fungování směrování se zajištěním kvality služeb a poukáže na možné výhody ale i problémy konfigurace těchto dvou technologií.

4.1 Návrhy sítí a metody proměření úloh

V první laboratorní úloze bude navrhovaná síť nejprve sestavena a testována v simulátoru Riverbed, bude sestavena ze tří směrovačů (Směrovač 1,2,3) podporujících potřebné technologie směrování a QoS a kvůli lepší přehlednosti provozovaných služeb bude síť obsahovat i vícero koncových zařízení. Tato koncová zařízení budou vybrána a pojmenována podle toho, jaké služby budou provozovat. V simulačním prostředí bude provozován síťový provoz služeb FTP, HTTP, VoIP a Video Conferencing. Pakety těchto služeb budou přiděleny do jednotlivých tříd a bude jim přidělen kód DSCP. Dále bude v simulaci nakonfigurován směrovací protokol RIP poté OSPF a nastaven výpadek mezi směrovači 1, 2.

Síť bude v prvním scénáři testována v konfiguraci směrovacího protokolu RIP, výpadku linky a obsluhy paketů frontou FIFO. Po skončení testování bude duplikován scénář a upraven pro obsluhu paketů frontu PQ, dále pak WFQ. Směrovací protokol RIP bude tedy použit ve třech scénářích, pak bude jeden ze scénářů duplikován a přenastaven na protokol OSPF. V konfiguraci s OSPF protokolem budou otestovány všechny frontové mechanismy podobně jako v předešlých třech scénářích.

Druhá laboratorní úloha v emulačním prostředí bude obsahovat pětici směrovačů Cisco IOS 3725 a dvě koncová Linuxová zařízení, která budou zastávat funkci klienta a serveru. Dva směrovače budu tvořit vstup do laboratorní sítě pro koncová zařízení, zbylé tři směrovače budou součástí sítě a všechny společně utvoří jednu hlavní a druhou redundantní linku mezi koncovými zařízeními. Na směrovačích budou nakonfigurovány směrovací protokoly a QoS, koncová zařízení budou obsahovat testovací program pro zatížení sítě.

Nejprve bude vytvořena a nakonfigurována síť pro první scénář, ve které bude použito RIP protokolu a fronty FIFO, ta bude poté dvakrát zkopírována a přenastavena pro zbylé dva druhy front PQ a WFQ. Nakonec všechny tři scénáře budou zkopírovány a v nich pak bude provedeno přenastavení na směrovací

protokol OSPF. Celá laboratorní úloha se tak bude skládat stejně jako první úloha ze šesti scénářů, které budou kombinací routování a QoS.

Testování všech scénářů bude provedeno skrze koncová zařízení, mezi kterými bude spuštěn test TCP a UDP, který bude zároveň sloužit jako zatížení sítě. Během testu bude vypnut jeden směrovač na hlavní lince a bude tak simulován výpadek linky, po zapojení redundantní linky bude po určitou dobu probíhat dále test TCP a UDP, poté bude ke konci testu zapojen opět směrovač na hlavní lince, což povede k obnově provozu na lince. Celkový průběh testu bude zaznamenáván na koncových zařízeních a jednom směrovači spojující server se sítí.

5. PRVNÍ LABORATORNÍ ÚLOHA V PROGRAMU RIVERBED

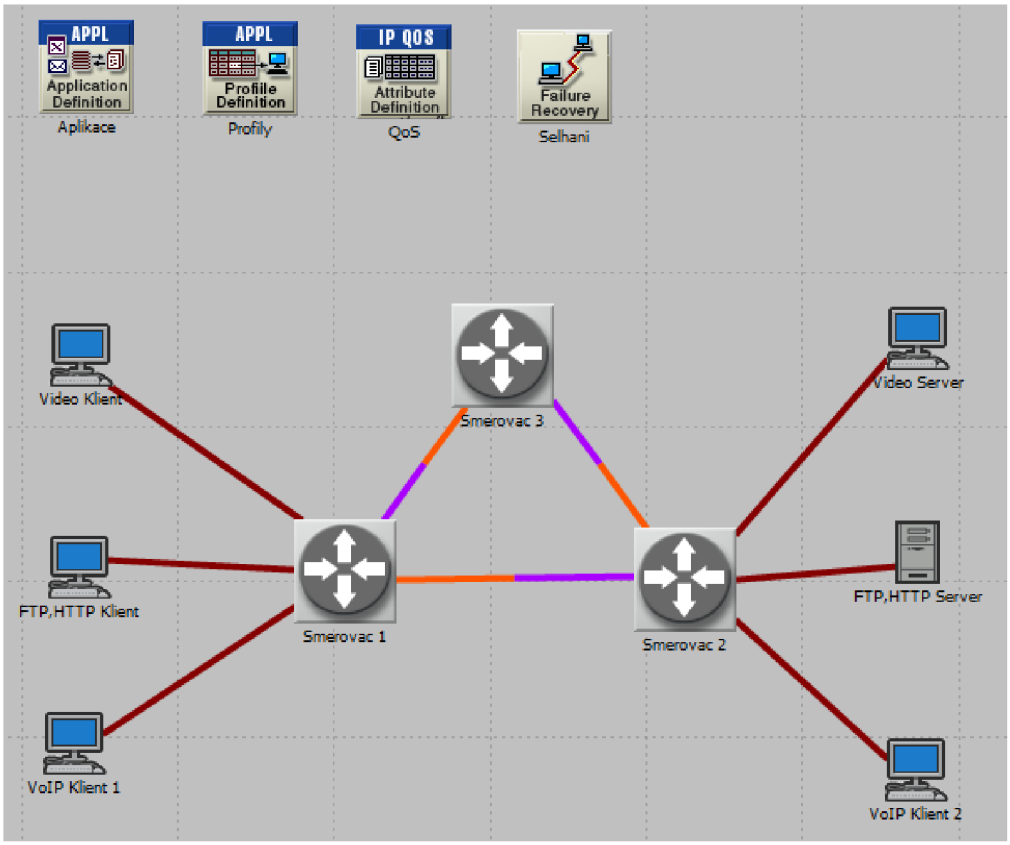
Tato kapitola pojednává o vytvoření první laboratorní úlohy. Je zde nejprve popsáno samotné sestavení sítě v programu, nastavení směrovacích protokolů, příprava zátěžového provozu služeb, dále příprava výpadku v síti, nastavení QoS a nakonec samotný průběh laboratorní úlohy.

5.1.1 Sestavení sítě

Po spuštění programu Riverbet Modeler Academic Edition byl nejprve založen projekt s názvem „privatni site“. To se provedlo vybráním File -> New -> OK. Poté byl projekt pojmenován již zmíněným názvem, scénář byl pojmenován jako „RIP FIFO“. Dále byla vybrána rozloha simulované sítě na Campus, třikrát zmáčknuto tlačítko Next a jednou dokončovací tlačítko Finish.

V okně Object Palette Tree byly vybrány a umístěny tři prvky ethernet4_slip8_gtwy, pět prvků ethernet_wkst, jeden ethernet_server a poté po jednom Application Config, Profile Config, QoS Attribute Config a Failure Recovery. Prvky ethernet4_slip8_gtwy byly vzájemně propojeny linkami PPP-DS1 s malou datovou propustností. Všechny prvky ethernet_wkst spolu s prvkem

ehternet_server byly připojeny k prvkům ethernet4_slip8_gtwy linkami 10Base-T. Všechny prvky byly uspořádány a pojmenovány tak jako na obr. 5.1.



Obr. 5.1 Topologie laboratorní úlohy

5.1.2 Nastavení směrovacích protokolů RIP a OSPF

Pro globální nastavení směrovacího protokolu RIP bylo kliknuto na ikonu Configure/Run DES v Global attributes, byla rozkliknuta položka IP, byly v ní změněny atributy IP Dynamic Routing Protocol na RIP a IP Interface Addressing Mode na Auto Addressed/Export. V položce Simulation Efficiency byla změněna atributa RIP Sim Efficiency na Disabled. Celé nastavení bylo poté potvrzeno tlačítkem Apply.

Později v úloze byl směrovací protokol RIP změněn na protokol OSPF. To bylo provedeno podobně jako u nastavení RIP jen s tím rozdílem, že nejprve byl duplikován jeden ze scénářů s RIP protokolem, ten byl poté zrušen a nastaven protokol OSPF.

5.1.3 Nastavení provozovaných služeb

Pro nastavení provozu služeb v síti je nejprve nutné nastavit potřebné aplikace v objektu Application Config. K tomuto nastavení bylo kliknuto pravým tlačítkem myši na objekt a vybrána možnost Edit Attributes. Po rozkliknutí položky Application Definition byl vybrán potřebný počet aplikací nastavením položky Number of Rows na hodnotu 4. Do kolonek Name byly vepsány názvy jednotlivých aplikací a v položce Description byly zvoleny typy jednotlivých aplikací a jejich konfigurace. Například pro službu FTP byla aplikace pojmenována jako FTP AP v položce Description byla zvolena v kolonce Ftp hodnota High Load, ta byla ještě upravena kliknutím na tuto hodnotu a zvolena možnost Edit. V nově otevřeném okénku byla přiřazena značka DSCP (CS0) po rozkliknutí Type of Service.

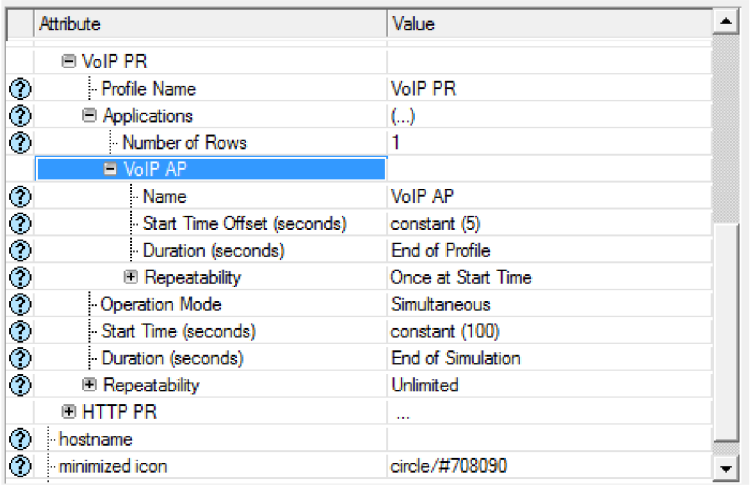
Podobným způsobem byly nastaveny i zbylé aplikace, značky DSCP a hodnoty Description pro jednotlivé aplikace jsou vyznačeny v tabulce tab. 5.1.

Tab. 5.1: DSCP hodnoty aplikací

Aplikace	Description	DSCP
FTP	High Load	CS0 = BE
Video	Low Rsolution	AF41
VoIP	PCM Quality Speech	AF31
HTTP	Video Browsing	AF21

Po nastavení aplikací v Application Config bylo nutné v objektu Profile Config přiřadit profily těmto aplikacím. Ten slouží k určení chování aplikací například ke stanovení okamžiku, kdy bude spuštěna požadovaná aplikace, jak dlouho bude trvat a zda se bude opakovat. Profily pro jednotlivé aplikace byly nastaveny tak, že bylo kliknuto pravým tlačítkem myši na objekt Profile Config a zvolena možnost Eddit Attributes. V nově otevřeném okně byl v položce Profile Configuration zvolen potřebný počet profilů odpovídající počtu aplikací. Jednotlivé profily byly pojmenovány podle aplikací a tyto aplikace jim byly přiřazeny. Dále byly atributy

profilů nastaveny dle požadavků na jejich funkčnost. Příklad nastavení profilu je na obrázku 5.2.



Obr. 5.2 Profil VoIP

Vytvořené aplikace s hotovými profily bylo nutno přiřadit jednotlivým koncovým zařízením, která je budou provozovat a vytvoří tak potřebné datové toky v síti. Přiřazení pro servery bylo provedeno tak, že byl vybrán potřebný server a bylo na něj kliknuto pravým tlačítkem myši a zvolena možnost Eddit Attributes. Například pro FTP, HTTP Server po kliknutí na Eddit Attributes byla v okně rozkliknuta položka Application: Suporter Services, dále pak byly vybrány potřebné aplikace ve sloupci Service Name. Podobné nastavení bylo nastaveno i pro Video Server a VoIP Klient 2, ten však nese stejné klientské nastavení jako VoIP Klient 1.

Pro nastavení příslušných profilů aplikací pro klienty bylo taktéž kliknuto pravým tlačítkem myši na nastavovaný prvek v síti a zvolena možnost Eddit Attributes, dále byla rozkliknuta podsložka Application: Supported Profiles a nastaven potřebný počet profilů podle hodnoty Rows. V položce Profile Name byly zvoleny jednotlivé a již připravené profily.

5.1.4 Nastavení výpadku linky

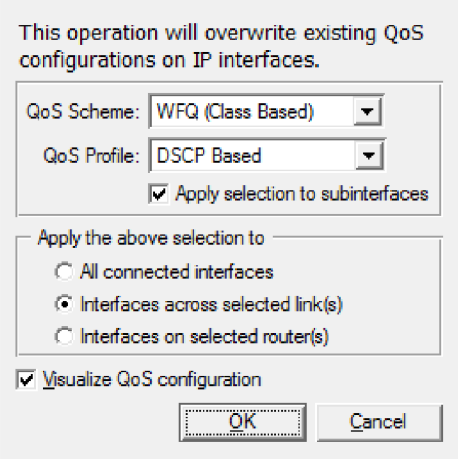
Výpadek linky je vhodná událost ke zjištění chování sítě, směrovacích protokolů a obslužných frontových mechanismů během problematických situací. K nastavení výpadku mezi směrovači 1 a 2 bylo využito objektu Failure Recovery. Bylo kliknuto

pravým tlačítkem myši na tento objekt, vybrána možnost Eddit Attributes, rozkliknuta položka Link Failure/Recovery Specification a zvoleno jedno selhání nastavením hodnoty Number of Rows na 1. Poté bylo zvoleno toto selhání mezi směrovači 1 a 2 vybráním potřebné položky v kolonce Name a doba selhání byla nastavena hodnotou Time na 140 vteřin, status byl ponechán v nastavení Fail.

5.1.5 Nastavení front QoS

Pro nastavení obslužné fronty FIFO byly označeny všechny linky mezi směrovači přidržením klávesy Ctrl a levým tlačítkem myši. Poté bylo kliknuto v horní liště na Protocols -> IP -> QoS -> Configure QoS, následně byla nastavena v kolonce QoS Scheme frontová mechanika FIFO a v kolonce QoS Profile byl zvolen profil FIFO.

Další scénáře byly duplikovány z prvního scénáře a upraveny buďto požadovanými frontovými mechanismy nebo jiným směrovacím protokolem. Duplikace byla provedena kliknutím na položku Scenarios v hlavním okně programu a vybráním možnosti Duplicate Scenario. V dalších scénářích byly podobně nastaveny zbylé dvě frontové mechaniky jen s tím rozdílem, že v kolonce QoS Profile byla zvolena vždy možnost DSCP Based. Příklad nastavení je zobrazen na obrázku 5.3.



Obr. 5.3 Nastavení WFQ

5.1.6 Spuštění simulace

Po úspěšném nastavení všech parametrů v simulaci je nutno zvolit, jaká data budou zaznamenána během simulace. Zaznamenávat se budou nejprve data o zahozených IP paketech globálně v síti, další data se budou týkat příjmů směrovacích informací na směrovači číslo 1. Tato data budou následně zpracována do čtyř grafů, dva grafy budou znázorňovat shromážděné údaje ze scénářů s protokolem RIP a další dva grafy zase údaje pro scénáře s OSPF protokolem. Pro zvolení zaznamenávání potřebných dat bylo pravým tlačítkem myši kliknuto kdekoliv na volnou plochu v projektu a zvolena možnost Choose Individual Statistics. V nově otevřeném okně byla rozkliknuta stromová podsložka Global Statistic a následně byla zaškrtnuta položka IP -> Traffic Dropped (packets/secs). Dále byla rozkliknuta další stromová podsložka s názvem Node Statistic a poté zaškrtnuta položka RIP -> Traffic Received (bits/sec) pro scénáře s RIP protokolem a pro scénáře s OSPF protokolem byla zaškrtnuta OSPF -> Traffic Received (bits/sec) položka.

V Project Toolbaru bylo pro spuštění simulace nejprve kliknuto na ikonu běžce (Configure/Run DES) a v položce Duration byl nastaven čas simulace na 5 minut, poté byla spuštěna simulace tlačítkem Run.

6. DRUHÁ LABORATORNÍ ÚLOHA V EMULAČNÍM PROSTŘEDÍ EVE-NG

Tato část se zabývá praktickou laboratorní úlohou, ve které si studenti ověří či vyvrátí teoretické poznatky o směrování s kvalitou služeb z první laboratorní úlohy. Postupně je zde rozebráno sestavení sítě v emulátoru, instalace programu Iperf, nastavení směrovacích protokolů, dále konfigurace QoS, příprava pomocných programů Wireshark, Iperf a nakonec samotný průběh měření laboratorní úlohy.

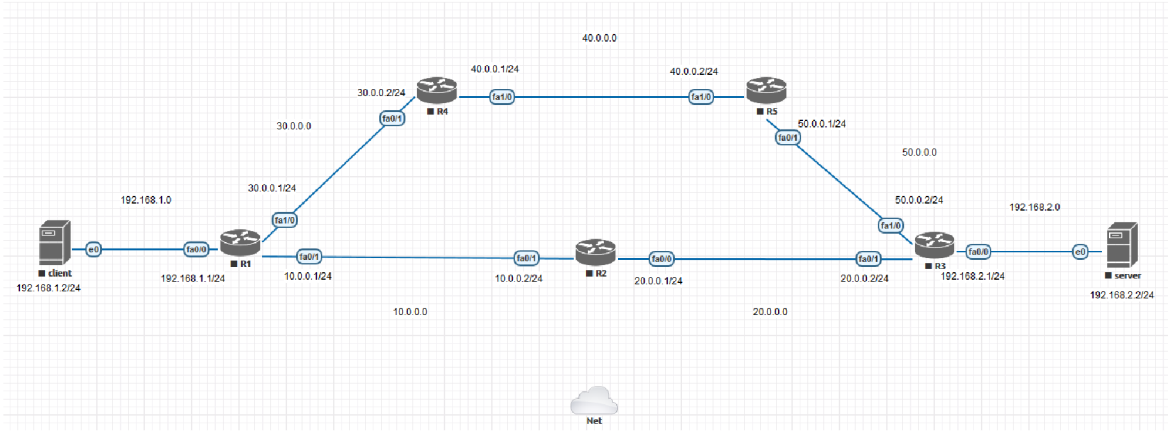
6.1.1 Sestavení sítě

V systému EVE-NG byl nejprve vytvořen první laboratorní scénář kliknutím na ikonu Add new lab. Poté se otevřelo okno s tabulkami pro popis, název, verzi laboratorní úlohy a jméno Autora. Po vyplnění názvu prvního scénáře „Routing RIP QoS FIFO“ bylo kliknuto na tlačítko save, což vedlo k otevření prázdné laboratoře.

Dalším krokem bylo vybrání a vložení pěti směrovačů do pracovní plochy úlohy. To bylo provedeno pravým kliknutím na ikonu plus pod názvem Add an object a vybrána možnost Node. Dále byl ve výběru zařízení vybrán druh směrovačů Cisco IOS 3725 (Dynamips), po zvolení druhu směrovačů bylo vyvoláno okno s detailnějším nastavením směrovačů. V tomto okně byl zvolen potřebný počet směrovačů v kolonce Number of nodes to add vepsáním číslice 5, dále byl zvolen rozšiřující slot NM-16ESW pro rozšíření počtu možných rozhraní ve směrovačích. Nakonec bylo veškeré nastavení potvrzeno tlačítkem Save, což vedlo k vygenerování směrovačů do laboratorní úlohy.

Podobným způsobem byly vygenerovány Linuxové servery, tedy znovu kliknutím na ikonu Add an object -> Node -> Linux. V detailnějším nastavení byl počet zařízení vybrán na dva v kolonce Number of nodes to add a ve výběru Image byl zvolen typ serveru linux-kali-large-2019.0. Nakonec následovalo potvrzení a vygenerování Linux serveru taktéž tlačítkem Save.

Nakonec kvůli instalaci programu Iperf na zařízení Linux byl přidán do laboratoře objekt Net, který poskytoval připojení do Internetu. To bylo provedeno tak, že bylo znovu kliknuto na možnost Add an object ale tentokrát byla vybrána možnost Network, to vyvolalo okno pro detailnější nastavení objektu. V tomto okně byla vybrána položka Managment(Cloud0) v kolonce Type, poté následovalo tlačítko Save.



Obr. 6.1: Topologie laboratorní úlohy

Všechny vygenerované objekty byly uspořádány do topologie tak, jak je uvedeno na obrázku 6.1. Objekty byly pojmenovány tak, jako na obrázku 6.1, sérií kroků: najetí myší na objekt, kliknutím pravým tlačítkem myši a zvolením možnosti Edit, po zobrazení editovacího okna bylo zařízení příslušně pojmenováno a potvrzeno tlačítkem Save.

Pro přehlednost byly přidány IPv4 adresy jednotlivých rozhraní na každém zařízení. Pro přidání těchto adres do pole laboratoře bylo kliknuto pravým tlačítkem myši na prázdné místo laboratoře a zvolena možnost Text, pak byla vepsána příslušná IPv4 adresa do kolonky Text v nově vyvolaném okně a potvrzení textu tlačítkem Save.

Posledním krokem sestavení sítě bylo propojení všech zařízení na pracovní ploše laboratoře. Tento krok byl proveden najetím myši na zařízení a kliknutím na oranžovou ikonu propojení a jejím tažením k příslušnému zařízení, dále pak puštěním tahu nad zařízením, což vyvolalo nové okno pro zvolení obou rozhraní mezi zařízeními. Ta byla zvolena tak, jako jsou na obrázku 6.1. Před dokončením celého propojení však bylo nutné nainstalovat program Iperf na Linuxová zařízení, provedení této instalace je popsáno v další podkapitole.

6.1.2 Instalace programu Iperf

Instalace programu Iperf na Linuxová virtuální zařízení byla provedena v následujících krocích: obě zařízení byla nejprve připojena na objekt Net najetím myši na zařízení, chycením oranžové ikony připojení a tažením stisknuté myši

k objektu Net. Po puštění myši bylo vybráno rozhraní e0, kterým se připojilo zařízení na objekt v nově otevřeném okně a výběr byl potvrzen stisknutím Save.

Po připojení obou zařízení k objektu Net byla zařízení spuštěna najetím myši, kliknutím pravého tlačítka a vybráním možnost Start. Dále bylo na Linuxová zařízení kliknuto levým tlačítkem a byly vyvolány jejich pracovní plochy, potom pokračovalo přihlášení zadáním uživatelského jména root a hesla toor. Po přihlášení bylo vyvoláno příkazové okno kliknutím vlevo na ikonu Terminal. Ve vyvolaném okně proběhla instalace programu Iperf zadáním příkazu apt-get install iperf.

Dalším krokem bylo po dokončení instalace programu vypnutí Linuxových zařízení a jejich přepojení do sítě směrovačů, to však lze provést pouze, když jsou zařízení vypnuta. Proto byly nejprve zavřeny pracovní plochy obou zařízení, pak následovalo jejich vypnutí klikem pravého tlačítka a vybráním možnosti Stop. Přerušení spojení s objektem Net bylo provedeno najetím na příslušnou linku, kliknutím pravým tlačítkem myši a vybráním možnosti Delete. Obdobnou sérii kroků, jako při připojení zařízení k objektu Net, byla zařízení připojena ke směrovačům R1 a R3. Ve výběru rozraní byla vybrána e0 pro Linuxová zařízení a fa0/0 pro směrovače.

6.1.3 Nastavení IP adres a směrovacího protokolu RIP

Prvním krokem pro nastavení bylo zapnutí a pojmenování všech směrovačů, to bylo provedeno označením všech směrovačů myší, kliknutím pravým tlačítkem na libovolný směrovač a vybrání možnosti Start Select. Dále byla dvojklikem například na směrovač R1 otevřena konfigurační konzole, v této konzoli bylo nastaveno pojmenování zařízení sérií příkazů: enable -> conf terminal -> hostname R1. Příkaz enable slouží k přepnutí do druhé konfigurační úrovně zařízení a conf terminal příkaz pak do úrovně třetí, ve které je možno nastavit směrování, IP adresy, jméno zařízení a mnoho dalšího. Stejným způsobem byl pojmenován i zbytek směrovačů jen s rozdílnými čísly.

Druhým krokem bylo nastavení IP adres na každém rozhraní u každého směrovače příkazy: interface FastEthernet(číslo rozhraní) -> ip address(číslo ip adresy s maskou). U směrovačů R1, R4 a R5 bylo nutné nejprve přepnout rozhraní FastEthernet1/0 příkazem no switchport do režimu L3, známy jako „směrový port“

pro zajištění správného fungování rozhraní a nastavení ip adresy. Na jednotlivých rozhraních byly přidány i další nastavení rozhraní, prvním bylo nastavení rychlosti na rozhraní příkazem speed 10, druhým pak bylo přepnutí rozhraní do plného duplexu příkazem full-duplex. Posledním příkazem byl příkaz no shutdown, kterým se zakázalo vypnutí daného rozhraní. Příklad nastavení směrovače je zobrazen na obrázku 6.2.

```
enable
conf terminal
hostname R1

interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 speed 10
 full-duplex
 no shutdown
 exit

interface FastEthernet0/1
 ip address 10.0.0.1 255.255.255.0
 speed 10
 full-duplex
 no shutdown
 exit

interface FastEthernet1/0
 no switchport
 ip address 30.0.0.1 255.255.255.0
 duplex full
 speed 10
 no shutdown
 exit
```

Obr. 6.2: Nastavení rozhraních směrovače R1

Nastavení směrovacího protokolu RIP proběhlo ve třetím stupni konfigurace směrovačů v terminálu. Nejprve bylo zapnuto směrování protokolem RIP příkazem router rip, dále byla nastavena druhá verze protokolu příkazem version 2 a nakonec byly přidány síťové adresy, na kterých měl směrovací protokol pracovat. To bylo provedeno příkazem network (ipv4 adresa sítě maska sítě). Celá série příkazů je zobrazena na obrázku 6.3.

```
router rip
 version 2
 network 10.0.0.0 255.255.255.0
 network 30.0.0.0 255.255.255.0
 network 192.168.1.0 255.255.255.0
```

Obr. 6.3: Nastavení protokolu RIP na R1

Později v úloze byl směrovací protokol RIP změněn na protokol OSPF. To bylo provedeno podobně jako u nastavení RIP jen s tím rozdílem, že nejprve byl duplikován jeden ze scénářů s RIP protokolem, ten byl poté zrušen a nastaven protokol OSPF.

6.1.4 Nastavení front QoS

První z testovaných QoS front je fronta FIFO, která odesílá pakety v tom stejném pořadí, v jakém je směrovač příjme. Na použitých laboratorních směrovačích Cisco je tato funkce nastavena v základu a není ji tedy nutné nastavovat, to si lze ověřit ve výpisu nastavení na daných rozhraních příkazem show interfaces. Příklad takového výpisu je na obrázku 6.4 výpisu z prvního směrovače.

```
R1#sh interfaces
FastEthernet0/0 is up, line protocol is up
  Hardware is Gt96k FE, address is c201.5910.0000 (bia c201.5910.0000)
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 10Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:04, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
```

Obr. 6.4: Výpis rozhraní směrovače R1

Pro otestování typu QoS fronty PQ bylo zapotřebí nejprve vytvoření kopie laboratorní úlohy a poté již nastavení samotného PQ ve druhé laboratorní úloze. Aby bylo možné vytvořit kopii laboratoře a odpadla nutnost všechny směrovače konfigurovat od začátku, bylo nejprve nutné uložit běžící konfigurace do spouštěcí konfigurace na všech směrovačích. To bylo provedeno příkazem cpoy running-config startup-config, déle byly konfigurace směrovačů vyexportovány do systému EVE-NG, jakožto počáteční startovací konfigurace směrovačů. Všechny běžící směrovače byly vybrány a pravým tlačítkem myši bylo kliknuto na jeden směrovač

a byla vybrána možnost Export CFG. Tak byly zachovány konfigurace i po zavření laboratoře a mohly být zkopírovány spolu s celou laboratoří.

K vytvoření kopie laboratoře bylo nutno vypnout všechny směrovače, to bylo provedeno podobně jako jejich zapnutí, jen s tím rozdílem že byla vybrána možnost Stop Selected. Taktéž byly vypnuty i Linuxová zařízení a v systému EVE-NG byla laboratoř vypnuta kliknutím na možnost Close lab. Uzavření laboratoře vedlo do návratu správy laboratoří, kde bylo myší najeto na název právě uzavřené laboratoře a byla vybrána možnost clone. Po vytvoření kopie laboratoře byla laboratoř přejmenována na „Routing RIP QoS PQ“ a poté byla otevřena.

Dalším krokem bylo spuštění všech směrovačů a nastavení mechaniky PQ postupně na všech směrovačích. Po otevření terminálu na směrovači bylo opět přepnuto do třetí úrovně konfigurace směrovače již zmíněnou sérií příkazů, pro nastavení mechaniky PQ bylo nejprve zapotřebí vytvoření seznamu priorit, které byly poté uplatněny na jednotlivá rozhraní směrovačů. Vytvoření priorit je zobrazeno na obrázku 6.5.

```
priority-list 1 protocol ip high tcp 5001
priority-list 1 protocol ip high udp 5002
priority-list 1 protocol ip low udp rip
priority-list 1 queue-limit 80 60 40 20
```

Obr. 6.5: Nastavení priorit pro scénáře PQ

Z obrázku je patrné nastavení vyšší priority pro TCP a UDP uživatelská data identifikovaná podle čísel portů a nastavení nejnižší priority pro všechna RIP data. Tím mělo být umožněno podpoření teorie vycházející z výsledků první laboratorní úlohy, která naznačovala ohrožení funkce směrování špatným nastavením funkce QoS. Na obrázku je patrný i poslední příkaz, kterým se nastavily velikosti jednotlivých front v mechanice PQ. Tedy 80 paketů pro frontu High, 60 paketů pro Medium frontu, 40 paketů pro frontu Normal a 20 paketů pro Low frontu. Toto nastavení mělo taktéž podpořit horší fungovaní směrovacího protokolu.

Dále už jen zbývalo nastavení uplatnění priorit na jednotlivá rozhraní směrovačů, to bylo provedeno po přepnutí do rozhraní příkazem priority-group 1.
Správné nastavení mechaniky PQ na jednotlivých rozhraních bylo ověřeno již jednou zmíněným příkazem show interfaces viz. Obr. 6.6.

```
R1#show interfaces
FastEthernet0/0 is up, line protocol is up
  Hardware is Gt96k FE, address is c201.0566.0000 (bia c201.0566.0000)
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 10Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:04, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: priority-list 1
  Output queue (queue priority: size/max/drops):
     high: 0/80/0, medium: 0/60/0, normal: 0/40/0, low: 0/20/0
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
```

Obr. 6.6: Výpis rozhraní směrovače R1 pro PQ scénáře

Poslední testovanou frontovou mechanikou bylo WFQ. Bylo tedy nutné stejně jako v předešlém případě uložení běžících konfigurací směrovačů do startovacích konfigurací, vyexportování konfigurací do systému EVE-NG, vytvoření kopie laboratoře a její přejmenování na „Routing RIP QoS WFQ“.

Po otevření laboratoře následovalo taktéž spuštění všech směrovačů stejně jako v předešlých případech a poté následovalo nejprve zrušení mechaniky PQ a zavedení mechaniky WFQ. Zrušení PQ bylo provedeno ve třetí úrovni konfigurace směrovače nejprve zrušením prioritních pravidel příkazy uvedenými na obrázku 6.7.

```
no priority-list 1 protocol ip high tcp 5001
no priority-list 1 protocol ip high udp 5002
no priority-list 1 protocol ip low udp rip
no priority-list 1 queue-limit 80 60 40 20
```

Obr. 6.7: Zrušení priorit ve scénářích PQ

Dále bylo zrušeno uplatnění PQ na samotných rozhraních směrovačů nejprve vstupem do nastavení rozhraní a zadáním příkazu no priority-group 1. Následně bylo v nastavení rozhraní zadání příkazu fair-queue 64 512 128 pro uplatnění mechaniky WFQ na daném rozhraní. První číslo určuje prahovou hodnotu paketů, při které začne preventivní zahazování paketů, druhé číslo udává počet dynamických front a poslední číslo určuje počet rezervovaných front. Tento příkaz a zavedení tak mechaniky WFQ byl zadán na všech rozhraních ve všech směrovačích. Ověření uplatnění mechaniky WFQ bylo provedeno znovu příkazem show interfaces, výpis jednoho takto nastaveného rozhraní je zobrazen na obrázku 6.8.

```
R1#sh inter
FastEthernet0/0 is up, line protocol is up
  Hardware is Gt96k FE, address is c201.2a06.0000 (bia c201.2a06.0000)
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 128 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 10Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:03, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/512 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 96 kilobits/sec
```

Obr. 6.8: Výpis rozhraní směrovače R1 pro WFQ scénáře

6.1.5 Nastavení směrovacího protokolu OSPF

Přenastavení ze směrovacího protokolu RIP na protokol OSPF proběhlo ze začátku podobně jako při nastavování QoS mechanik, tedy uložením běžících konfigurací do startovacích konfigurací, které byly taktéž pak vyexportovány do systému EVE-NG, vypnutím všech zařízení a vytvořením kopie laboratoře. To proběhlo u všech tří laboratoří s nastaveným protokolem RIP a poté byly tyto kopie přejmenovány na „Routing OSPF QoS (zkratka QoS mechaniky)“. Každá s těchto laboratoří byla poté překonfigurována na použití směrovacího protokolu OSPF a zrušení RIP protokolu.

V každé laboratoři, ve které měl být použit OSPF protokol bylo nejprve nutno zrušit protokol RIP na každém směrovači, to bylo provedeno, po zapnutí všech směrovačů, ve třetí konfigurační úrovni příkazem no router rip, dále byl příkazem router ospf 1 povolen směrovací protokol OSPF a zároveň bylo vstoupeno do jeho konfigurace. V konfiguraci směrovacího protokolu byla poté definována ip adresa sítě s wildcard maskou, ve které fungoval směrovací protokol na rozhraní směrovače a ID oblasti, to bylo provedeno sérií příkazů jako na obrázku 6.9.

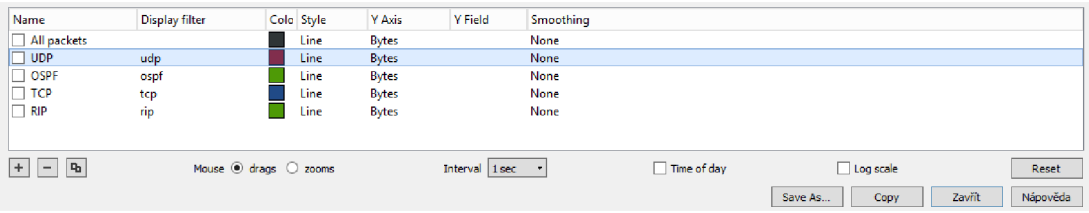
```
router ospf 1
 network 10.0.0.0 0.0.0.255 area 1
 network 30.0.0.0 0.0.0.255 area 1
 network 192.168.1.0 0.0.0.255 area 1
```

Obr. 6.9: Nastavení OSPF protokolu na R1 směrovači

Posledním krokem nastavení protokolu OSPF bylo nastavení ceny cest na rozhraních směrovačů, nejprve bylo vstoupeno do konfigurace jednotlivých rozhraní příkazem interface FastEthernet (číslo rozhraní) a určení ceny cesty použitím příkazu ip ospf cost 10. Cena 10 byla nastavena na každé rozhraní všech směrovačů, stejně jako i ID oblasti tak aby byla použita vždy nejkratší cesta mezi klientem a serverem.

6.1.6 Příprava programů Wireshark a Iperf

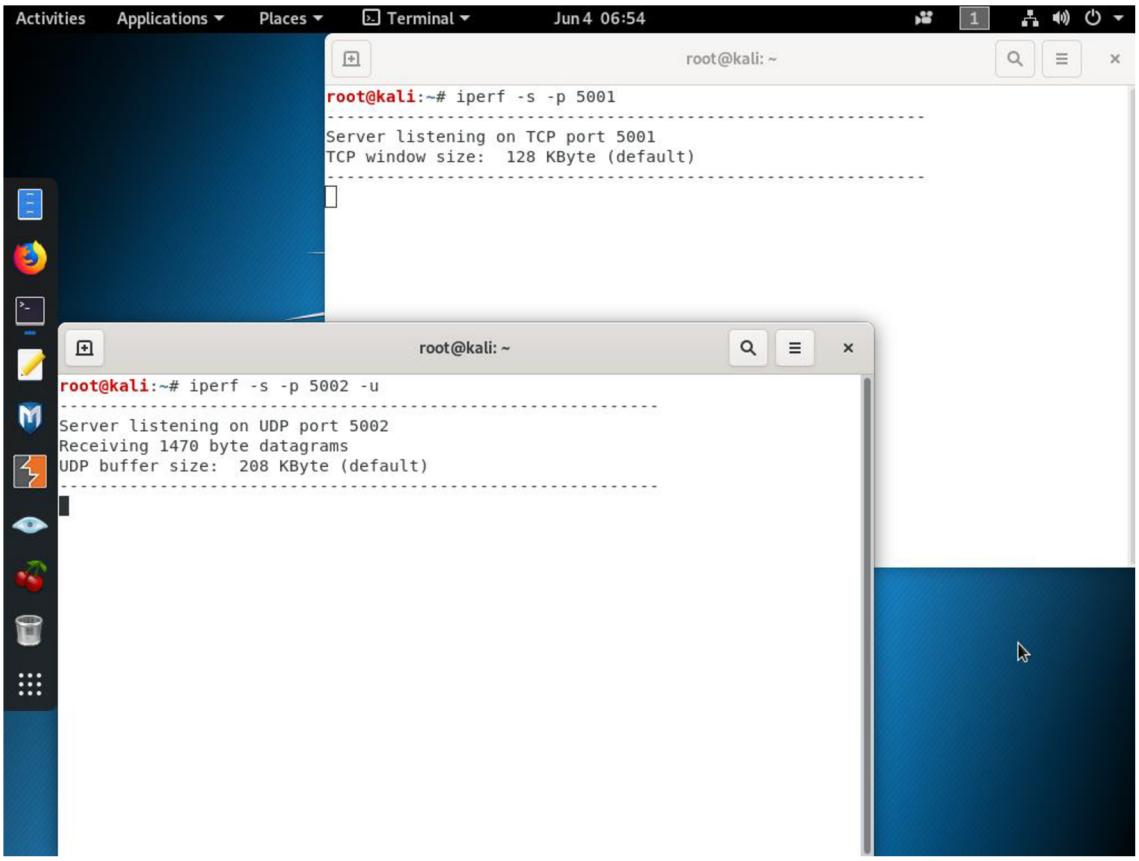
Konečným krokem před otestováním všech šesti laboratoří byla příprava spuštění zátěžového programu Iperf mezi klientem a serverem, nejprve však bylo nutno předpřipravit program Wireshark pro zaznamenávání přijatých dat na směrovači R3 přesněji na jeho rozhraní FastEthernet0/0, které bylo propojeno se serverem. Proto bylo kliknuto pravým tlačítkem myši na R3 směrovač a zvolena možnost Capture a poté vybráno rozhraní fa0/0 to vedlo ke spuštění programu Wireshark ve kterém bylo v horní liště kliknuto na položku Statistics a vybrána možnost I/O Grahp. Tato možnost vyvolala okno s rozhraním pro vytváření grafů, bylo nutno zvolit, jaké grafy budou zobrazeny. Navazujícím krokem bylo kliknutí na ikonu plus v levém dolním rohu pro přidání nového grafu, v seznamu grafů bylo potom možno nastavit v novém grafu, jaká data budou vykreslena. Ve sloupci Name byl každý graf příslušně pojmenován, dále ve sloupci Display filter byl zvolen potřebný filtr grafu a ve sloupci Y Axis byla vybrána jednotka Bytes. Seznam všech grafů je vyobrazeno na obrázku 6.10.



Obr. 6.10: Seznam grafů v programu Wireshark

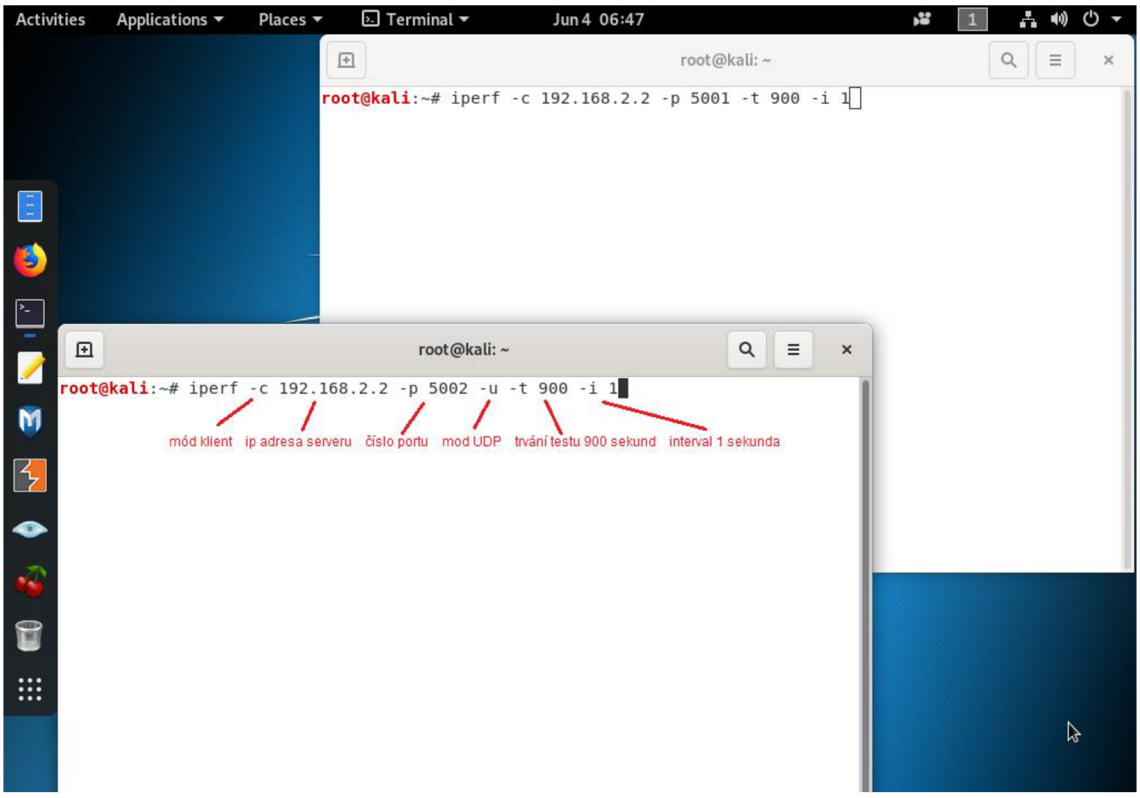
Po úspěšném připravení programu Wireshark následovala příprava testovacího programu Iperf na serveru a klientovi. Proto byla nejprve spuštěna obě Linuxová zařízení a následovalo otevření jejich pracovních ploch stejně, jak již bylo popsáno

výše při instalaci Iperf programu. Dále byly vyvolány dva terminály na každém zařízení, jeden pro TCP test a druhý pro test UDP. Na straně serveru bylo nutno nastavit chování programu Iperf jako server, proto byly zadány do terminálu příkazy iperf -s -p 5001 pro naslouchání Iperfu jako TCP server v prvním terminálu a ve druhém pak iperf -s -p 5002 - u. Písmeno s nastavuje program Iperf jako server, následuje písmeno p a číslo portu pro poslech relace, nakonec písmeno u, které přepíná program do modu UDP. Příklad nastavení zobrazen na obrázku 6.11.



Obr. 6.11: Nastavení programu Iperf na straně serveru

Nakonec byly přichystány příkazy, které jsou vidět na obrázku 6.12 pro spuštění testů přenosu programu Iperf na straně klienta. Tyto příkazy, jak na straně serveru tak i na straně klienta jsou vždy stejné pro všechny laboratoře.



Obr. 6.12: nastavení programu Iperf na straně klienta

6.1.7 Spuštění a průběh úlohy

Testování nastavené laboratorní sítě ve všech laboratořích probíhalo nejprve spuštěním všech směrovačů a zhruba minutové čekání načtení všech konfigurací, dále následovalo již samotné spuštění testů UDP a TCP v terminálech klienta. Průběh testů byl sledován jak výpisy programu Iperf na terminálech, tak i postupným vykreslováním grafů v programu Wireshark.

Podle orientace grafu v čase 125 sekund byl vypnut směrovač R2 tvořící hlavní linku mezi směrovači R1 a R3, došlo tak k přerušení hlavní linky a probíhalo čekání reakce směrovacího protokolu na přepnutí datového toku na vedlejší linku tvořenou směrovači R4 a R5. Po přepnutí na vedlejší linku byl dále průběh datového toku zaznamenáván na směrovači R3 a v čase 700 sekund byl znovu zapnut směrovač R2. To mělo za následek obnovení hlavní linky a přesměrování datového provozu opět na původní linku, dále se už jen čekalo na dokončení testu a ukončení provozu v čase 900 sekund. Tímto způsobem byly otestovány všechny laboratoře a zaznamenané grafické průběhy i výsledky z programu Iperf byly zpracovány do kapitoly s výsledky druhé laboratorní úlohy.

7. VÝSLEDKY PRVNÍ LABORATORNÍ ÚLOHY

Po skončení simulace byly výsledky vygenerovány do grafů. Kliknutím pravým tlačítkem myši na volné pole projektu a zvolením View Results bylo vyvoláno okno Results Browser. Dále byla v kolonce Results for: vybrána možnost Current Project a byly zaškrtnuty všechny scénáře s protokolem RIP. Poté byl vygenerován graf znázorňující globální zahazování IP paketů všech tří scénářů. Tento graf je znázorněn na Obr. 7.1. Byl taktéž vygenerován graf o zahazování IP paketů pro všechny scénáře se směrovacím protokolem OSPF, který je znázorněn na Obr. 7.2.

Na Obr. 7.1 lze pozorovat nejprve obvyklé zahazování paketů při zátěži linky, dále je viditelná prodleva mezi výpadkem spojení mezi směrovači 1 a 2, a přepnutím na záložní linku vedoucí skrze směrovač č. 3. V této prodlevě, která trvala 13 sekund, došlo k velkému zahození paketů ve všech scénářích. Po vytvoření nového spojení došlo k poklesu v zahazování paketů. Důvodem tohoto poklesu je fakt, že před spuštěním datového provozu skrze redundantní linku nebyla tato linka využívána, stejně tak jako zdroje směrovače 3. Těsně po spuštění datového provozu tedy nedocházelo k žádnému zahazování paketů, poté vzrostlo vytížení zdrojů a opět začalo obvyklé zahazování paketů podobně jako před výpadkem. Zahazování paketů ve frontách FIFO a WFQ je téměř stejné, zato u fronty PQ docházelo celkově k většímu zahazování dat, a to z důvodu upřednostnění služby s nejvyšší prioritou.

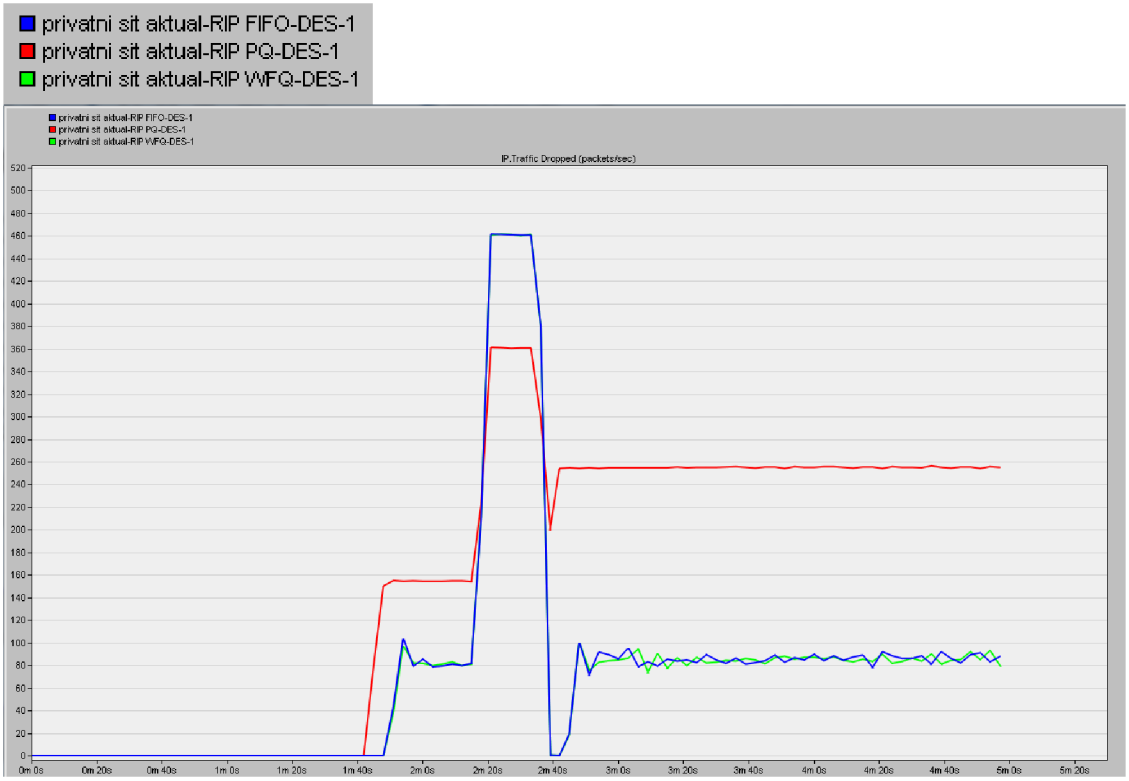
Dále byly vygenerovány grafy znázorňující příjmy směrovacích dat z prvního směrovače. První obrázek 7.3 znázorňuje příjmy směrovacích dat pro jednotlivé RIP scénáře a druhý obrázek. 7.4 znázorňuje data ze scénářů s OSPF protokolem.

Z obr. 7.3 je patrné množství přijatých dat v intervalech před, během i po výpadku linky mezi směrovači 1 a 2. Před výpadkem linky a po výpadku je patrný pokles množství přijatých dat, dále je patrná reakce na výpadek linky začínající v čase 2 minuty a 20 sekund. Ve scénářích s FIFO a WFQ mechanismy je vidět

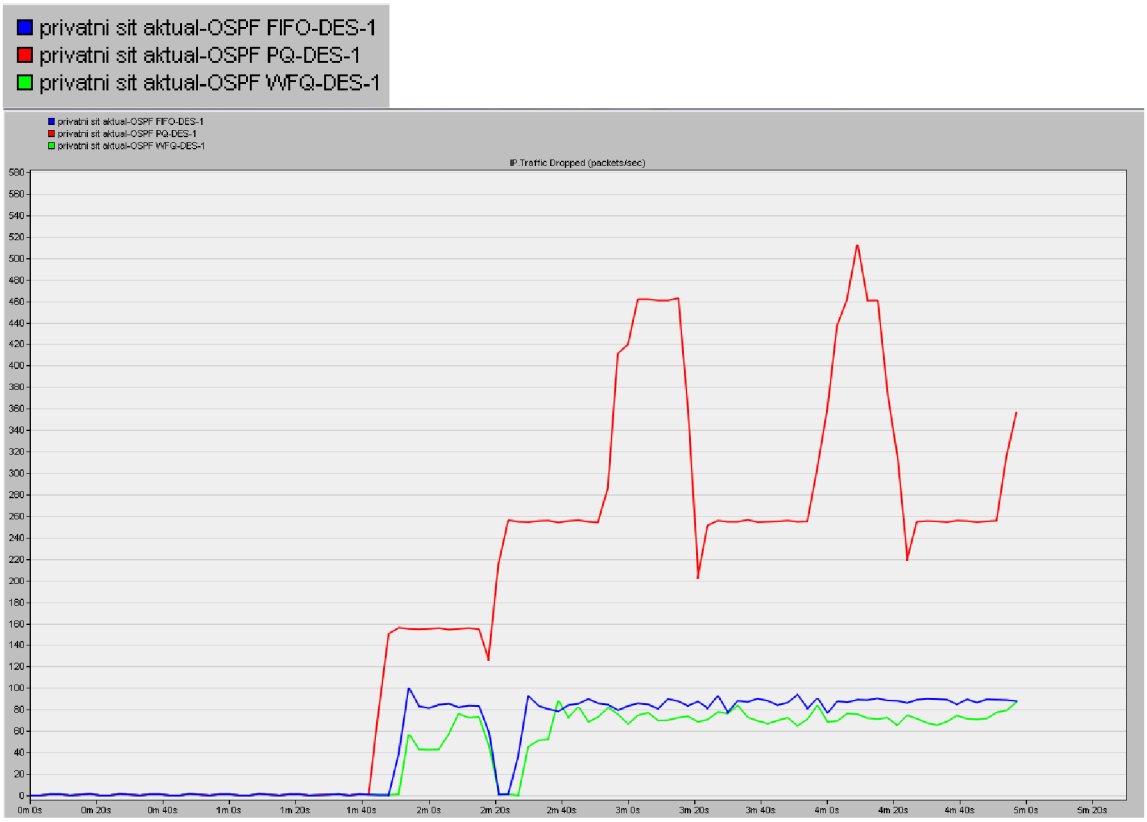
normální fungování RIP protokolu, ale ve scénáři s PQ zhruba od času 3 minuty a 10 sekund směrovač nepřijal žádná směrovací data. Na vině je pravděpodobně mechanika PQ, která obsluhovala jen frontu s nejvyšší prioritou a zbylé fronty nebyly obslouženy, což mělo za následek zahození směrovacích dat. Stav, kdy směrovače nepřijímají potřebná data, je velmi nežádoucí a ohrožuje schopnost reakce sítě na změny v topologii.

Po prozkoumání obr. 7.2 zahazování paketů lze usoudit, že protokol OSPF zareagoval na výpadek linky okamžitě a není patrné zahazování paketů z důvodu výpadku linky. Je zde však více patrný pokles v zahazování paketů po navázání náhradního spojení. Ve scénáři WFQ došlo k celkově menšímu množství zahazování než ve scénáři FIFO. Proto se mechanismus WFQ jeví jako lepší obslužný mechanismus. Dále je patrné, že k největšímu zahazování došlo ve scénáři PQ a během simulace docházelo k nerovnoměrnému zahazování paketů, to je patrné z extrémů v grafu.

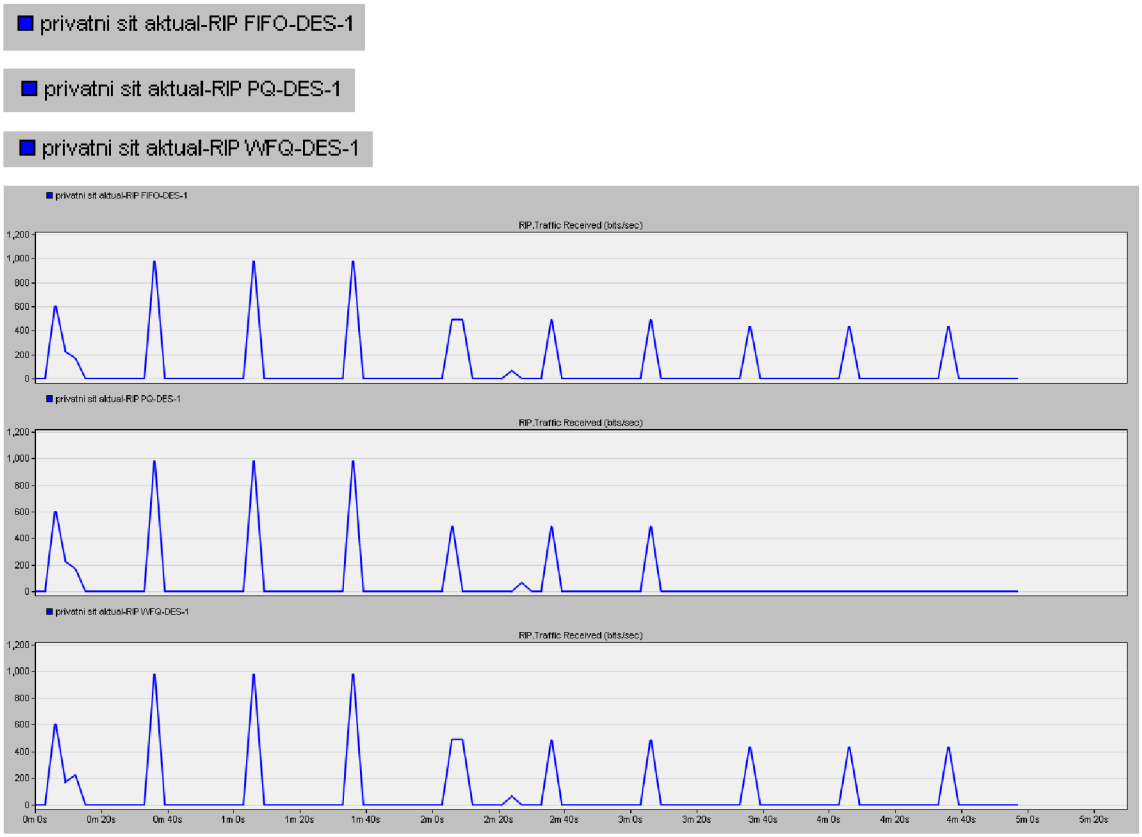
Obrázek 7.4 znázorňující přijímání směrovacích dat směrovače č. 1 vyobrazuje obvyklé fungování protokolu OSPF. Na začátku všech průběhů grafu lze pozorovat nárůst přijímaných dat při sestavování směrovacích tabulek a sousedských vztahů. Dále je vidět přijímání paketů Hello pro udržení vztahů. Ve scénáři s mechanikou FIFO lze pozorovat hladší průběh grafu, to značí malé zahazování směrovacích dat. Zhruba v čase 2:20 lze vidět reakci na výpadek linky a přepnutí na záložní linku. Průběh grafu s PQ mechanikou je podobný jako průběh grafu s RIP protokolem se stejnou mechanikou. I z tohoto grafu je patrné ohrožení funkčnosti směrování, jediným rozdílem je nárazové přijetí směrovacích dat, tedy obsloužení těchto dat mechanismem PQ.



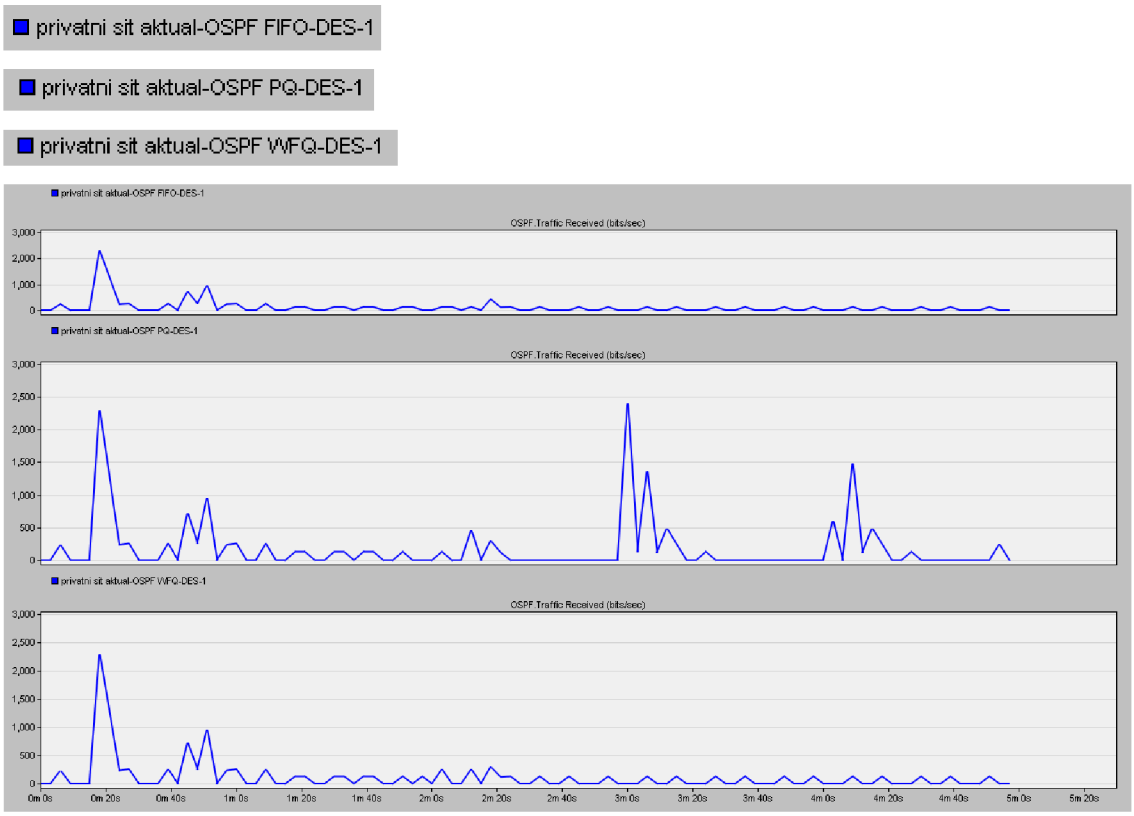
Obr. 7.1 Zahazování IP paketů s použitím RIP protokolu



Obr. 7.2 Zahazování IP paketů s použitím OSPF protokolu



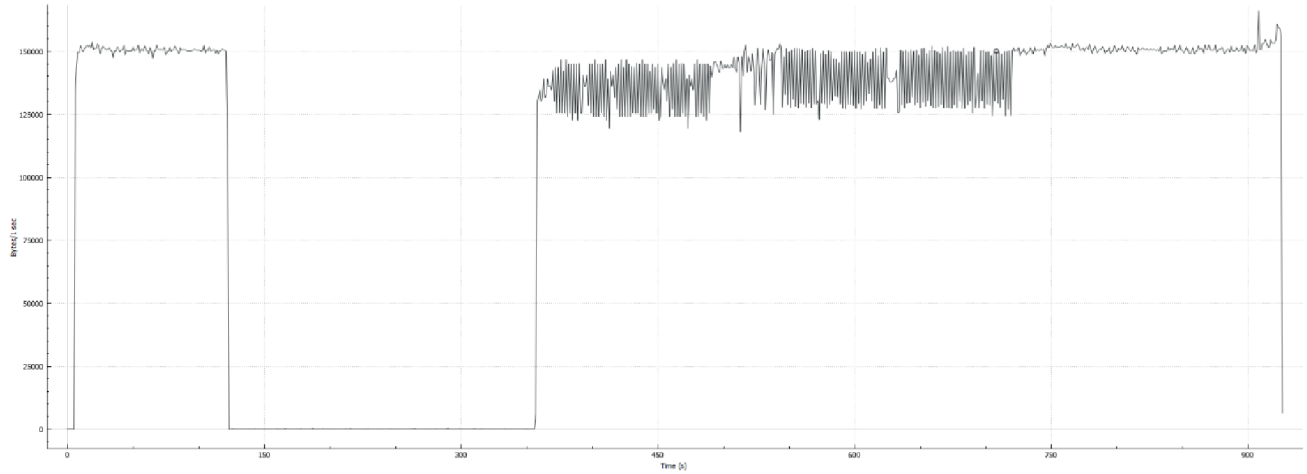
Obr. 7.3 Příjem směrovacích dat protokolu RIP na směrovači 1



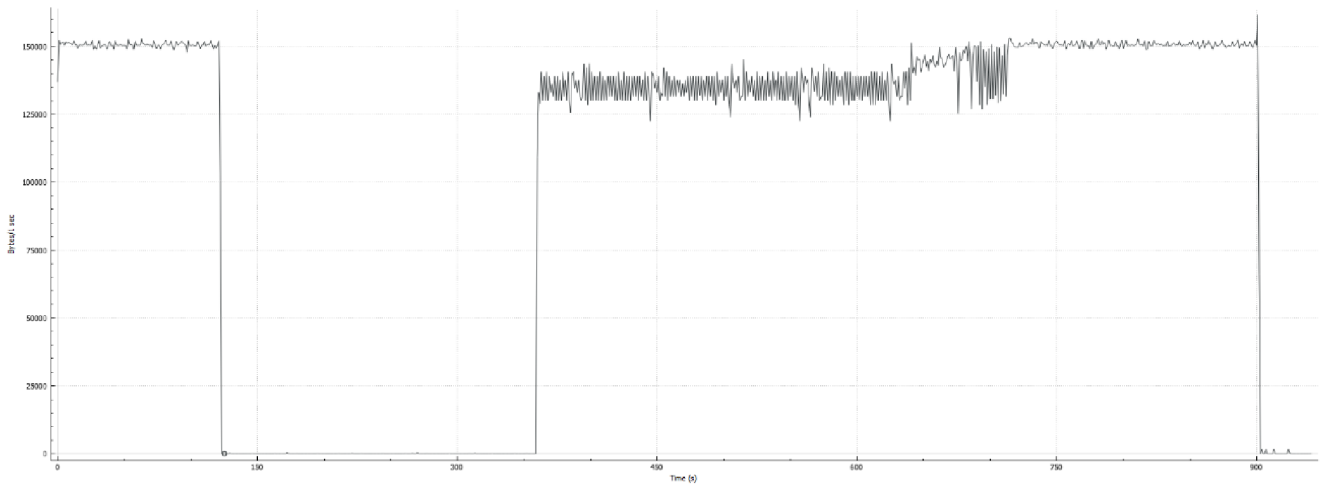
Obr. 7.4 Příjem směrovacích dat protokolu OSPF na směrovači 1

8. VÝSLEDKY DRUHÉ LABORATORNÍ ÚLOHY

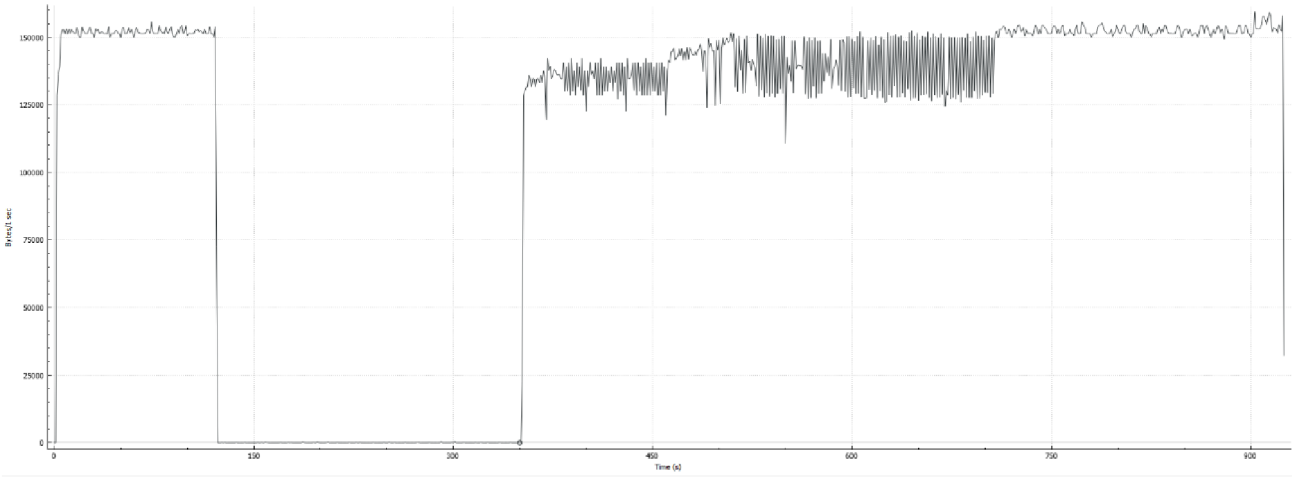
Po dokončení testování sítě byly výsledky ze směrovače R3 v každém scénáři zaznamenány programem Wireshark v němž byly vygenerovány do grafů znázorňující celkový provoz, TCP/UDP provoz a přijatá směrovací data protokolu RIP nebo OSPF. Tyto grafy byly uloženy funkcí Print Screen a výsledky zátěžového programu Iperf byly zaznamenány do tabulky 8.1.



Obr. 8.1: Celkový příjem dat RIP FIFO



Obr. 8.2: Celkový příjem dat RIP PQ

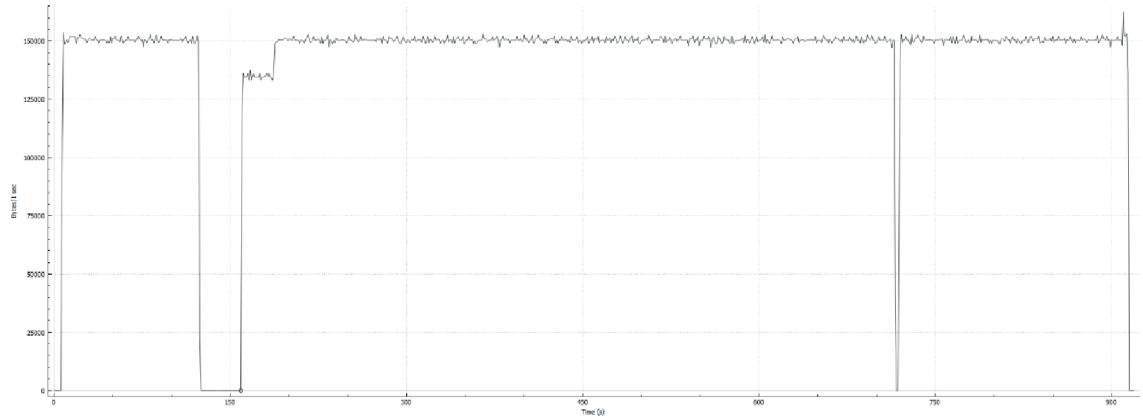


Obr. 8.3: Celkový příjem dat RIP WFQ

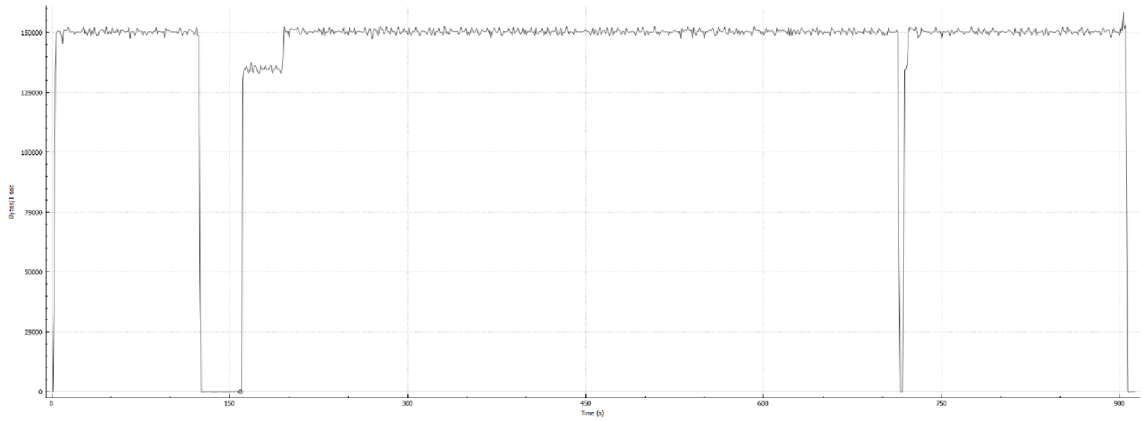
Všechny tři grafy zaznamenávající příjem všech paketů ve scénářích s protokolem RIP na směrovači R3 znázorňují nejprve maximální vytížení linky zátěžovým programem Iperf. V této části lze tak hlavně pozorovat fungování QoS, které ale bude více probíráno u grafů znázorňující UDP a TCP provoz. Dále v čase 125 sekund následuje nulový příjem dat způsobený vypnutím směrovače R2 a přerušení tak hlavní datové linky. Po 240 sekundách, tedy v čase 365 sekund lze pozorovat přepnutí na záložní linku a obnovení datového toku skrze směrovače R4 a R5. Z toho vyplývá normální fungování protokolu RIP a jeho rozesílání směrovacích tabulek po 30 sekundách. Po přepnutí na záložní linku lze vidět nestabilitu provozu ve všech scénářích, nejvíce nestabilní provoz je ve scénáři s frontou FIFO Obr. 8.1, druhým nejvíce nestabilním provozem je scénář s WFQ Obr. 8.3, a nakonec scénář s PQ mechanikou. Nestabilitu pravděpodobně způsobily pokusy protokolu TCP o obnovení spojení v kombinaci s protokolem RIP, jehož pakety jsou větší oproti OSPF paketům, tedy tento protokol kladl na síť větší nároky.

V průbězích všech grafů mezi nestabilními úseky lze pozorovat stabilnější úseky přenosu dat, od okamžiku navázání redundantního spojení fungoval pouze UDP přenos, na začátku stabilnějších úseků byl obnoven i provoz TCP, to však dále mělo za následek větší nestabilitu ve všech scénářích. Po 725 sekundách došlo ve všech scénářích k přepnutí zpět na původní datovou linku důsledkem opětovného zapnutí směrovače R2, to se projevilo stabilnějším přenosem všech dat podobným na

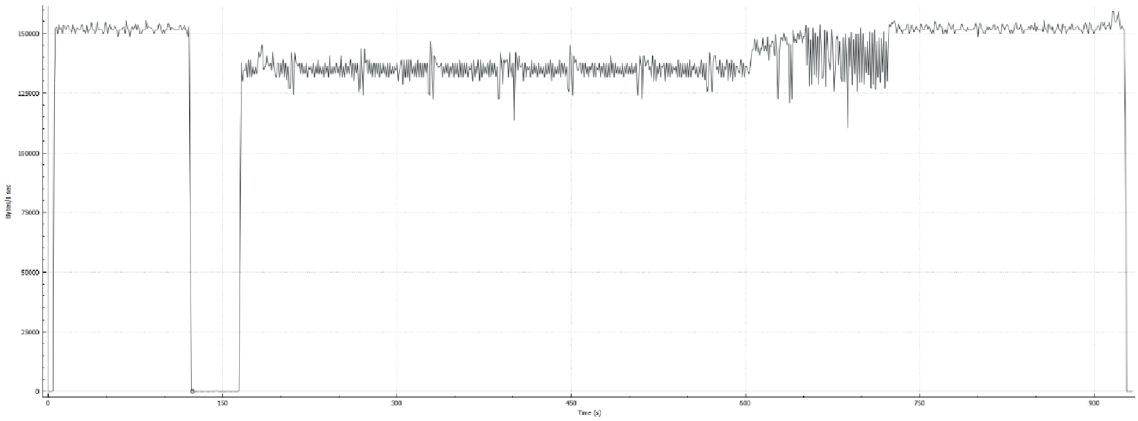
začátku všech grafů. Při porovnání úseků grafů, kdy datový provoz probíhal skrze hlavní linku, tedy začátek a konec grafů, lze pozorovat stabilitu provozu dat v jednotlivých scénářích neovlivněných směrovacími protokoly a přerušením hlavní linky. Je patrné, že scénář s frontovou mechanikou WFQ má nejméně stabilní datový provoz skrze hlavní linku, zbylé dva scénáře mají stabilitu provozu skrz zmíněnou linku téměř totožnou.



Obr. 8.4: Celkový příjem dat OSPF FIFO



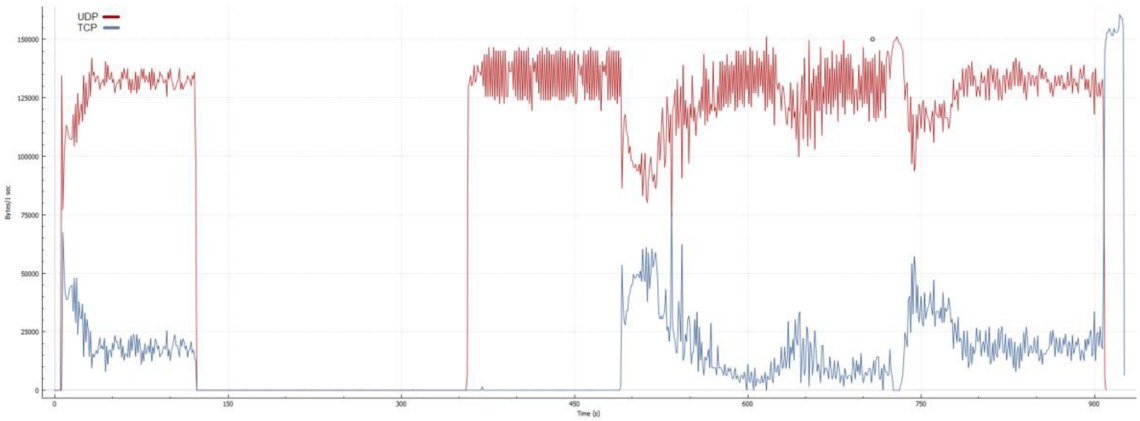
Obr. 8.5: Celkový příjem dat OSPF PQ



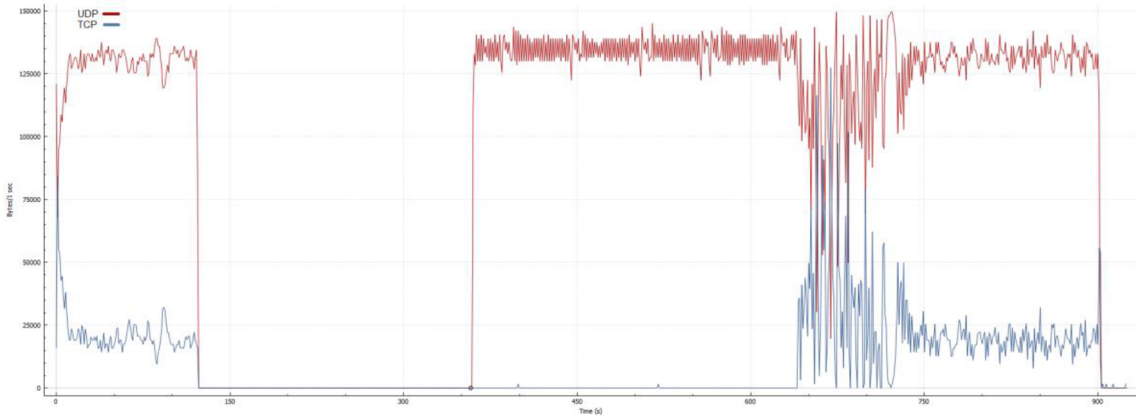
Obr. 8.6: Celkový příjem dat OSPF WFQ

Grafy znázorňující celkový provoz dat všech scénářů s použitím směrovacího protokolu OSPF zobrazují taktéž nejprve stabilní zátěžový datový provoz programu Iperf, dále jako ve scénářích s RIP protokolem došlo v čase 125 sekund k přerušení hlavní datové linky, což mělo též za následek přerušení datového toku. Ve scénářích s mechanikami FIFO a PQ došlo k přepnutí na záložní linku ve 160. sekundě, tedy za 35 sekund. K pozdějšímu přepnutí na redundantní linku došlo ve scénáři s WFQ frontovou mechanikou, a to v čase 163 sekund. Tuto prodlevu má pravděpodobně na vině spravedlivé, a tedy i stejné zacházení se všemi pakety mechanika WFQ.

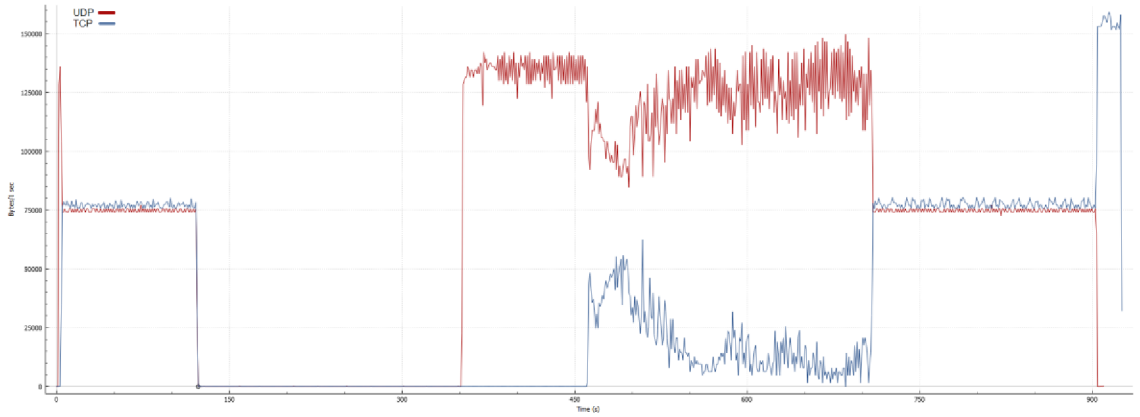
Po navázání náhradní línky ve scénářích s FIFO a PQ mechanikou lze pozorovat nejprve obnovený UDP provoz, poté navazuje vzrůst vytížení linky způsobený obnovením přenosu TCP dat. U scénáře s mechanikou WFQ se obnovil pouze datový přenos UDP a až na začátku největší nestability provozu v grafu došlo k obnovení i TCP přenosu. Dále následuje úsek, kdy data byla doručována po náhradní lince stejně jako ve scénářích s protokolem RIP. V 715 sekundě došlo k obnovení hlavní linky, a tedy i přesměrování datového toku, v prvních dvou scénářích lze zpozorovat krátké výpadky toku způsobené přesměrováním. Ve scénáři s PQ mechanikou je tento výpadek výraznější. Největší stabilitu datového toku před, během i po výpadku hlavní linky lze přisoudit scénářům s FIFO a PQ mechanikou, nejmenší pak scénáři s mechanikou WFQ.



Obr. 8.7: Příjem UDP/TCP dat RIP FIFO



Obr. 8.8: Příjem UDP/TCP dat RIP PQ



Obr. 8.9: Příjem UDP/TCP dat RIP WFQ

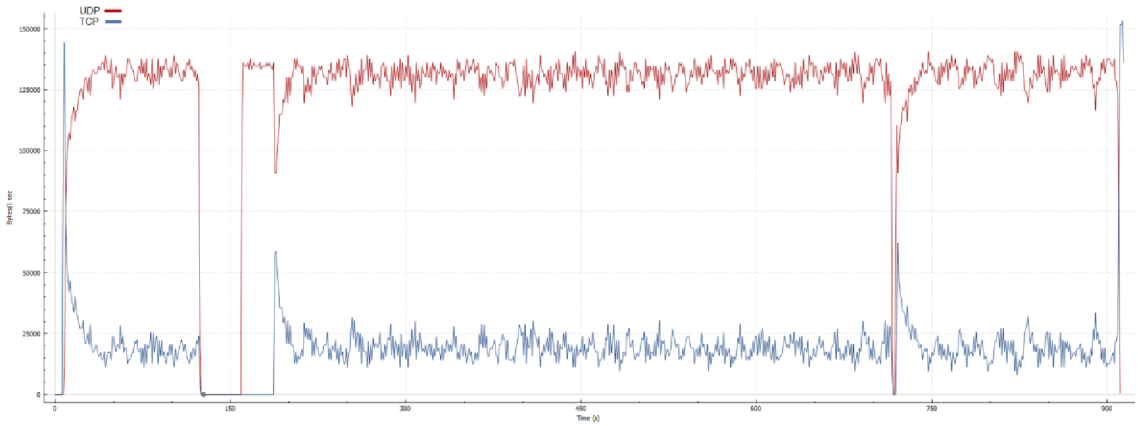
V této sérii grafů znázorňující provoz TCP a UDP všech RIP scénářů můžeme nejprve vidět graf ze scénáře s FIFO mechanikou. Tento graf zobrazuje zrcadlový provoz TCP a UDP s větším datovým vytížením ze strany UDP, díky zrcadlení obou průběhů lze předpokládat maximální využití přenosu obou datových linek. Po výpadku hlavní linky a přepnutí na redundantní linku je patrný přenos UDP a opožděný TCP přenos, který se spustil v čase 485 sekund, tedy o 120 sekund později

než UDP. Důvodem zpoždění TCP je patrně jeho způsob navazování spojení, které je daleko složitější a náročnější na síťové prostředky než přenos UDP. S obnovením TCP přenosu začalo opět zrcadlení obou linek stejně jako na začátku grafu, s rozdílem větší nestability u obou druhů přenosu až do chvíle znovuobnovení hlavní linky. Tato nestabilita byla pravděpodobně způsobena větším vytížením síťových prostředků vlivem ztrát a opětovným odesíláním TCP paketů, což vedlo k většímu vytížení mechaniky FIFO, a tedy i k většímu zahazování paketů. Přepnutí na hlavní linku bylo téměř okamžité ale přesto je patrná krátkodobá ztráta TCP přenosu v 725 sekundě, této ztráty využil podle grafu přenos UDP. Graf končí obvyklým, byť trochu méně stabilním zrcadleným přenosem TCP a UDP podobně jak tomu bylo na začátku grafu, úplným koncem grafu je pak doposílání ztracených TCP paketů.

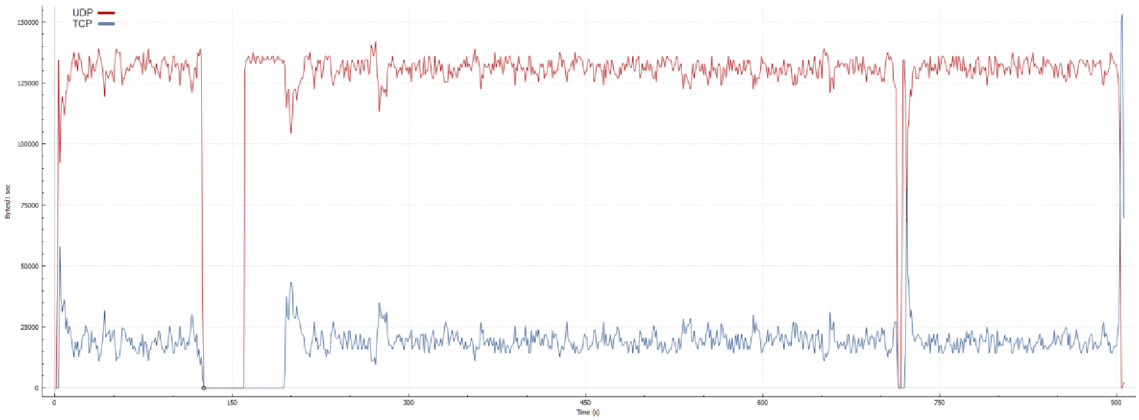
Druhý graf scénáře s mechanikou PQ vykazuje zezačátku téměř totožný průběh přenosu TCP i UDP dat, dále pak pokračuje přenosem UDP méně stabilním oproti začátku grafu ale více stabilním v porovnání s UDP přenosu v grafu prvním. To je způsobeno nejspíše mechanikou PQ a již zmíněným složitějším fungováním přenosu TCP. Přenosy TCP s UDP byly v mechanice PQ nastaveny na stejnou prioritu, ale vzhledem k jednoduchosti UDP protokolu došlo k jeho upřednostnění. V čase 640 sekund došlo k navázání provozu TCP, což vyústilo k velké nestabilitě obou druhů přenosů, na vině je pravděpodobně opět protokol TCP v kombinaci PQ mechaniky. S navázáním hlavní linky taktéž v 725 sekundě se provoz TCP i UDP více stabilizoval podobně jako u prvního grafu.

Ve třetím grafu můžeme nejprve pozorovat chvilkový UDP extrém způsobený pozdějším spuštěním testu TCP, dále již je vidět správné fungování mechaniky WFQ, která spravedlivě rozdělila síťové prostředky mezi oba druhy provozu dat. Trochu větší přednost WFQ mechanika věnovala přenosu TCP, což byla v pozdější části průběhu grafu výhoda v rychlejším navázání protokolu TCP. Po navázání redundantní linky graf vykazuje jistou podobnost s předešlými grafy, konkrétně v horší stabilitě přenosu a opožděném obnovení přenosu TCP dat. Ovšem v porovnání s předchozími grafy se zde TCP přenos obnovil nejrychleji a to v 460

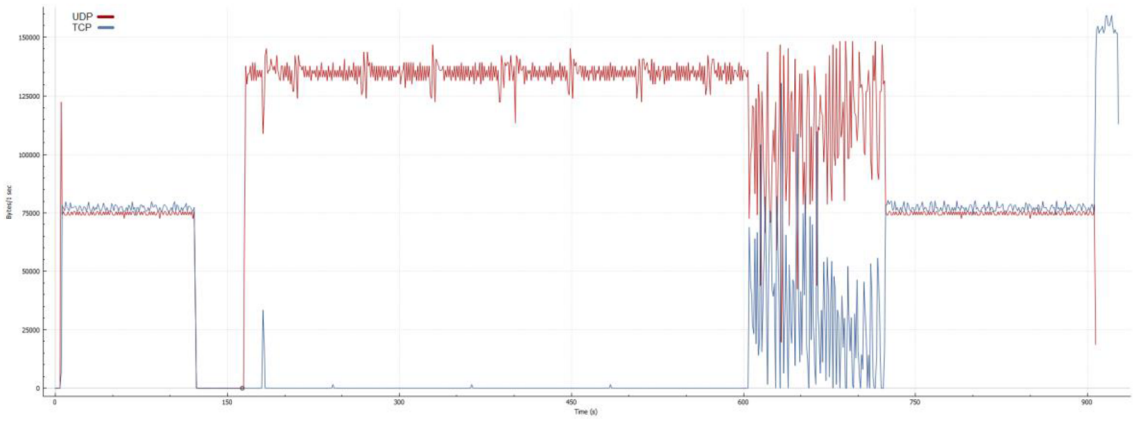
sekundě. Po obnově TCP přenosu se oba průběhy, jak TCP tak i UDP, projevily podobně jako v grafu scénáře FIFO. Z toho lze usoudit, že mechanika WFQ se v tomto bodě začala chovat podobně jako mechanika FIFO, tento fenomén mohl být způsobený nejspíše opět chováním TCP protokolu, bylo by však vhodné jej více prozkoumat. Průběh grafu dále pokračuje stabilním tokem TCP i UDP dat podobně jako na začátku grafu, tedy správným fungováním WFQ mechaniky, a to od chvíle opětovného zprovoznění hlavní linky. Z grafu je patrné velmi rychlé přepnutí linky, průběh grafu končí podobně jako první graf posíláním ztracených TCP paketů.



Obr. 8.10: Příjem UDP/TCP dat OSPF FIFO



Obr. 8.11: Příjem UDP/TCP dat OSPF PQ



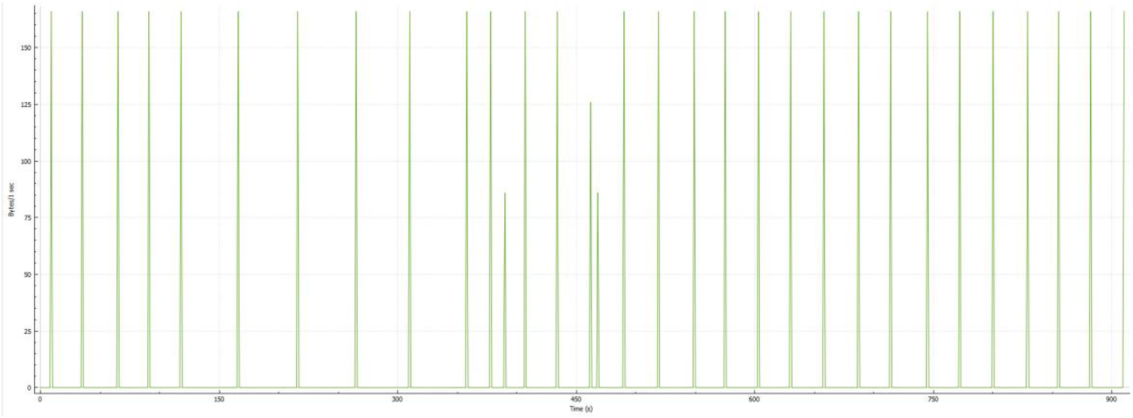
Obr. 8.12: Příjem UDP/TCP dat OSPF WFQ

První dva ze tří grafů zobrazující TCP a UDP průběhy ze scénářů s OSPF protokolem jsou až na pár výjimek téměř identické, oba začínají podobným zrcadlovým průběhem TCP a UDP jako v prvních dvou scénářích s RIP protokolem. Rozdíl je patrný v rychlosti přepnutí na redundantní linku oproti předchozím scénářům s RIP protokolem, dalším rozdílem je kratší doba obnovení TCP spojení.

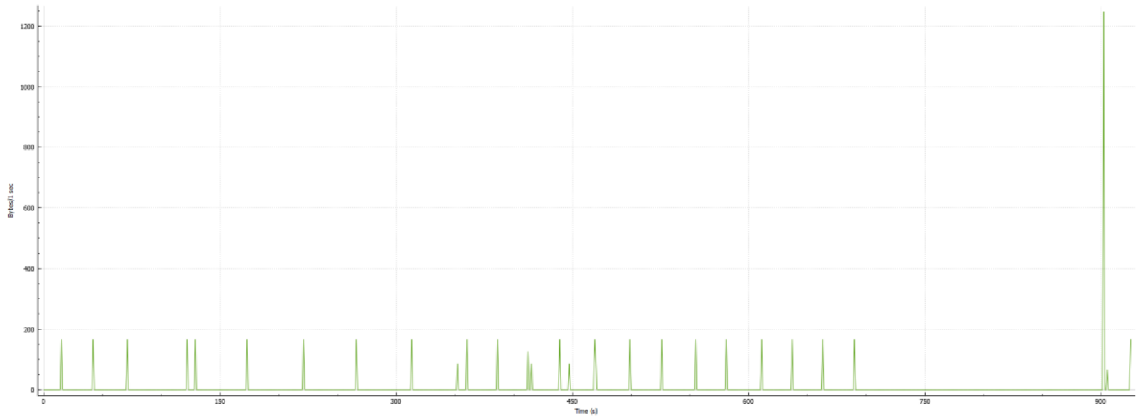
V prvním grafu s použitou FIFO mechanikou se TCP provoz obnovil nejrychleji a to v 185. sekundě, ve druhém grafu s PQ mechanikou pak v 195. sekundě. Oba grafy poté zobrazují zrcadlové provozy po náhradní lince podobně jako u scénářů s RIP protokolem, tyto průběhy se však jeví jako stabilnější. Při opětovném přepnutí na hlavní linku, tedy v 725. sekundě, došlo k poklesu přenosu dat TCP i UDP, ve druhém grafu však protokol UDP využil pomalejšího obnovení TCP podobně jako ve scénáři s RIP protokolem a mechanikou PQ. Po obnovení provozu na hlavní linku následuje v obou scénářích normální přenos TCP a UDP dat stejně jako při užití náhradní linky, z toho je patrné, že v těchto případech vedlejší linka neměla vliv na přenos dat. K větší stabilitě přenosu dat nejspíše přispívá i sám OSPF protokol, a to z důvodu menší náročnosti na síťové prostředky a rychlejší odezvy než RIP protokol.

Poslední graf se scénářem WFQ mechaniky a protokolem OSPF vykazuje podobné průběhy jako scénář s RIP protokolem a stejnou mechanikou zacházení paketů. Na začátku grafu je taktéž UDP extrém následovaný normálním průběhem fungování mechaniky WFQ, poté výpadkem spojení a přepojením na náhradní linku.

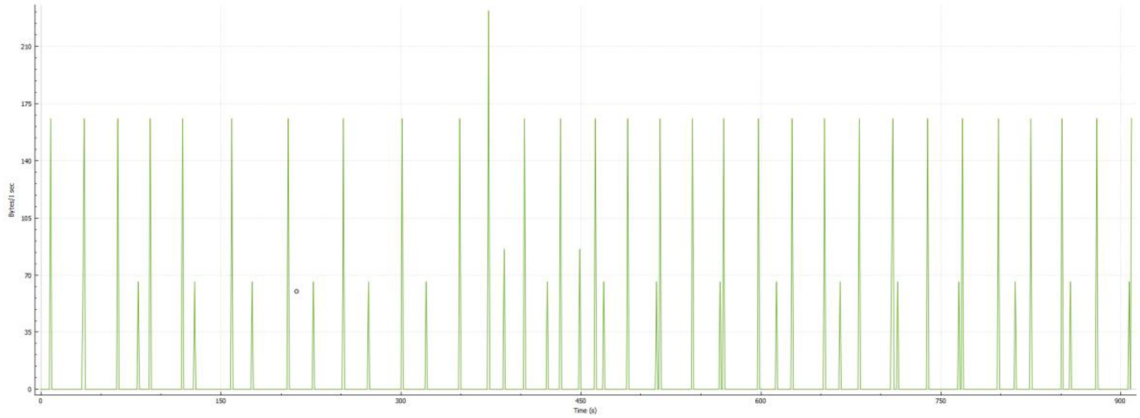
Navázání na náhradní linku bylo provedeno rychle a 3sekundový rozdíl oproti scénářům s PQ a FIFO mechanikou je prakticky zanedbatelný. Po sepnutí náhradní linky však došlo k obnovení pouze UDP spojení, které navíc bylo méně stabilní. Na začátku této události je vidět chvilkový nárůst TCP a v průběhu grafu i další malé nárůsty, jedná se nejspíše o neúspěšné pokusy o navázání TCP spojení, samotné obnovení spojení TCP přichází až o 442 sekund později v čase 605. Od tohoto okamžiku se destabilizovaly oba přenosy pravděpodobně kvůli velkému počtu znovu odesílaných TCP paketů. Graf dále pokračuje obnovením původní linky v čase 725 sekund a následuje opět stabilní provoz jako na začátku relace, poté je graf zakončen taktéž posíláním zahozených TCP paketů podobně jako v předešlých grafech. Z grafu vyplývá správné fungování mechaniky WFQ na hlavní lince stejně jako ve scénáři s RIP protokolem a stejnou mechanikou třízení, ovšem při výpadku linky a přesměrování datového toku na redundantní linku přestane mechanika WFQ správně fungovat. Důvodem by mohla být špatná konfigurace WFQ na směrovačích tvořících náhradní línku, to ale bylo vyvráceno náhledem do výpisů jejich konfigurací. Z tohoto důvodu by bylo vhodné mechaniku WFQ dále prostudovat.



Obr. 8.13: Příjem směrovacích dat RIP FIFO



Obr. 8.14: Příjem směrovacích dat RIP PQ



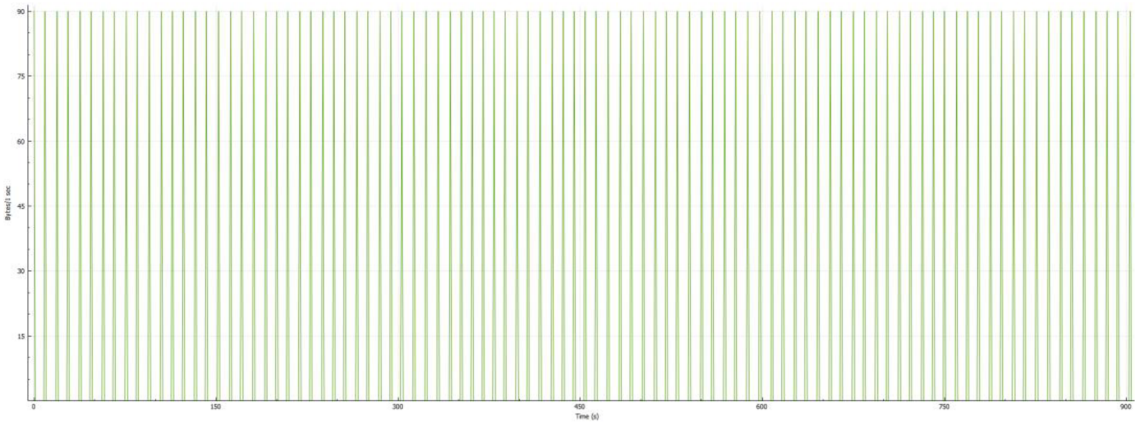
Obr. 8.15: Příjem směrovacích dat RIP WFQ

Dále byly vygenerovány grafy zobrazující příjmy směrovacích RIP paketů ze třetího směrovače. První graf ze scénáře s FIFO mechanikou začíná obvyklým a správným fungováním protokolu RIP, tedy příjmem směrovacích tabulek zhruba každých 30 sekund, drobné časové rozdíly v příjmu paketů byly pravděpodobně způsobeny jejich zpožděním ve frontě FIFO. Po výpadku hlavní datové linky lze pozorovat časovou prodlevy mezi RIP pakety, důvodem prodlev byl nejspíše vysoký datový provoz a vytížení síťových prostředků v kombinaci s FIFO mechanikou. Graf pokračuje po obnově datového provozu skrze náhradní linku a v úseku, kdy fungovalo pouze UDP spojení, lze vidět chvilkový menší příjem paketů nejspíše kvůli pokusu o obnovu TCP přenosu. viz obrázek 8.7. Dále je v grafu patrná série dalších dvou menších poklesů RIP dat, ty byly způsobeny taktéž nejspíše protokolem TCP, při již úspěšném pokusu o obnovení datového toku. Zbytek grafu pokračuje až do konce správně načasovaným příjmem paketů RIP protokolu, lze tak usoudit, že po úspěšném obnovení TCP a UDP přenosu nic negativně neovlivňovalo správné fungování protokolu RIP.

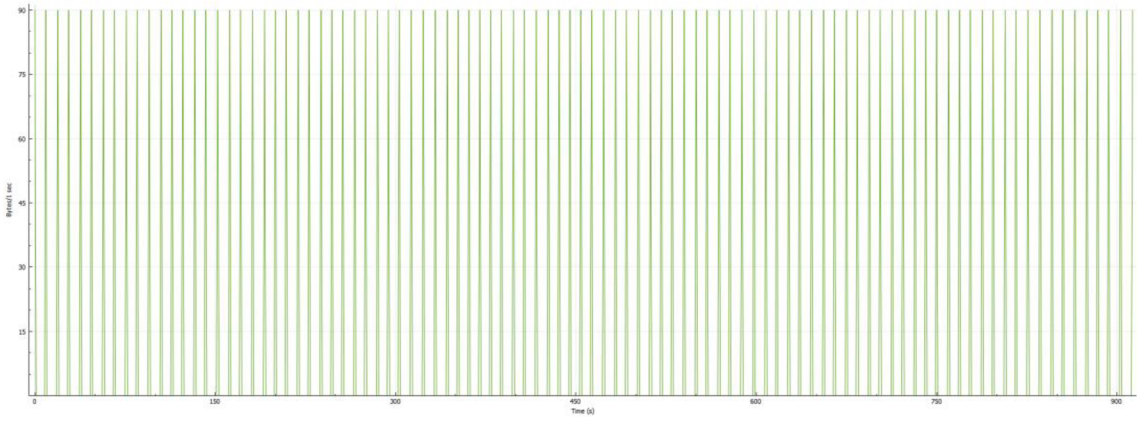
Další graf zabývající se scénářem s mechanikou PQ sice díky jinému měřítku vypadá velmi rozdílně oproti prvnímu grafu, ale při bližším pohledu lze najít pouze malé rozdílnosti kromě začáteční a konečné části grafu. Zezačátku lze vyčíst podobné správné fungování protokolu RIP jako u předešlého grafu s rozdílem asi 50sekundového úseku bez dat. Ten byl nejspíš způsoben opožděním paketů vlivem mechaniky PQ, která měla nastavenou vyšší prioritu pro zátěžové datové toky TCP a UDP než data RIP protokolu. Tento výrok podporují dva po sobě jdoucí vrcholky přijatých směrovacích dat za již zmíněným 50sekudnovým úsekem. Dále graf pokračuje podobně jako u prvního grafu výpadkem linky, větší prodlevou při přijetí směrovacích dat a poté opětovným správně načasovaným přijetím dat po přepnutí na linku vedlejší. V této části jsou patrné tři odchylky v načasování a množství přijatých dat, které byly taktéž pravděpodobně způsobeny protokolem TCP. Po přepnutí zpět na hlavní linku je ke konci grafu patrné, že přestala přicházet všechna směrovací data. Tento fakt podporuje výsledky z první laboratorní úlohy, kdy bylo zjištěno potenciální ohrožení funkce směrování kvůli mechanice PQ, jenž byla úmyslně nesprávně nastavena pro upřednostnění uživatelských dat před daty servisními. Graf končí extrémem o přijetí směrovacích dat po skončení zátěžového datového provozu, tato data ovšem přišla se značným zpožděním a nejsou tedy relevantní.

Průběh grafu s WFQ mechanikou nejprve jeví podobné normální fungování přijímání směrovacích dat jako dva předešlé grafy. Celkový průběh je více podobný průběhu v prvním grafu s mechanikou FIFO až na pravidelnou sérii vrcholků o přijetí menšího objemu dat, které jsou mezi sebou vzdáleny 50 sekund, na extrém nárazového přijetí dat po přepnutí na záložní linku a na dva vrcholy menší než obvyklá směrovací data ale větší, než je pravidelná série zmíněných menších přijatých dat. První anomálie, tedy série přijatých dat o menším objemu než je obvyklý v 50sekundovém rozmezí byla patrně způsobena frontovou mechanikou WFQ, jenž způsobila desynchronizaci toků směrovacích dat oproti normálu. Extrémní příjem směrovacích dat byl nejspíše způsoben zapojením náhradní datové linky a přišla nová směrovací data. Zbylé dva středně velké příjmy dat mohly být

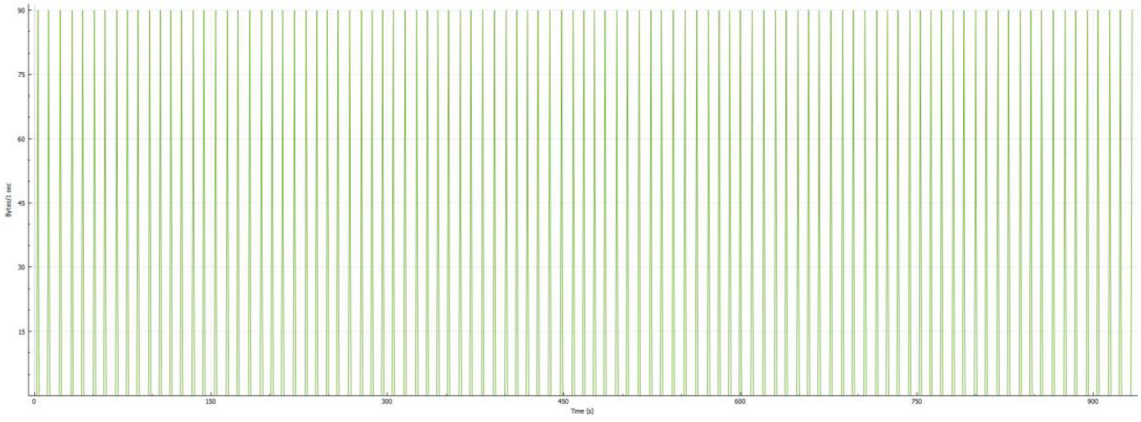
způsobeny protokolem TCP podobně jako u prvního grafu, to je však v rozporu s grafem Obr. 8.9, který v daný časový úsek nevykazuje TCP aktivitu.



Obr. 8.16: Příjem směrovacích dat OSPF FIFO



Obr. 8.17: Příjem směrovacích dat OSPF PQ



Obr. 8.18: Příjem směrovacích dat OSPF WFQ

Směrovací data přijatá na směrovači R3 ze všech OSPF scénářů vygenerovaných do grafů jeví prakticky stejný průběh, to je způsobeno menší náročností protokolu na síťové prostředky než při použití protokolu RIP. Menší objem a častější příjem

směrovacích dat měl za následek rychlejší přesměrování zátěžového toku dat TCP a UDP na náhradní linku, a jak již bylo vidět v grafech celkového příjmu dat na R3 směrovači, díky protokolu OSPF byl celkový příjem stabilnější než při použití RIP protokolu.

Tab. 8.1: Výsledky programu Iperf

		TCP		UDP			
		Transfer [MBytes]	Bandwidth [Kbits/sec]	Transfer [MBytes]	Bandwidth [Kbits/sec]	Jitter [ms]	Lost/Total [%]
OSPF	FIFO	15,2	141	105	973	2,285	7,1
	PQ	14,5	135	104	973	2,489	7,1
	WFQ	26,7	243	87,8	817	5,610	22
RIP	FIFO	12,0	34,5	79,2	736	1,907	30
	PQ	10,8	85,0	79,9	743	1,990	29
	WFQ	25,2	257	63,4	590	9,131	44

Tabulka 8.1 zobrazuje výsledky testů programu Iperf ze všech scénářů. Po nahlédnutí do tabulky je patrný největší přenos a největší šířka pásma TCP dat ve scénářích s použitou mechanikou WFQ ale na úkor UDP přenosu, Jitteru, šířky pásma a většího procenta ztrátovosti. Největší hodnotu Jitteru, s nejmenší šířkou UDP pásma a ztrátovost má pak RIP scénář s WFQ mechanikou. Z toho lze vyvodit správné fungování mechaniky v obou scénářích, ale kvůli delší časové prodlevě mezi přepnutím z hlavní datové linky na vedlejší linku má scénář RIP s WFQ nejhorší výsledky UDP testu. Nejlepší výsledky UDP testu má OSPF scénář s frontou FIFO, kromě hodnoty Jitteru. Tuto hodnotu mají nejlepší scénáře s protokolem RIP kromě již zmíněného scénáře s WFQ mechanikou.
Celkově ve všech scénářích s OSPF protokolem bylo přeneseno více dat než ve scénářích s RIP protokolem, u kterých byla zaznamenána menší šířka pásma v testech TCP i UDP.

9. POROVNÁNÍ LABORATORNÍCH ÚLOH

První laboratorní úloha poskytuje studentům podrobnější teoretický základ, zatímco její praktická stránka je jednodušší, takže studenti při prvním kontaktu s touto problematikou nejsou zahlceni informacemi. Umožňuje snadnější získávání informací ze simulace a také informací, které by v reálném světě nebylo možné získat jako například průběh zahozených IP paketů. Výstup z praktické části první laboratorní úlohy více odpovídá teoretickému základu, pravděpodobně kvůli jednodušší simulaci, zároveň ale neodráží reálnou komplexnost situace na rozdíl od druhé laboratorní úlohy.

Navazující laboratorní úloha, která reálněji simuluje fyzická zařízení v laboratoři, věrněji vypovídá o dané problematice a klade větší nároky na studenty při interpretaci naměřených výsledků. Výsledky této laboratorní úlohy vesměs potvrzují teoretické poznatky z první laboratorní úlohy až na určité výjimky. Například zde nebylo prokázáno ohrožení směrovacího protokolu OSPF a mechanismus WFQ vyjevil nové chování, které by bylo vhodné dále prozkoumat. V návaznosti na tuto laboratorní úlohu se nabízí prostor pro otestování dalších komplexnějších směrovacích protokolů a složitějších QoS mechanik.

10. ZÁVĚR

Tato bakalářská práce se zabývá problematikou kombinace směrování se zajištěním kvality služeb. Čtenáře nejprve seznamuje s teorií v prvních čtyřech kapitolách, které pojednávají o směrování, protokoly směrování a technologii QoS.

V praktické části diplomové práce jsou představeny dvě laboratorní úlohy, které byly vytvořeny za účelem seznámit studenty s výhodami ale i s možnými problémy kombinace technologii směrování s QoS. První laboratorní úloha přináší obecnější poznatky a klade na studenty menší nároky, takže je lepší volbou pro demonstraci správného fungování obou technologií a více vyzdvihuje možné problémy spjaté s jejich kombinací.

Druhá laboratorní úloha je zaměřena na reálnější chování jak zařízení v síti, tak i zmíněných technologií. Na studenty klade větší nároky ohledně teoretických znalostí sítí nezbytných ke zpracování výsledků a jejich pochopení. Tyto výsledky ovšem věrohodněji vykreslují možnou situaci v reálném prostředí a poukazují na fakt, že skutečnost se může lišit od teorie.

Celkově takto práce představuje nový pohled na zmíněné technologie a jejich vzájemnou interakci. Jako logické pokračování zkoumání těchto technologií se zde nabízí možnost otestovat další funkce QoS a jiné typy směrovacích protokolů za účelem ověřit, jestli se za stejných podmínek chovají obdobným způsobem, nebo jsou efektivnější.

LITERATURA

[1] Jeřábek, J. (2013). *Komunikační technologie* [Online] (1st ed.). Brno: Vysoké učení technické v Brně.

[2] Bouška, P. (2009). Cisco Routing 1 - obecné vlastnosti směrovacích protokolů [Online]. Retrieved December 19, 2019, from https://www.samuraj-cz.com/clanek/cisco-routing-1-obecne-vlastnosti-smerovacich-protokolu/

[3] Stodůlka, T. (2017). *Směrování v datových sítích* (Bakalářská diplomová práce). Brno.

[4] Bloudíček, J. (2014). *Modelování směrovacích protokolů EIGRP* (Magisterská diplomová práce). Brno.

[5] Daněk, M. (2015). Směrovací mechanizmy mezi autonomními systémy internetu (Bakalářská diplomová práce). Brno.

[6] Celárek, O. (2014). *Sledování parametrů směrování v páteřních sítích* (Bakalářská diplomová práce). Brno.

[7] Oujezský, V. (2009). Technologie pro zajištění kvality služeb v IP sítích a jejich vzájemná spolupráce (Bakalářská diplomová práce). Brno.

[8] Ludvíček, P. (2009). Vliv nastavení parametrů řízení provozu na efektivnost technologie DiffServ (Bakalářská diplomová práce). Brno.

[9] Kiška, M. (2012). Návrh spolehlivé podnikové sítě s podporou kvalitativních požadavků služeb (Bakalářská diplomová práce). Brno.

[10] Wireshark. (2001). In *Wikipedia: the free encyclopedia*. Wikimedia Foundation. https://en.wikipedia.org/wiki/Wireshark

[11] Dzerkals, U., & Lim, C.Doe, M. (Ed.). *EVE-NG Professional Cookbook*. In (pp. 1-197). https://www.eve-ng.net/images/EVE-COOK-BOOK-1.0.pdf

[12] Iperf. (2001). In *Wikipedia: the free encyclopedia*. Wikimedia Foundation. https://en.wikipedia.org/wiki/Iperf

SEZNAM SYMBOLŮ A ZKRATEK

RIP	Routing Internet Protocol
OSPF	Open Shortest Path First
EIGRP	Enhanced Interior Gateway Routing Protocol
IS-IS	Intermediate Systém to Intermediate System
BGP	Border Gateway Protocol
FIFO	First In First Out
PQ	Priority Queuing
WFQ	Weighted Fair Queueing
WRR	Weighted Round Robin
QoS	Quality of Services
IntServ	Integreated Services
DiffServ	Differentiated Services
DSCP	Differentiated Services Codepoint
PHB	Per Hop Behavior
IP	Internet Protocol
RSVP	Resource Reservation Protocol
COPS	Common Open Policy Service
VoIP	Voice over Internet Protocol
TCP	Transport Control Protocol
UDP	User Datagram Protocol
PC	Personal Computer
HTTP	Hypertext Transfer Protocol
FTP	File Transport Protocol

SEZNAM PŘÍLOH

První lab úloha: Manuál pro vyučujícího/studenty 69
První lab úloha: Vzorový protokol 80
Druhá lab úloha: Manuál pro vyučujícího/studenty 83
Druhá lab úloha: Vzorový protokol 95
Druhá lab úloha: Výpisy směrovačů.zip ..... Odevzdáno elektronicky

Testování privátní sítě – Testování QoS v kombinaci se směrovacími protokoly

Úvod k úloze

Cílem laboratorní úlohy je srovnání QoS frontových FIFO, PQ a WFQ mechanik v kombinaci směrovacích protokolů RIP a OSPF na jednoduché síti se třemi směrovači a skupinou koncových zařízení. V síti simulujte provoz služeb operujících v reálním čase se službou FTP a HTTP, současně simulujte výpadek jedné linky mezi směrovači a pozorujte chování sítě. Zjistěte rychlost reakce směrovacích protokolů při výpadku linky a množství zahozených paketů po obnovení spojení při použití různých frontových mechanik.

Výstupem této laboratorní úlohy by měly být grafy zobrazující zahazování paketů v síti ze scénářů pro frontové mechanismy FIFO, PQ a WFQ s použitím nejprve směrovacího protokolu RIP a poté OSPF.

Cíle laboratorní úlohy

1. Sestavte síť podle obrázku a návodu. – *tuto část předpřipravuje vyučující*
2. Nastavte směrovací protokol RIP
3. Nastavte výpadek linky
4. Nastavte provoz v síti - *tuto část předpřipravuje vyučující*
5. Nastavte frontové mechanismy FIFO, PQ, WFQ a postupně je proměřte.
6. Síť přenastavte na směrovací protokol OSPF a proměřte frontové mechanismy stejně jako v předchozím bodě.
7. Porovnejte výsledky všech frontových mechanismů v kombinaci se směrovacími protokoly.

Teoretický úvod

Směrování je technika, která propojuje jednotlivé sítě po síťové vrstvě. Tato technika v podstatě nachází nejlepší cesty skrze mezilehlé uzly k propojení uživatelů. K tomu jsou využívány různé protokoly, které mají kromě úkolu najít nejvhodnější cesty taky v případě přerušení již používané cesty, například výpadkem mezilehlého uzlu nebo přerušení linky, najít vhodné náhradní cesty.

Routing Information Protocol

Dynamický směrovací protokol a spadá pod třídu Distance Vector protokolu, který pracuje s metrikou počítání skoků k cíli a pravidelným rozesíláním směrovacích tabulek sousedním směrovačům. Maximální počet skoků je 15 a nejlepší cesta je označena jako ta s nejmenším počtem skoků. Pokud je cílová síť vzdálená 16 a více skoků skrze směrovače, je označena za nedosažitelnou. Tento protokol určuje pouze jednu cestu k cíli, a tedy jasnou nevýhodou je nemožnost rozložení zátěže na více cest. RIP protokol existuje ve verzích RIPv1, RIPv2 a RIPng/RIPv6.

RIP protokol využívá pro své fungování 4 časovače:

- **Update Timer** – je časovač, po jehož vypršení směrovač posílá všem svým sousedním směrovačům svou celou směrovací tabulku. (defaultně 30 vteřin)

- **Invalid Timer** – časovač odpočítává dobu platnosti cesty (řádku) ve směrovací tabulce. Pokud před vypršením časovače obdrží směrovač aktualizaci cesty tak je časovač resetován na původní hodnotu, v případě vypršení časovače je cestě přiřazena metrika 16 a přesouvá se do stavu hold-down. (defaultně 180 vteřin)
- **Hold-down Timer** – odpočítává dobu, kdy jsou ještě informace o cestě přidrženy, ale již ji nejde obnovit ani v případě obdržení nových aktualizací o dosažitelnosti cíle. Tento časovač slouží k poskytnutí dostatku času k přizpůsobení změnám v síti. (defaultně 180 vteřin)
- **Flush Timer** – slouží k vymazání záznamů o cestě a běží současně s Invalid Timer časovačem. Po 60 vteřinách od označení cesty za neplatnou, dojde k jejímu vymazání. (defaultně 240 vteřin)

Rozdíl mezi verzemi RIPv1 a RIPv2 protokolu je ten, že druhá verze podporuje třídní adresování, tedy proměnné délky masek podsítí, což má za následek efektivnější sumarizaci a méně adres ve směrovacích tabulkách v jednotlivých směrovačích. RIPv2 také podporuje autentizaci za pomoci MD5 šifrování.

Open Shortest Path First

Protokol typu Link State a směrovač s využitím tohoto protokolu získává informace o celé topologii sítě za pomoci skupiny paketů. Nejvhodnější cesta se určuje na základě topologie a metriky linek mezi jednotlivými uzly, která se skládá například ze zpoždění, zabezpečení a šířky pásma. Výsledná hodnota se pak označuje jako „cena spoje“ a nejvýhodnější cesta je ta s nejmenší hodnotou. Tento protokol využívá Dijkstrova algoritmu k odvození nejvhodnější cesty a podporuje dělení sítí na subsítě. Rozdělením na subsítě (oblasti) lze snížit počet zpráv, které si směrovače posílají a zmenšit tak celkové zatížení sítě. Směrovače znají topologii jen o dané oblasti, ve které se nacházejí a informace o okolních sítí nebo subsítí jim poskytují hraniční směrovače, které propojují oblasti mezi sebou. Hraniční směrovače shrnují informace o sítích, v nichž se nachází a posílají je do sousedních sítí. Výhodou tohoto protokolu oproti RIPv2 protokolu je hlavně, že při změně v síti směrovač neposílá celou směrovací tabulku, což značně snižuje zatížení v síti a jeho reakce na změny v topologii je rychlejší. Směrovač si kontroluje dostupnost sousedních směrovačů a pokud dojde ke zemně v sítí tak ihned zasílá informaci jen o dané změně hierarchicky všem směrovačům v síti. Směrovače si přeposílají aktualizace vždy při změně anebo každých 30 minut v případě, že nenastane žádná změna. U tohoto protokolu není omezen počet skoků a cenu spoje lze nastavit manuálně, a tak upřednostnit pomalejší cestu v případě nízkých požadavků na přenos. Protokol podporuje proměnné délky masky podsítí a druhá verze podporuje autentizaci za pomoci šifrování MD5.

OSPF pro své fungování využívá 5 druhů paketů:

- **Hello** – slouží k navázání a udržení spojení se sousedním směrovačem a využívá dvou časovačů. Hello interval udržuje spojení a je posílán každých 10 vteřin. Dead interval je zpravidla čtyřnásobek Hello intervalu, tedy 40 vteřin a pokud směrovač nepřijme žádný Hello paket od souseda do vypršení tohoto intervalu spojení mezi směrovači zanikne.
- **Database Description** – slouží k přeposílání informací o topologii sítě.
- **Link State Request** – žádá o zaslání chybějící položky po obdržení paketu DBD.
- **Link State Update** – je odpověď na LSR paket s chybějící položkou.
- **Link State Acknowledgment** – slouží jako potvrzení o přijetí paketu LSU.

- Při navazování nového spojení a výměnu směrových informací mezi směrovači procházejí směrovače 7 stavy:
- **Down** – je stav před obdržení Hello paketu, kdy ještě směrovač neví o existenci souseda.
- **Init** – stav je po obdržení Hello paketu, stáje ještě bez zařízení obousměrné komunikace.
- **Two-Way** – označuje stav, kdy je zřízena obousměrná komunikace a určuje se pověřený směrovač a záložní pověřený směrovač. Všechny směrovače pak při změně v topologii komunikují přes pověřený směrovač.
- **Exstart** – ustanovuje směrovače Master a Slave podle velikosti sekvenčního čísla v paketu DBD. Komunikaci začíná směrovač Master.
- **Exchange** – výměna DBD paketů mezi směrovači.
- **Loading** – výměna paketů LSR, LSU a LSA.
- **Full** – je stav plné synchronizace mezi směrovači.

QoS Quality of Services

Technologie OoS je skupina podpůrných funkcí, které mají zajistit upřednostnění přenosu dat pro náročnější služby před nenáročnými službami. Obecně se technologií QoS podporují služby v reálném čase, jako jsou například hlasové služby, live streaming, videokonference, nicméně je nutné zajistit i funkčnost nenáročných služeb (spolehlivý přenos dat, emailové služby, chat, a td). QoS tedy musí efektivně rozdělovat síťové prostředky pro služby realtime i obecné služby, k tomu je nutné provádění klasifikaci a značení datového provozu. Tato klasifikace probíhá na hranici domény, tedy na hraničním přepojovacím prvku, popřípadě s příslušnou dohodou lze přebírat způsob klasifikaci datových toků od poskytovatele. Upřednostnění různých služeb lze použít i pro přepojovací prvky uvnitř sítě. Vložením informací do hlaviček datových jednotek o klasifikaci se umožní prioritní obsluhování služeb uvnitř sítě, za použití jednoho z frontových mechanismů. V těchto mechanismech se třídí datové jednotky podle priority do front, ze kterých se jim přidělují dostupné a požadované síťové prostředky.

Mezi důležité parametry pro službu QoS jsou:

- **Jitter** – proměnlivost zpoždění je negativní parametr zejména pro služby pracující v reálném času. Zpoždění se dynamicky mění podle aktuálního zatížení a stavu sítě. Pro eliminaci tohoto parametru se na příjímací straně využívá Jitter Bufferu, který slouží jako zpožďovací paměť a mění tak zpoždění z dynamického na konstantní.
- **Zpoždění** – je parametr, který sčítá celkově všechna zpoždění počínaje od kódování zdroje, paketizační zpoždění, propagační zpoždění po Jitter Buffer zpoždění. Propagačním zpožděním se myslí potřebný čas k přenosu dat od zdroje k cíli po síti. Paketizační zpoždění je čas strávený umísťování bitů do paketů, například do hlaviček. Kódováním zdroje se myslí A/D a D/A převod hlasu u zdroje a cíle.
- **Ztrátovost paketů** – určuje množství paketů které buď dorazili pozdě a u služeb pracujících v reálném čase se nedají už použít, nebo které vůbec nedorazili. Jde taktéž o negativní parametr, který se však u služeb v reálném čase dá do jisté míry tolerovat. Ztrátovost paketů může například zhoršit kvalitu hlasu či videa anebo u TCP služeb může zapříčinit nutnost znovu odeslání dat.

Pro provoz sítě podporující prioritizaci služeb existují dva modely řízení sítě. Prvním modelem je Integrated Services (**IntServ**), model rezervuje zdroje v síťových prvcích po celé cestě mezi zdrojem a příjemcem dat. Rezervace zdrojů v síti obsahuje nalezení vhodné cesty skrze uzly v síti a alokaci zdrojů pro každý směr přenosu od příjemce. U každého směrovače v cestě se zjišťuje požadavek oprávnění a dostatek zdrojů pro uskutečnění relace. K tomu se používají různé protokoly jako například RSVP (Resource Reservation Protocol) a COPS, které využívá firma CISCO. Druhým modelem je Differentiated Services (**DiffServ**). Síť s využitím tohoto modelu třídí datové toky jen na hraničních uzlech. V těchto uzlech síťové prvky třídí datové toky stejných druhů do jednotlivých tříd, a to úpravou DSCP pole. Nerozlišují tak například jednotlivé hovory VoIP, ale upřednostní všechny VoIP hovory jako skupinu. Prvky uvnitř sítě se řídí podle PHB (Per Hop Behavior). Služba DiFfServ je implementována například ve standardech RFC 247, RFC 2475 a RFC 2598.

Třízení datových toků

Třízení datových toků probíhá na hraničních uzlech DiffServ domény. Dochází zde nejen ke klasifikaci příchozích paketů ale i k označování kódem DCSP, díky němuž se pakety dále dělí do skupin PHB. Ke značkování IP paketů se používají pole v jejich záhlaví. U IPv4 protokolu jsou to Type of Service pole a u IPv6 zase pole Traffic Classes. V těchto 8bitových polích se využívá 6 bitů pro DSCP. Obecně paket s vyšší binární hodnotou DSCP má i vyšší prioritu.

Per Hop Behavior (PHB) slouží k identifikaci chování paketu v síti a dělí na dvě skupiny. První skupinou je Expedited Forwarding PHB neboli urychlené doručování a druhou skupinou je Assured Forwarding PHB, tedy zaručené doručování.

EF PHB zaručuje jistou šířku pásma a je vhodná služby v reálném čase jako jsou například video či hlas. V této skupině mají pakety nízké zpoždění, malý jitter a malou ztrátovost. Hodnota DSCP pro tuto skupinu je 101110.

Skupina AF PHB je spíše určena pro spolehlivý přenos dat, a tedy pro TCP, jelikož jitter a zpoždění nejsou v této skupině důležité oproti ztrátovosti paketů. V této skupině jsou čtyři třídy AF1 až AF4, které jsou rozděleny ještě na další tři podtřídy. Tyto podtřídy slouží pro priority zahození a rozlišují se na nízkou, střední a vysokou prioritu. Třídy AF mají různé přidělení šířky pásma, velikosti bufferu a dalších síťových prostředků. V těchto třídách můžou být použity metody prevence proti zahlcení. Jedná se o metodu Random Early Detection a Weighted Random Early Detection. Při použití jedné z metod je vyšší pravděpodobnost zahození u paketů s vyšší prioritou zahození v případě hrozícího zahlcení.

Třída PHB	Podtřída PHB	Priorita zahození	DSCP
EF			101110
AF4	AF41	Malá	100010
	AF42	Střední	100100
	AF43	Velká	100110
AF3	AF31	Malá	011010
	AF32	Střední	011100
	AF33	Velká	011110
AF2	AF21	Malá	010010
	AF22	Střední	010100
	AF23	Velká	010110
AF1	AF11	Malá	001010
	AF12	Střední	001100
	AF13	Velká	001110
Best Effort			000000

Tab. č. 1 Třídy PHB

Systém proti zahlcení front

Random Early Detection a Weighted Random Early Detection metody slouží k prevenci proti zahlcení front. Pokud se problém zahlcení front začne řešit až při jeho uskutečnění, tedy zaplní se fronty, nově příchozí pakety budou zahazovány, a to bez ohledu na jejich prioritu do doby uvolnění front. Tento problém může tvořit ještě další problém a tím je tvoření zahlcovacích vln na směrovačích střídající se s jevem sníženým tokem dat přes síť. K tvorbě tohoto problému přispívá TCP protokol, dojedli k zahození paketů využívající TCP protokol tak jsou tyto pakety znovu odeslány sníženou rychlostí odesílání. Pokud dojde k hromadnému zahození takových paketů od více uživatelů tak dojde k celkovému snížení rychlosti odesílaných dat a nevyužití kapacity linky. Po obnovení rychlostí u uživatelů může dojít k opětovnému zahlcení front ve směrovačích a celý problém se opakuje.

Takové problémy eliminují metody RED a WRED. RED metoda zahazuje pakety z TCP spojení s určitou pravděpodobností od určitého procenta zaplnění fronty. Pravděpodobnost zahození roste se zvyšujícím se zaplnění fronty. Při zahazování paketů dojde ke snížení datových toků u některých uživatelů, ale zabrání se tím zaplnění front a vyrovnání příchozích a odchozích datových objemů. Metoda WRED je modifikovaná metoda, která je schopna rozlišit klasifikaci dat, a tak může méně prioritním datům přiřadit větší pravděpodobnost zahození a datům více prioritním naopak pravděpodobnost zahození snížit.

Fronty v technologii DiffServ

- **FIFO** Jedná se o nejjednodušší mechanismus fronty bez jakéhokoliv algoritmu řízení. Pakety jsou obsluhovány tak jak přijdou, podle čehož nese tento mechanismus i své jméno „First In First Out“. Výhoda tohoto mechanismu je jeho jednoduchost, je zapotřebí jen jedna vyrovnávací paměť, která odesílá data ve stejném pořadí, ve kterém přišly. Hlavní nevýhodou je taktéž jednoduchost, kvůli absenci algoritmu řízení tento mechanismus nerozlišuje druhy dat, třídy provozu, a v případě zaplnění

front je se všemi druhy dat zacházeno stejně. Tento mechanismus je vhodný pro službu Besteffort, pro provoz bez podpory QoS.

- **PQ** Mechanismus front Priority Queunig řeší neduhy fronty FIFO, ale zároveň si zachovává jednoduchost, ovšem i tento mechanismus má své nevýhody. PQ mechanismus třídí klasifikované pakety do front podle priorit. Fronty s pakety s vyšší prioritou jsou obsluhovány do doby vyprázdnění fronty, pak mechanismus začne obsluhovat frontu s nižší prioritou. Zde může vzniknout problém „uvíznutí" paketů ve frontách s nižší prioritou, pokud fronta s vyšší prioritou je stále doplňována dalšími pakety. Tento mechanismus je vhodný pro služby založené na UDP protokolu, jako například stream video či videohovor. Mechanismus PQ lze rozdělit například do dvou front, kde první fronta bude pro služby pracující v reálném čase a druhá fronta bude obsluhovat služby založené na TCP protokolu a budou tedy obsluhovány službou Best Effort. Při této implementaci je však nutné počítat s rizikem, že se pakety ve druhé frontě zpozdí a vyšší vrstvy je budou považovat za ztracené, což zapříčiní znovu odeslání těchto paketů od zdroje. Hrozí tak vyšší a zbytečné zatížení sítě. [ludvicek 8]

- **WFQ** Frontový mechanismus Weighted Fair Queuing je mechanismus s váženou spravedlivou obsluhou. Je založen na skupině front se stejnou prioritou, které jsou průběžně obsluhovány a každá z nich má svou určitou váhovou hodnotu. Podle těchto hodnot je pak frontám přidělována část kapacity výstupní linky, v případě, že fronta zrovna nevyužívá svou část kapacity linky, může být tato část kapacity přidělena jiné frontě. Součet všech váhových hodnot pak odpovídá celé šířce pásma. Mechanismus za pomoci teoretického modelu, lze jej označit jako vážená bitová obsluha, vypočítává pro každý příchozí paket čas, kdy nejpozději musí být obsloužen. Podle tohoto času je pak paket přidělen do určité fronty s určitou garancí kapacity linky.

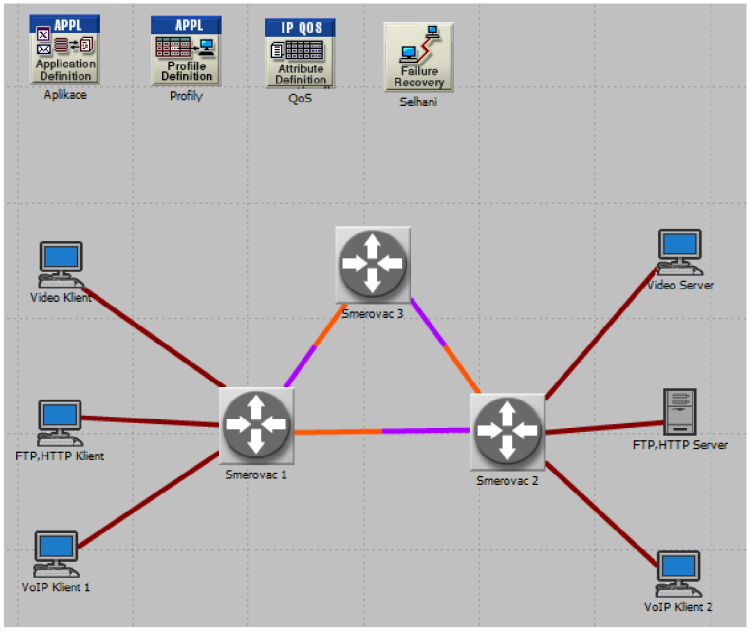
Manuál pro sestavení úlohy

1. **Sestavte síť podle obrázku a návodu** - *tuto část předpřipravuje vyučující*

Začněte spuštěním programu Riverbed a klikněte a **File** -> **New** -> **OK**. Pojmenujte projekt na "Privatni sit" a scénář na RIP FIFO, poté stiskněte **OK**.

V okně Startup Wizard: Initial Topology zanechte nastavení na **Create Empty Scenario** -> **Next** -> vyberte **Campus** -> třikrát **Next** -> **Finish**.

V okně Object Palette Tree vyberte prvek **ethernet4_slip8_gtwy** a vložte ho třikrát na simulační plochu v programu. Dále pak vyberte a umístěte na plochu pět **ethernet_wkst**, jeden **ethernet_server** a po jednom pak **Application Config**, **Profile Config**, **QoS Attribute Config** a **Failure Recovery**. Všechny prvky rozmístěte podobně jako na obrázku. *Obr.4. Topologie sítě.* K vyhledávání prvků a kabelů můžete použít vyhledávač **Search by name**, do něhož zadáte název hledané položky a stisknete **Find Next**.



Obr. č. 4 Topologie sítě

Prvky přejmenujte kliknutím pravým tlačítkem myši na prvek a výběrem možnosti **Set Name** tak jak jsou na obrázku.

Všechny klienty i servery spojte se směrovači podle obrázku linkou **10BaseT** a samotné směrovače propojte linkou **PPP-DS1**. Projekt uložte.

2. Nastavte směrovací protokol RIP

V Project Toolbaru klikněte na ikonu běžce  . V **Global attributes** rozklikněte **IP** nastavte **IP Dynamic Routing Protocol** na **RIP** a **IP Interface Addressing Mode** nastavte na **Auto Addressed/Export**. Poté rozklikněte **Simulation Efficiency** a **RIP Sim Efficiency** změňte na **Disabled**. Změny potvrďte tlačítkem **Apply**.

3. Nastavte výpadek linky

Klikněte pravým tlačítkem myši na objekt **Selhani** a zvolte **Edit Attributes -> Link Failure/Recovery Specification** -> **Number of Rows** nastavte na **1**. Rozklikněte položku **Unspecified**, nastavte Name na **Smerovac 1 <-> Smerovac 2** a **Time** na hodnotu **140**. Nastavení potvrďte stisknutím **OK**.

4. Nastavte provoz v síti - *tuto část předpřipravuje vyučující*

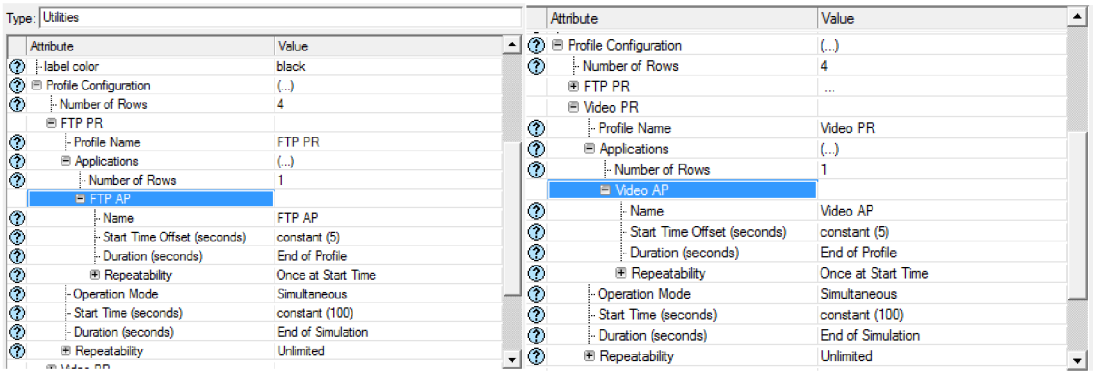
Nastavte FTP aplikaci kliknutím na objekt **Aplikace** -> **Edit Attributes** -> Rozklikněte **Application Definitions** -> **Number of Rows** nastavte na 4. Rozklikněte **Enter Application Name** -> do kolonky **Name** napište **FTP AP**, poté rozklikněte **Description** a hodnotu **Ftp** nastavte na **High Load**. Klikněte na hodnotu **High Load** -> **Edit** -> **Type of Service**. Zobrazí se okno **Configure TOS/DSCP**,

zaškrtněte **Differentiated Services Code Point** a v kolonce vyberte **CS0**. Potvrďte dvakrát **OK**. Stejným způsobem nastavte zbylé dvě aplikace. Nejprve rozklikněte **Enter Application Name** -> **Name** nastavte jako **Video AP** -> **Description** -> **Video Conferencing** -> **Low Resolution** -> **Edit** -> **Type of Service** -> **DSCP** nastavte na **AF41** -> dvakrát **OK**.

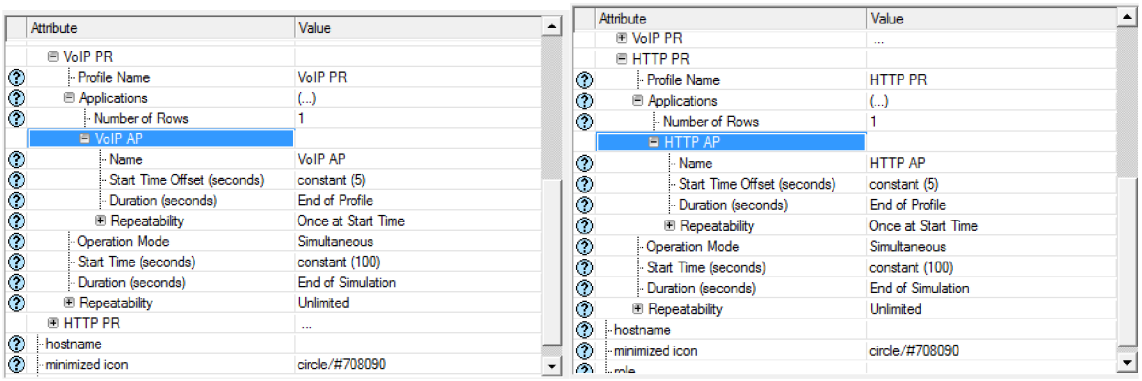
VoIP aplikaci pojmenujte jako **VoIP AP**. **Desctiption** -> **Voice** nastavte **PCM Quality Speech**, poté znovu **Edit -> Type of Service** -> **DSCP** nastavte na **AF31**. Potvrďte dvakrát **OK**.

Pro HTTP volte jméno **HTTP AP**. **Desctiption** -> Http nastavte **Video Browsing**, poté znovu **Edit** -> **Type of Service** -> **DSCP** nastavte na **AF21**. Potvrďte dvakrát **OK**.

Nastavte profily aplikací kliknutím na objekt **Profily** -> **Edit Attributes** -> Rozklikněte **Profile Configuration** -> **Number of Rows** nastavte na **4**. Jednotlivé profily pojmenujte postupně po každé aplikaci, například **FTP PR**. Poté každý profil nastavte podle obrázků níže. Nakonec potvrďte stisknutím **OK** a projekt uložte.



Obr. č. 5 Profil FTP a Profil Video



Obr. č. 6 Profil VoIP a Profil HTTP

Nastavení služeb FTP a HTTP na klientském PC proveďte kliknutím pravým tlačítkem myši na **FTP, HTTP Klient** -> **Edit Attributes** -> **Application : Suported Profiles** -> **Edit** -> **Rows** nastavte na **2** -> **Profile Name** pro první řádek vyberte **FTP PR** a pro řádek druhý **HTTP PR**. Nastavené potvrďte dvakrát tlačítkem **OK**.

Klikněte pravým tlačítkem myši na **FTP, HTTP Server** -> **Edit Attributes** -> **Application: Suported Services** -> **Edit** -> **Rows** nastavte na **2** -> **Service Name** pro první řádek vyberte **FTP AP** a pro řádek druhý **HTTP AP**. Nastavené potvrďte dvakrát tlačítkem **OK**.

Nastavení služby Video proveďte kliknutím pravým tlačítkem myši na **Video Klient** -> **Edit Attributes** -> **Application : Suported Profiles** -> **Edit** -> **Rows** nastavte na **1** -> **Profile Name** vyberte **Video PR**. Nastavené potvrďte dvakrát tlačítkem **OK**.

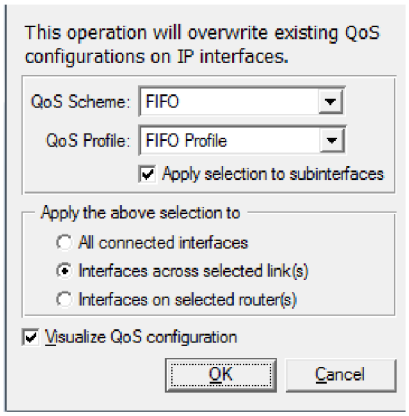
Kliknutím pravým tlačítkem myši na **Video Server** -> **Edit Attributes** -> **Application: Suported Services** -> **Edit** -> **Rows** nastavte na **1** -> **Service Name** vyberte **Video AP**. Nastavené potvrďte dvakrát tlačítkem **OK**.

Nastavení služby VoIP proveďte kliknutím pravým tlačítkem myši na **VoIP Klient 1** -> **Edit Attributes** -> **Application : Suported Profiles** -> **Edit** -> **Rows** nastavte na **1** -> **Profile Name** vyberte **VoIP PR**. **Application : Suported Services** -> **Edit** -> **Rows** nastavte na **1** -> **Service Name** vyberte **VoIP AP**. Nastavené potvrďte dvakrát tlačítkem **OK**.

Stejné nastavení proveďte i pro druhé koncové zařízení, **VoIP Klient 2**. Projekt uložte.

5. Nastavení frontových mechanismů FIFO, PQ, WFQ a simulace

Pro nastavení fronty FIFO označte všechny linky mezi směrovači. Přidržte klávesu Ctrl a levým tlačítkem myši vyberte linky. Poté klikněte v horní liště na **Protocols** -> **IP** -> **QoS** -> **Configure QoS** a ujistěte se, že je nastavení shodné s tím na obrázku Obr. č2 a potvrďte tlačítkem **OK**.



Obr. č. 7 Nastavení front

Vyberte, jaké statistiky budete měřit. Pravým tlačítkem myši klikněte kdekoliv na volnou plochu v projektu a zvolte **Choose Individual Statistics**.

V **Global Statistic** zaškrtněte tyto položky:

IP -> Traffic Dropped (packets/secs)

V **Node Statistic** zašktněte :

RIP -> Traffic Received (bits/sec)

OSPF -> Traffic Received (bits/sec)

Nakonec potvrďte stisknutím **OK** a projekt uložte.

V Project Toolbaru klikněte na ikonu běžce  a spusťte simulaci tlačítkem **Run**. Po skončení simulace můžete ověřit výsledky kliknutím na ikonu **Results Browser** a v záložce **DES Graphs** ve druhém okně ověřit naměřené hodnoty jednotlivých statistik.

Projekt uložte a pokračujte duplikací scénáře pro zbylé druhy front. V záložce **Scenarios** klikněte na **Duplicate Scenario** a pojmenujte ho na **RIP PQ**.

Přenastavte QoS na Priority Queuing. Označte linky mezi směrovači a pak klikněte na: **Protocols** -> **IP** -> **QoS** -> **Configure QoS**. **QoS Scheme** nastavte na **Priority Queuing** a **QoS Profile** na **DSCP Based**. Ujistěte se, že jste označili všechny linky mezi směrovači. Nastavení potvrďte stisknutím **OK**, projekt uložte a spusťte simulaci. Po skončení simulace znovu uložte projekt a duplikujte scénář. Ten pojmenujte jako RIP WFQ, QoS přenastavte na schéma WFQ podobným způsobem jako u scénáře RIP PQ a spusťte simulaci. Znovu uložte.

6. Nastavení směrovacího protokolu OSPF a simulace

Duplikujte scénář a přejmenujte ho na OSPF WFQ.

Změňte směrovací protokol z RIP na OSPF. Podobně jako při nastavování směrovacího protokolu RIP

Klikněte na ikonu běžce (Cinfigure/Run DES) a v **Global attributes** změňte v **IP** -> **IP Dynamic Routing Protocol** na **OSPF**, v **Simulation Efficiency** změňte **OSPF Sim Efficiency** na **Disabled** a **RIP Sim Efficiency** na **Enabled**. Změny potvrďte stisknutím **Apply** a spusťte simulaci tlačítkem **Run**. Po skončení simulace projekt uložte. Duplikujte scénář a pojmenujte ho OSPF PQ, v něm pak přenastavte QoS na PQ stejným způsobem jako v RIP PQ scénáři a následně spusťte simulaci. Uložte projekt. Obdobným způsobem duplikujte a přejmenujte na scénář OSPF FIFO, přenastavte QoS schéma na FIFO uložte a simulujte.

7. Generace výsledků

Vygenerujte grafické výsledky z laboratorní úlohy kliknutím na ikonu Results Browser po skončení simulace, nebo kliknutím pravým tlačítkem myši na volné pole projektu a zvolením View Results.

V kolonce **Results for**: vyberte **Current Project** a zaškrtněte všechny RIP scénář.

Z **Global Statistic** vygenerujte graf:

IP -> Traffic Dropped (packets/sec) (Presentation -> Overaid Statistics)

Z **Object Statistics** po rozkliknutí **Campus Network** vygenerujte ze směrovače 1 graf RIP Traffic Received (bit/sec) (Presentation -> Stacked Statistics).

Označte všechny OSPF scénáře, zrušte označení všech RIP scénářů a vygenerujte ty samé typy grafů jako u RIP scénářů jen u směrovače 1 místo RIP vyberte OSPF Traffic Received (bit/sec).

Vygenerované grafy si uložte například funkcí Print Screen.

Dodatečné otázky a úkoly:

Jaká byla prodleva přepnutí na záložní linku u obou směrovacích protokolů? Tuto prodlevu zdůvodněte.

Zhodnoťte zahazování IP paketů všech frontových mechanismů pro oba směrovací protokoly, proč těsně po přepnutí záložní linky je viditelný pokles zahození paketů?

(Choose Individual DES -> Node Statistic -> IP -> vyberte Traffic Received a Traffic Sent a spusťte znovu simulaci aktuálního scénáře. Ve výsledcích vygenerujte Traffic Dropped z prvního směrovače a Traffic Received ze třetího směrovače.)

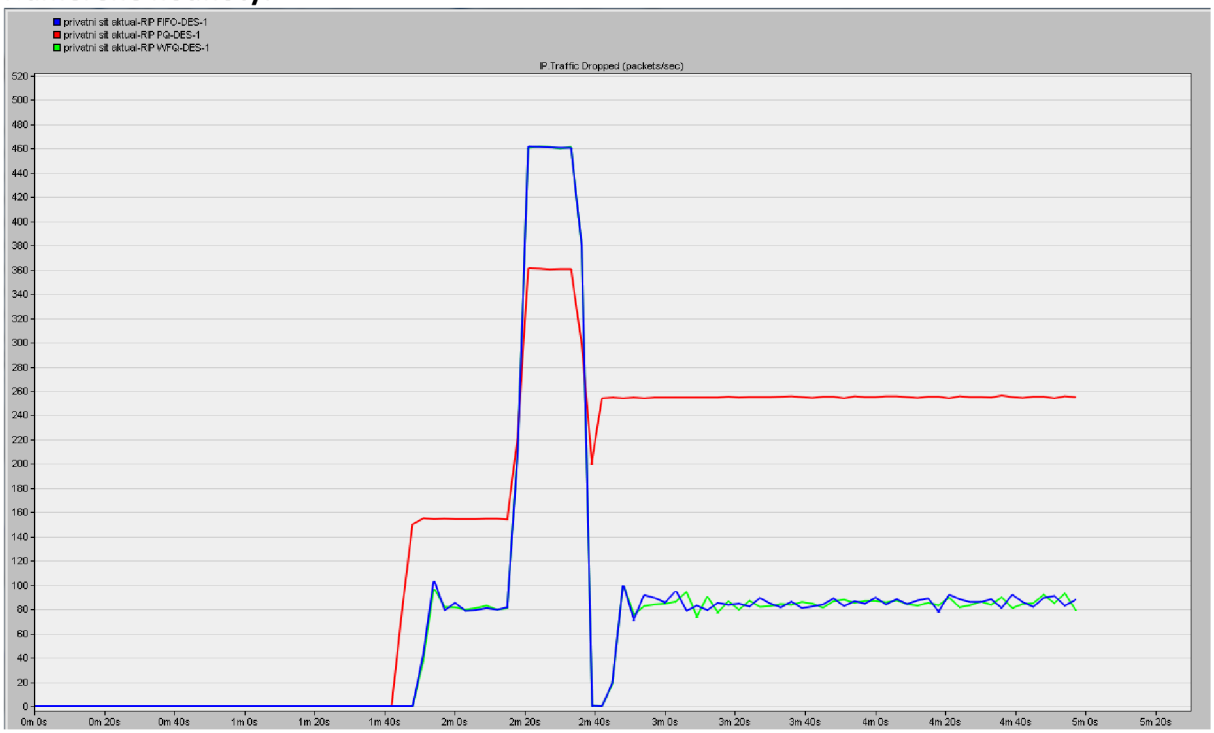
Nakonec prozkoumejte příjmy směrovacích dat obou směrovacích protokolů.

FEKT	Předmět		
	Jméno		
	Ročník	Studijní skupina	
	Spolupracoval	Měřeno dne	
Kontroloval	Hodnocení	Dne	
Číslo úlohy	Název úlohy **Testování QoS front s kombinací směrovacích protokolů**		

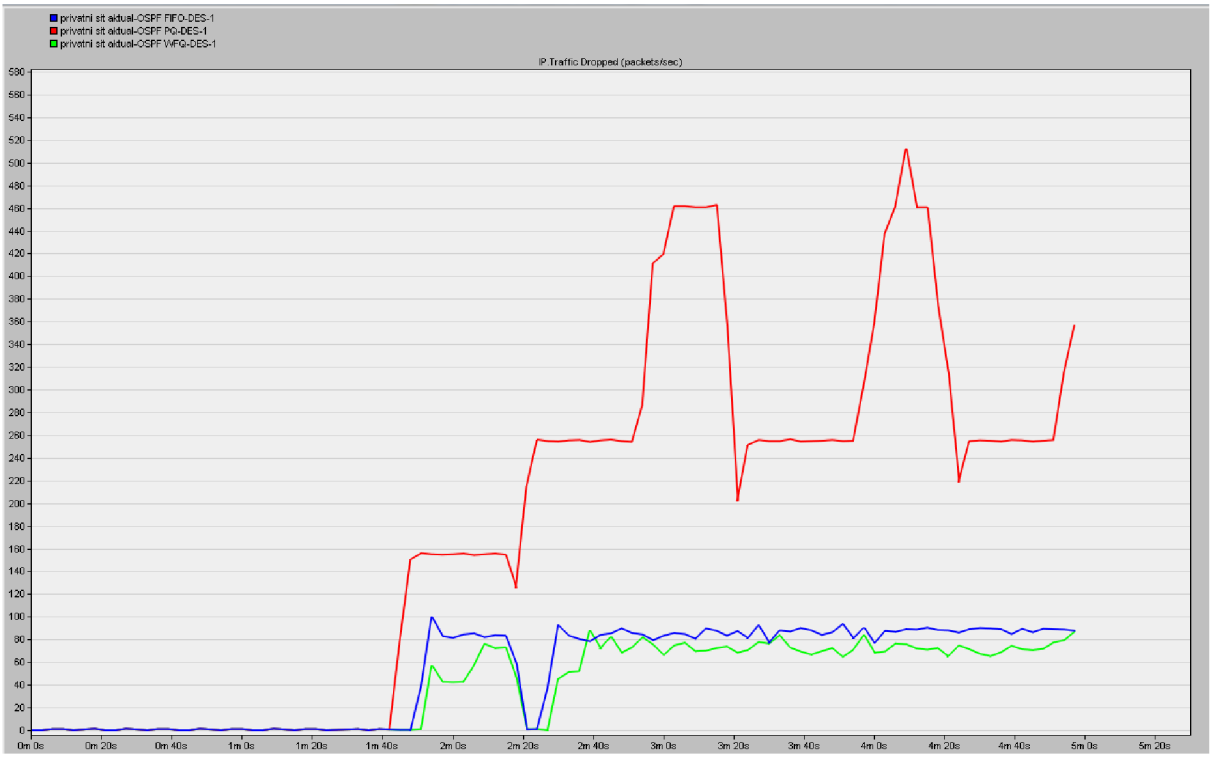
Zadání:

1. Nastavte směrovací protokol RIP
2. Nastavte výpadek linky
3. Nastavte frontové mechanismy FIFO, PQ a WFQ a postupně je proměřte.
4. Síť přenastavte na směrovací protokol OSPF a proměřte frontové mechaniky stejně jako v předchozím bodě.
5. Porovnejte výsledky všech a frontových mechanismů se směrovacími protokoly.

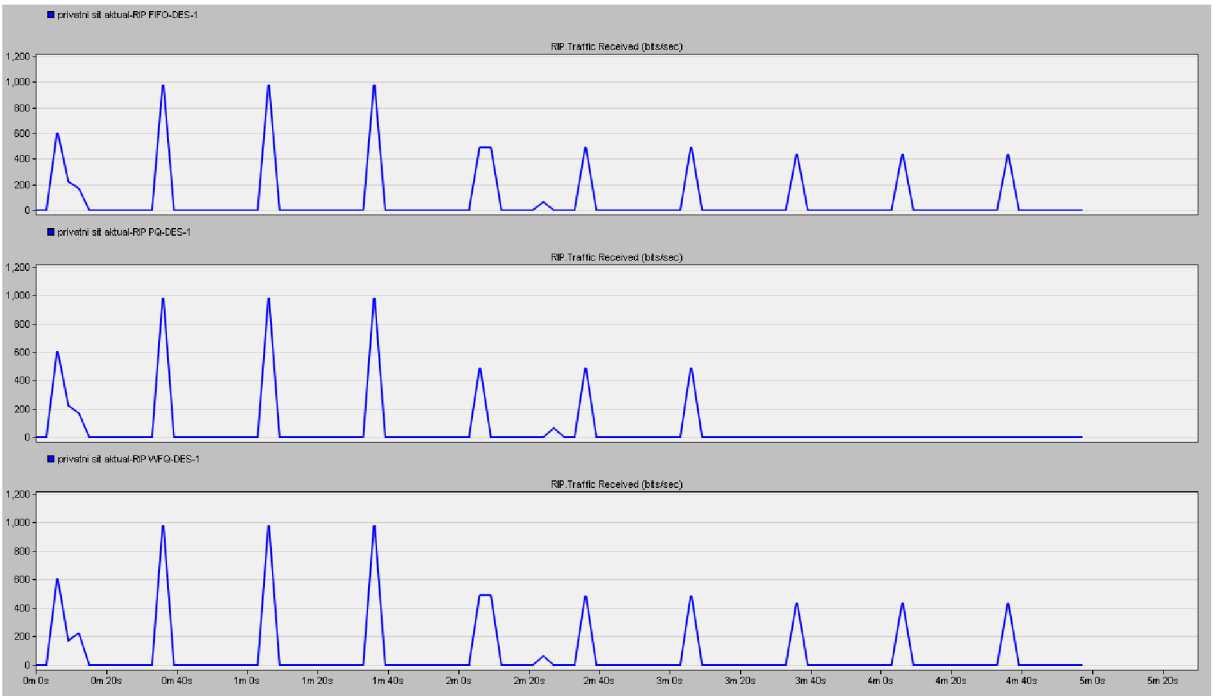
Naměřené hodnoty:



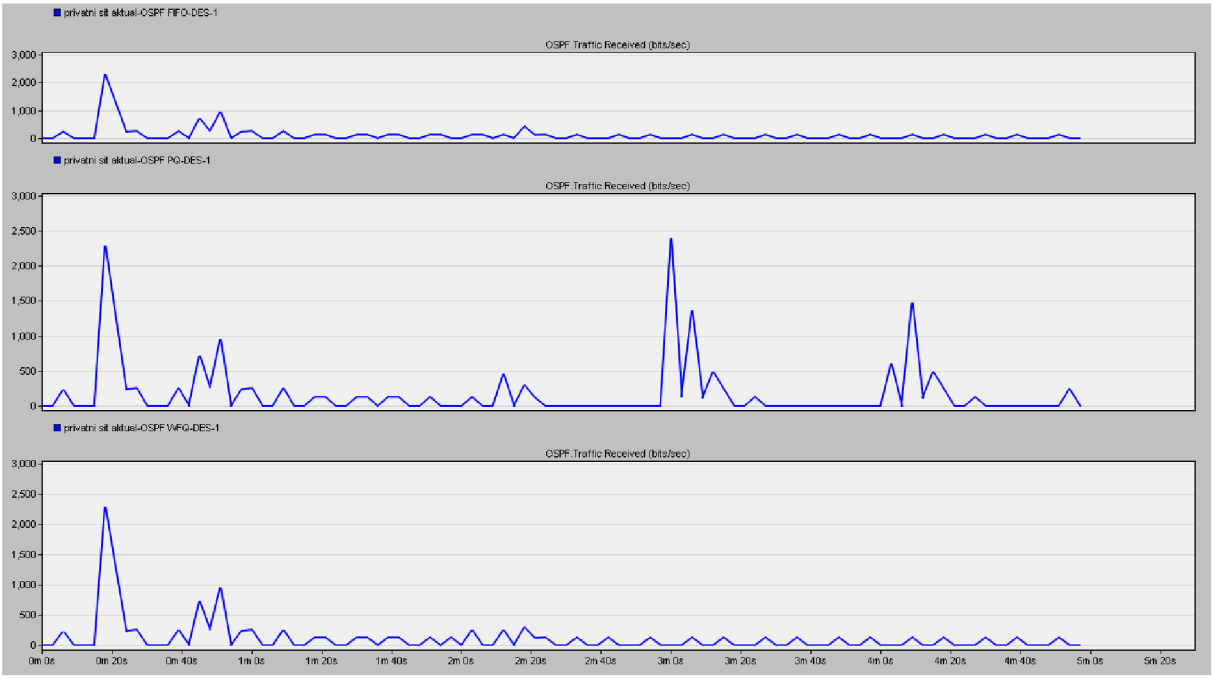
Graf. č. 1: Zahazování IP paketů s použitím RIP protokolu



Graf. č. 2: Zahazování IP paketů s použitím OSPF protokolu



Graf. č. 3: Příjem směrovacích dat protokolu RIP na směrovači 1



Graf. č. 4: Příjem směrovacích dat protokolu OSPF na směrovači 1

Závěr:

Prodleva přepnutí na záložní linku u protokolu RIP byla zhruba 13 vteřin (10 vteřin po výpadku došlo k rozeslání směrovacích tabulek a do 3 vteřin došlo k navázání nového spojení), důvodem tak velké prodlevy je rozesílání aktualizovaných tabulek každých 30 vteřin. Prodleva u OSPF protokolu není měřitelná a to proto, že topologie sítě je velmi jednoduchá a selhání linky bylo nastaveno na 140 vteřin po spuštění simulace. OSPF každých 10 vteřin ověřuje spojení se sousedem.

Zahazování IP paketů se prudce zvýšilo při výpadku linky ve scénářích s RIP protokolem podle očekávání, u scénářích s OSPF protokolem zahazování paketů kvůli výpadku není znatelné, ale je zde více patrný pokles zahazování těsně po navázání nového spojení. Ten je zapříčiněn tím, že náhradní linka nebyla využívaná před navázáním záložního spojení. Poté se linka začala využívat a zahazování paketů se vrátilo do normálu. Stejné odůvodnění mají poklesy zahazování i v simulacích s RIP protokolem.

Dále můžeme vidět u simulací RIP i OSPF velké zahazování při použití PQ fronty, to může být zapříčiněno zahlcením všech front ale obsloužením pouze fronty s nejvyšší prioritou a neobsloužením dalších služeb. Zbylé frontové mechanismy jsou si téměř podobné, avšak WFQ je o trochu lepší než FIFO v simulacích s OSPF protokolem.

V grafu číslo 3 lze pozorovat periodické rozesílání směrovacích tabulek po síti. Po výpadku linky mezi směrovači 1 a 2 je patrný i pokles množství dat, to je zapříčiněno snížením množství informací ve směrovacích tabulkách. Ve scénáři PQ lze pozorovat úplné zaniknutí příjmu směrovacích tabulek, to mohlo být způsobeno funkcí QoS. Tedy byla obsluhována jen fronta s vysokou prioritou a ostatní fronty nebyly obslouženy, tento problém může mít za následek ochromení RIP protokolu, a i celé sítě.

V simulacích s OSPF protokolem můžeme pozorovat obvyklé fungování směrovacího protokolu OSPF v grafu číslo 4. Na začátku grafu lze vidět vysoký příjem dat, ten je zapříčiněn sestavováním spojení mezi směrovači a výměnou dat mezi nimi. Po skončení sestavování spojení lze už jen pozorovat příjem Hello a reakci na výpadek linky v čase 2 minuty a 20 sekund, tato reakce je pozorovatelná i ve scénářích s RIP protokolem. Ve scénáři PQ lze vidět stejný problém jako u scénáře RIP PQ, jen s tím rozdílem, že občas došlo k návalovému příjmu směrovacích dat.

Testování privátní sítě – Testování QoS v kombinaci se směrovacími protokoly v emulátoru EVE-NG

Úvod k úloze

Cílem laboratorní úlohy je prozkoumání QoS frontových FIFO, PQ a WFQ mechanik v kombinaci směrovacích protokolů RIP a OSPF v emulátoru EVE-NG. Síť je tvořena pěti směrovači a dvěma koncovými zařízeními typu Linux, která zastupují role serveru a klienta. V šesti scénářích postupně spusťte zátěžový TCP a UDP test pomocí programu Iperf, během testu simulujte výpadek hlavní linky vypnutím směrovače R2 a průběžně zaznamenávejte programem Wireshark příchozí data na směrovači R3. Zjistěte rychlost reakce směrovacích protokolů při výpadku linky a prozkoumejte příjmy dat včetně výsledků programu Iperf ze všech scénářích.

Výstupem laboratorní úlohy by měla být tabulka výsledků z testovacího programu a skupina grafů zobrazujících celkový příjem všech dat, příjem TCP/UDP z testu a průběh detekce směrovacích informací na směrovači R3.

Body laboratorní úlohy

1. Vytvoření laboratoře pro první scénář – *vyučující*
2. Sestavení sítě a instalace programu Iperf – *vyučující*
3. Nahrání konfigurací do směrovačů – *vyučující*
4. Vytvoření kopií laboratoře pro další scénáře – *vyučující*
5. Měření laboratorní úlohy – *student*

Teoretický úvod

Směrování je technika, která propojuje jednotlivé sítě po síťové vrstvě. Tato technika v podstatě nachází nejlepší cesty skrze mezilehlé uzly k propojení uživatelů. K tomu jsou využívány různé protokoly, které mají kromě úkolu najít nejvhodnější cesty taky v případě přerušení již používané cesty, například výpadkem mezilehlého uzlu nebo přerušení linky, najít vhodné náhradní cesty.

Routing Information Protocol

Dynamický směrovací protokol a spadá pod třídu Distance Vector protokolu, který pracuje s metrikou počítání skoků k cíli a pravidelným rozesíláním směrovacích tabulek sousedním směrovačům. Maximální počet skoků je 15 a nejlepší cesta je označena jako ta s nejmenším počtem skoků. Pokud je cílová síť vzdálená 16 a více skoků skrze směrovače, je označena za nedosažitelnou. Tento protokol určuje pouze jednu cestu k cíli, a tedy jasnou nevýhodou je nemožnost rozložení zátěže na více cest. RIP protokol existuje ve verzích RIPv1, RIPv2 a RIPng/RIPv6.

RIP protokol využívá pro své fungování 4 časovače:

- **Update Timer** – je časovač, po jehož vypršení směrovač posílá všem svým sousedním směrovačům svou celou směrovací tabulku. (defaultně 30 vteřin)

- **Invalid Timer** – časovač odpočítává dobu platnosti cesty (řádku) ve směrovací tabulce. Pokud před vypršením časovače obdrží směrovač aktualizaci cesty tak je časovač resetován na původní hodnotu, v případě vypršení časovače je cestě přiřazena metrika 16 a přesouvá se do stavu hold-down. (defaultně 180 vteřin)
- **Hold-down Timer** – odpočítává dobu, kdy jsou ještě informace o cestě přidrženy, ale již ji nejde obnovit ani v případě obdržení nových aktualizací o dosažitelnosti cíle. Tento časovač slouží k poskytnutí dostatku času k přizpůsobení změnám v síti. (defaultně 180 vteřin)
- **Flush Timer** – slouží k vymazání záznamů o cestě a běží současně s Invalid Timer časovačem. Po 60 vteřinách od označení cesty za neplatnou, dojde k jejímu vymazání. (defaultně 240 vteřin)

Rozdíl mezi verzemi RIPv1 a RIPv2 protokolu je ten, že druhá verze podporuje třídní adresování, tedy proměnné délky masek podsítí, což má za následek efektivnější sumarizaci a méně adres ve směrovacích tabulkách v jednotlivých směrovačích. RIPv2 také podporuje autentizaci za pomoci MD5 šifrování.

Open Shortest Path First

Protokol typu Link State a směrovač s využitím tohoto protokolu získává informace o celé topologii sítě za pomoci skupiny paketů. Nejvhodnější cesta se určuje na základě topologie a metriky linek mezi jednotlivými uzly, která se skládá například ze zpoždění, zabezpečení a šířky pásma. Výsledná hodnota se pak označuje jako „cena spoje“ a nejvýhodnější cesta je ta s nejmenší hodnotou. Tento protokol využívá Dijkstrova algoritmu k odvození nejvhodnější cesty a podporuje dělení sítí na subsítě. Rozdělením na subsítě (oblasti) lze snížit počet zpráv, které si směrovače posílají a zmenšit tak celkové zatížení sítě. Směrovače znají topologii jen o dané oblasti, ve které se nacházejí a informace o okolních sítí nebo subsítí jim poskytují hraniční směrovače, které propojují oblasti mezi sebou. Hraniční směrovače shrnují informace o sítích, v nichž se nachází a posílají je do sousedních sítí. Výhodou tohoto protokolu oproti RIPv2 protokolu je hlavně, že při změně v síti směrovač neposílá celou směrovací tabulku, což značně snižuje zatížení v síti a jeho reakce na změny v topologii je rychlejší. Směrovač si kontroluje dostupnost sousedních směrovačů a pokud dojde ke zemně v sítí tak ihned zasílá informaci jen o dané změně hierarchicky všem směrovačům v síti. Směrovače si přeposílají aktualizace vždy při změně anebo každých 30 minut v případě, že nenastane žádná změna. U tohoto protokolu není omezen počet skoků a cenu spoje lze nastavit manuálně, a tak upřednostnit pomalejší cestu v případě nízkých požadavků na přenos. Protokol podporuje proměnné délky masky podsítí a druhá verze podporuje autentizaci za pomoci šifrování MD5.

OSPF pro své fungování využívá 5 druhů paketů:

- **Hello** – slouží k navázání a udržení spojení se sousedním směrovačem a využívá dvou časovačů. Hello interval udržuje spojení a je posílán každých 10 vteřin. Dead interval je zpravidla čtyřnásobek Hello intervalu, tedy 40 vteřin a pokud směrovač nepřijme žádný Hello paket od souseda do vypršení tohoto intervalu spojení mezi směrovači zanikne.
- **Database Description** – slouží k přeposílání informací o topologii sítě.
- **Link State Request** – žádá o zaslání chybějící položky po obdržení paketu DBD.
- **Link State Update** – je odpověď na LSR paket s chybějící položkou.
- **Link State Acknowledgment** – slouží jako potvrzení o přijetí paketu LSU.
- Při navazování nového spojení a výměnu směrových informací mezi směrovači procházejí směrovače 7 stavy:

- **Down** – je stav před obdržení Hello paketu, kdy ještě směrovač neví o existenci souseda.
- **Init** – stav je po obdržení Hello paketu, stáje ještě bez zařízení obousměrné komunikace.
- **Two-Way** – označuje stav, kdy je zřízena obousměrná komunikace a určuje se pověřený směrovač a záložní pověřený směrovač. Všechny směrovače pak při změně v topologii komunikují přes pověřený směrovač.
- **Exstart** – ustanovuje směrovače Master a Slave podle velikosti sekvenčního čísla v paketu DBD. Komunikaci začíná směrovač Master.
- **Exchange** – výměna DBD paketů mezi směrovači.
- **Loading** – výměna paketů LSR, LSU a LSA.
- **Full** – je stav plné synchronizace mezi směrovači.

QoS Quality of Services

Technologie OoS je skupina podpůrných funkcí, které mají zajistit upřednostnění přenosu dat pro náročnější služby před nenáročnými službami. Obecně se technologií QoS podporují služby v reálném čase, jako jsou například hlasové služby, live streaming, videokonference, nicméně je nutné zajistit i funkčnost nenáročných služeb (spolehlivý přenos dat, emailové služby, chat, a td). QoS tedy musí efektivně rozdělovat síťové prostředky pro služby realtime i obecné služby, k tomu je nutné provádění klasifikaci a značení datového provozu. Tato klasifikace probíhá na hranici domény, tedy na hraničním přepojovacím prvku, popřípadě s příslušnou dohodou lze přebírat způsob klasifikaci datových toků od poskytovatele. Upřednostnění různých služeb lze použít i pro přepojovací prvky uvnitř sítě. Vložením informací do hlaviček datových jednotek o klasifikaci se umožní prioritní obsluhování služeb uvnitř sítě, za použití jednoho z frontových mechanismů. V těchto mechanismech se třídí datové jednotky podle priority do front, ze kterých se jim přidělují dostupné a požadované síťové prostředky.

Mezi důležité parametry pro službu QoS jsou:

- **Jitter** – proměnlivost zpoždění je negativní parametr zejména pro služby pracující v reálném času. Zpoždění se dynamicky mění podle aktuálního zatížení a stavu sítě. Pro eliminaci tohoto parametru se na příjímací straně využívá Jitter Bufferu, který slouží jako zpožďovací paměť a mění tak zpoždění z dynamického na konstantní.
- **Zpoždění** – je parametr, který sčítá celkově všechna zpoždění počínaje od kódování zdroje, paketizační zpoždění, propagační zpoždění po Jitter Buffer zpoždění. Propagačním zpožděním se myslí potřebný čas k přenosu dat od zdroje k cíli po síti. Paketizační zpoždění je čas strávený umísťování bitů do paketů, například do hlaviček. Kódováním zdroje se myslí A/D a D/A převod hlasu u zdroje a cíle.
- **Ztrátovost paketů** – určuje množství paketů které buď dorazili pozdě a u služeb pracujících v reálném čase se nedají už použít, nebo které vůbec nedorazili. Jde taktéž o negativní parametr, který se však u služeb v reálném čase dá do jisté míry tolerovat. Ztrátovost paketů může například zhoršit kvalitu hlasu či videa anebo u TCP služeb může zapříčinit nutnost znovu odeslání dat.

Pro provoz sítě podporující prioritizaci služeb existují dva modely řízení sítě. Prvním modelem je Integrated Services (**IntServ**), model rezervuje zdroje v síťových prvcích po celé cestě mezi zdrojem a příjemcem dat. Rezervace zdrojů v síti obsahuje nalezení vhodné cesty skrze uzly v síti a alokaci zdrojů pro každý směr přenosu od příjemce. U každého směrovače v cestě se zjišťuje požadavek

oprávnění a dostatek zdrojů pro uskutečnění relace. K tomu se používají různé protokoly jako například RSVP (Resource Reservation Protocol) a COPS, které využívá firma CISCO. Druhým modelem je Differentiated Services (**DiffServ**). Síť s využitím tohoto modelu třídí datové toky jen na hraničních uzlech. V těchto uzlech síťové prvky třídí datové toky stejných druhů do jednotlivých tříd, a to úpravou DSCP pole. Nerozlišují tak například jednotlivé hovory VoIP, ale upřednostní všechny VoIP hovory jako skupinu. Prvky uvnitř sítě se řídí podle PHB (Per Hop Behavior). Služba DiFfServ je implementována například ve standardech RFC 247, RFC 2475 a RFC 2598.

Systém proti zahlcení front

Random Early Detection a Weighted Random Early Detection metody slouží k prevenci proti zahlcení front. Pokud se problém zahlcení front začne řešit až při jeho uskutečnění, tedy zaplní se fronty, nově příchozí pakety budou zahazovány, a to bez ohledu na jejich prioritu do doby uvolnění front. Tento problém může tvořit ještě další problém a tím je tvoření zahlcovacích vln na směrovačích střídající se s jevem sníženým tokem dat přes síť. K tvorbě tohoto problému přispívá TCP protokol, dojedli k zahození paketů využívající TCP protokol tak jsou tyto pakety znovu odeslány sníženou rychlostí odesílání. Pokud dojde k hromadnému zahození takových paketů od více uživatelů tak dojde k celkovému snížení rychlosti odesílaných dat a nevyužití kapacity linky. Po obnovení rychlostí u uživatelů může dojít k opětovnému zahlcení front ve směrovačích a celý problém se opakuje.

Takové problémy eliminují metody RED a WRED. RED metoda zahazuje pakety z TCP spojení s určitou pravděpodobností od určitého procenta zaplnění fronty. Pravděpodobnost zahození roste se zvyšujícím se zaplnění fronty. Při zahazování paketů dojde ke snížení datových toků u některých uživatelů, ale zabrání se tím zaplnění front a vyrovnání příchozích a odchozích datových objemů. Metoda WRED je modifikovaná metoda, která je schopna rozlišit klasifikaci dat, a tak může méně prioritním datům přiřadit větší pravděpodobnost zahození a datům více prioritním naopak pravděpodobnost zahození snížit.

Fronty v technologii DiffServ

- **FIFO** Jedná se o nejjednodušší mechanismus fronty bez jakéhokoliv algoritmu řízení. Pakety jsou obsluhovány tak jak přijdou, podle čehož nese tento mechanismus i své jméno „First In First Out“. Výhoda tohoto mechanismu je jeho jednoduchost, je zapotřebí jen jedna vyrovnávací paměť, která odesílá data ve stejném pořadí, ve kterém přišly. Hlavní nevýhodou je taktéž jednoduchost, kvůli absenci algoritmu řízení tento mechanismus nerozlišuje druhy dat, třídy provozu, a v případě zaplnění front je se všemi druhy dat zacházeno stejně. Tento mechanismus je vhodný pro službu Besteffort, pro provoz bez podpory QoS. Fronta FIFO je znázorněna na Obr.

- **PQ** Mechanismus front Priority Queunig řeší neduhy fronty FIFO, ale zároveň si zachovává jednoduchost, ovšem i tento mechanismus má své nevýhody. PQ mechanismus třídí klasifikované pakety do front podle priorit. Fronty s pakety s vyšší prioritou jsou obsluhovány do doby vyprázdnění fronty, pak mechanismus začne obsluhovat frontu s nižší prioritou. Zde může vzniknout problém „uvíznutí“ paketů ve frontách s nižší prioritou, pokud fronta s vyšší prioritou je stále doplňována dalšími

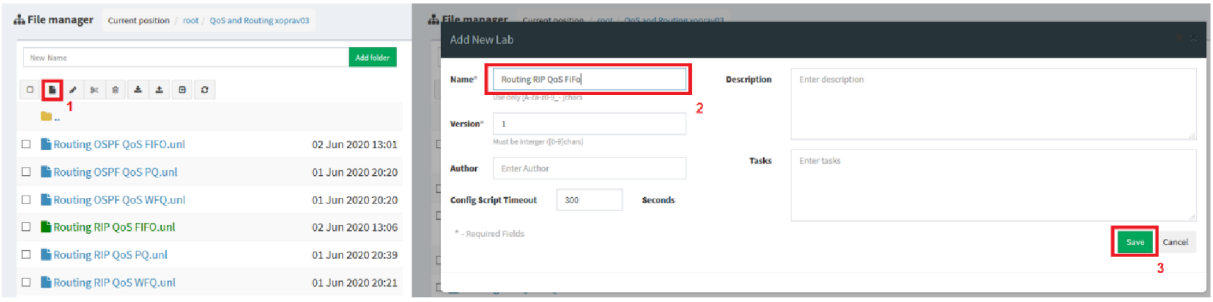
pakety. Tento mechanismus je vhodný pro služby založené na UDP protokolu, jako například stream video či videohovor. Mechanismus PQ lze rozdělit například do dvou front, kde první fronta bude pro služby pracující v reálném čase a druhá fronta bude obsluhovat služby založené na TCP protokolu a budou tedy obsluhovány službou Best Effort. Při této implementaci je však nutné počítat s rizikem, že se pakety ve druhé frontě zpozdí a vyšší vrstvy je budou považovat za ztracené, což zapříčiní znovu odeslání těchto paketů od zdroje. Hrozí tak vyšší a zbytečné zatížení sítě. [ludvicek 8]

- **WFQ** Frontový mechanismus Weighted Fair Queuing je mechanismus s váženou spravedlivou obsluhou. Je založen na skupině front se stejnou prioritou, které jsou průběžně obsluhovány a každá z nich má svou určitou váhovou hodnotu. Podle těchto hodnot je pak frontám přidělována část kapacity výstupní linky, v případě, že fronta zrovna nevyužívá svou část kapacity linky, může být tato část kapacity přidělena jiné frontě. Součet všech váhových hodnot pak odpovídá celé šířce pásma. Mechanismus za pomoci teoretického modelu, lze jej označit jako vážená bitová obsluha, vypočítává pro každý příchozí paket čas, kdy nejpozději musí být obsloužen. Podle tohoto času je pak paket přidělen do určité fronty s určitou garancí kapacity linky.

Manuál pro sestavení úlohy a proměření úlohy

1. **Vytvoření laboratoře pro první scénář:**

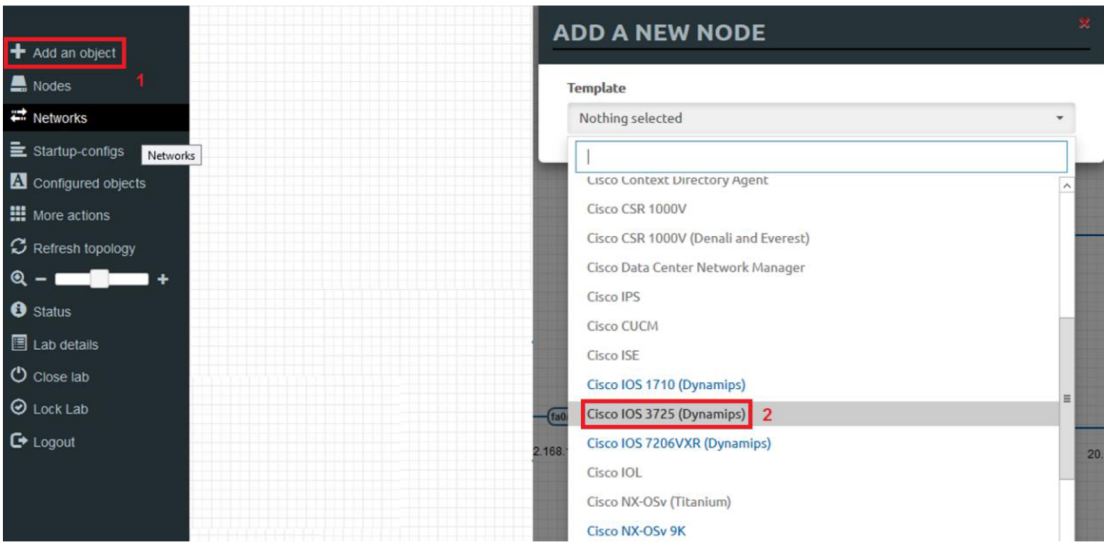
První laboratoř vytvořte nejprve kliknutím na ikonu pro vytvoření nové laboratoře, poté se zobrazí okno s kolonkami pro detaily laboratoře. Do kolonky vepište název první laboratoře „Routing RIP QoS FIFO“ a klikněte na tlačítko **Save**, čímž potvrdíte vytvoření laboratoře.



Obr č.1: File manager

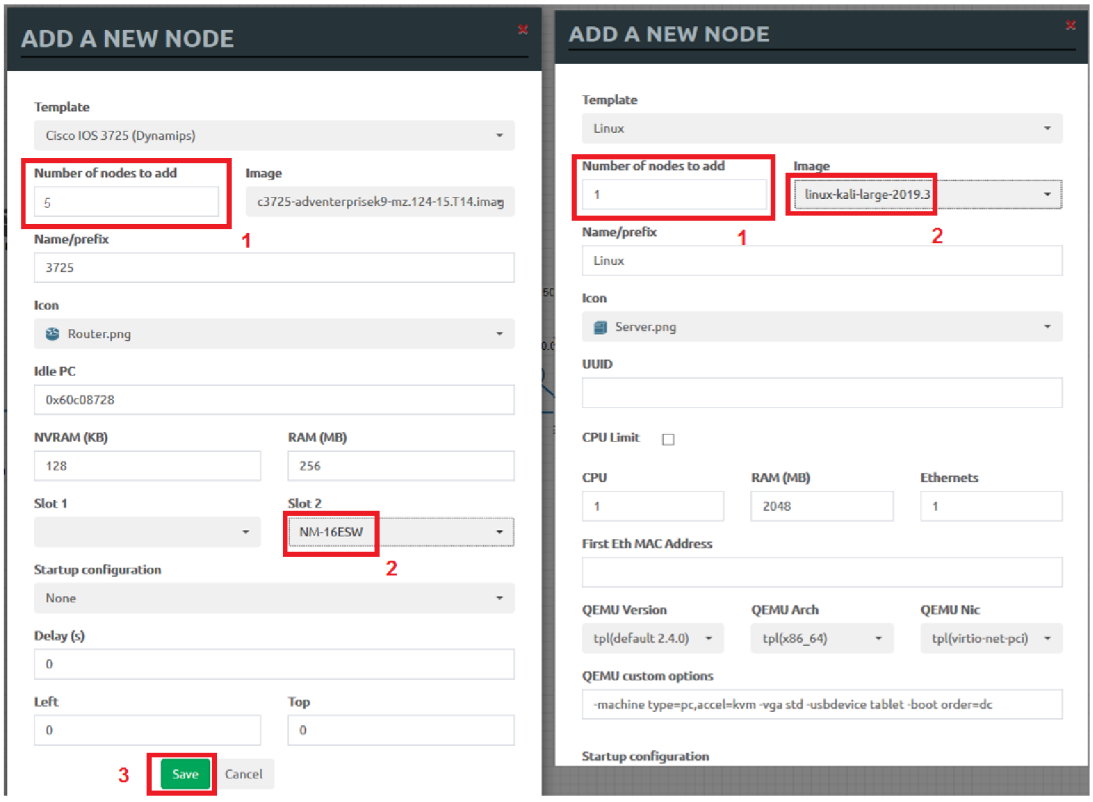
2. **Sestavení sítě podle a instalace programu Iperf na zařízení Linux:**

Po vytvoření laboratoře nejprve přidejte síťové prvky. V levém výběru možností zvolte **Add an object** a dále vyberte možnost **Node**.



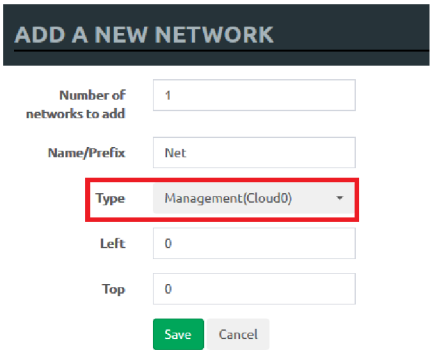
Obr č.2: Nový objekt

Po zobrazení okna **ADD A NEW NODE** vyberte směrovač **Cisco 3725**, zobrazí se nové okno s detailnějším výběrem zařízení. Zde podle obrázku [obr new node cisco a linux] nastavte **Numbers of nodes to add** na hodnotu **5** a ve slotě 1 či 2 vyberte rozšíření **NM-16ESW**. Výběr potvrďte tlačítkem **Save**. Dále podobným způsoben do laboratoře přidejte zařízení Linux, ve výběru detailnějšího nastavení nastavte **Numbers of nodes to add** na hodnotu dvě a v kolonce **Image** vyberte **linux-kali-large-2019.3**.



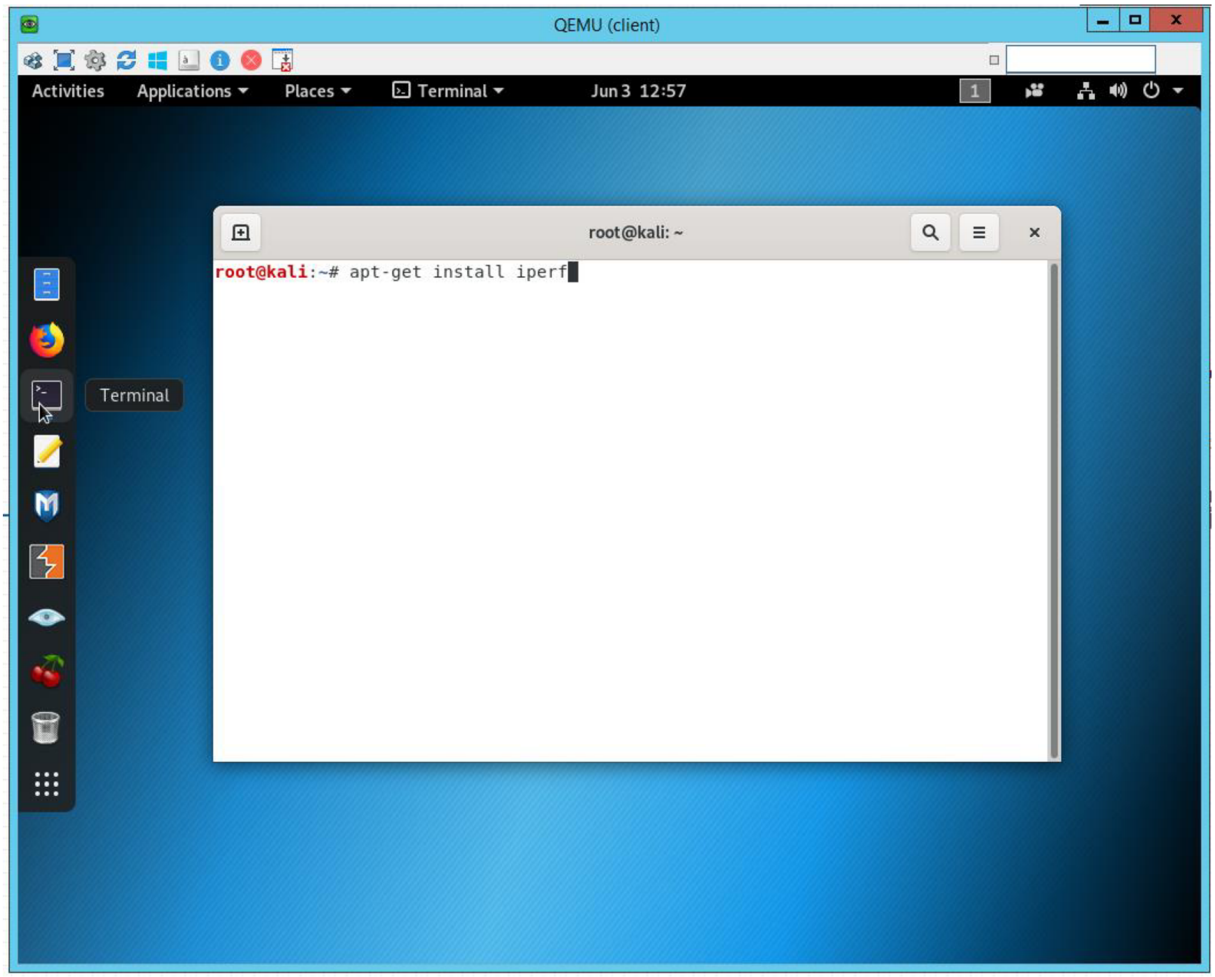
Obr č.3: Nastavení objektu

Na linuxová zařízení je nutno nainstalovat ještě program Iperf. Nejprve přidejte objekt **Net** opět kliknutím na možnost **Add an object**, zde vyberte položku **Network** a v novém okně zvolte **Managment(Cloud0)** v kolonce **Type**, potvrďte tlačítkem **Save**.



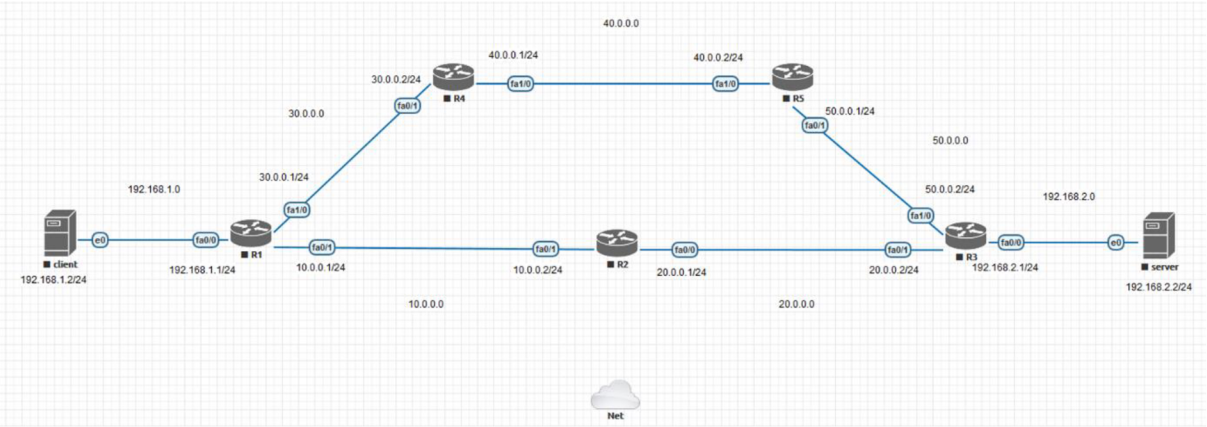
Obr č.4: Objekt Net

Dále připojte stisknutím a tažením myší linuxová zařízení k objektu **Net** a spusťte je pravým tlačítkem a výběrem Start. Do virtuálních linuxových zařízeních se přihlaste pomocí uživatelského jména **root** a hesla **toor**. Po načtení plochy otevřete terminál a nainstalujte program Iperf příkazem: **apt-get install iperf.**



Obr č.5: Iperf instalace

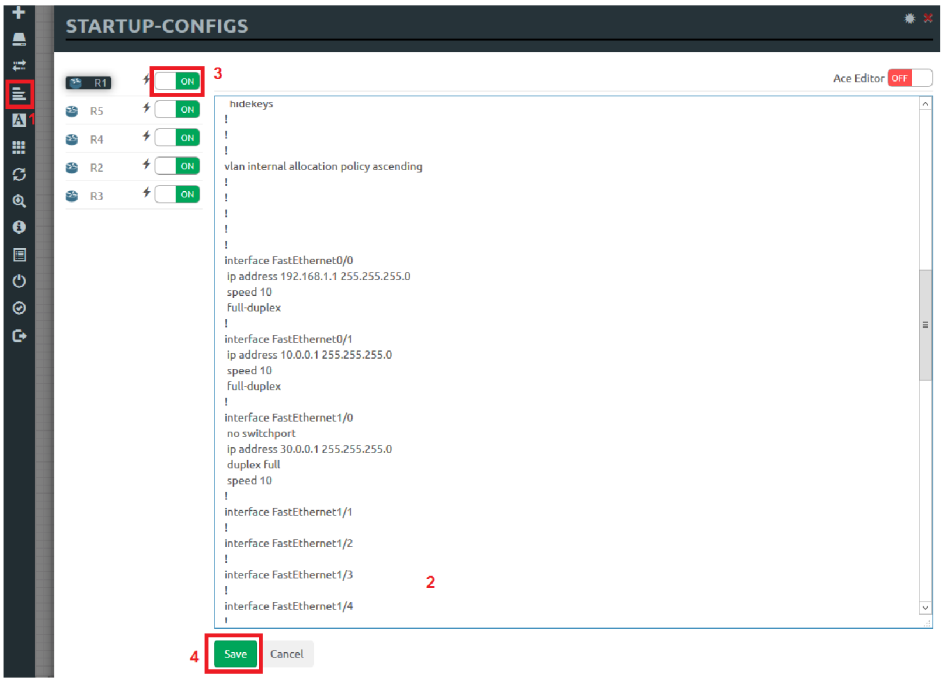
Po nainstalování programu Iperf obě zařízení vypněte a utvořte ze všech zařízeních síť jako je na obrázku [topologie], dbejte i na stejné připojení rozhraní mezi směrovači. Pojmenování zařízení provedete kliknutím pravým tlačítkem myši na zařízení a zvolení možnosti **Edit**, v nově otevřeném okně pojmenujte zařízení podle předlohy.



Obr č.6: Topologie

3. Nahrání přiložených konfigurací do směrovačů skrze systém EVE-NG:

Nahrání konfigurací do směrovačů provedete kliknutím v levé liště na možnost **Startup-configs** a vybráním směrovače, do kterého má být konfigurace nahrána. Dále překopírujte příslušnou konfiguraci z přílohy do okna a přepněte směrovač do stavu **ON**, poté konfiguraci uložte. Takto nahrajte konfigurace do všech směrovačů, každý směrovač má v příloze vždy svou před chystanou konfiguraci pro každý scénář.

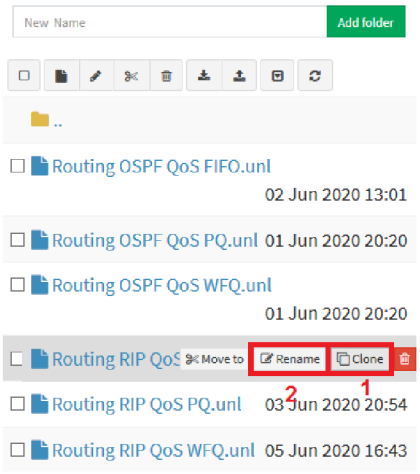


Obr č.7: Vložení konfigurace

Po nahrání všech konfigurací jděte do pole laboratoře a označte všechny směrovače, klikněte pravým tlačítkem myši na libovolný směrovač a zvolte **Wipe**. Po dokončení promazání směrovačů se nově konfigurace automaticky nahrají do směrovačů po jejich spuštění. Vypněte všechna zařízení a ukončete laboratoř.

4. **Vytvoření kopií laboratoře pro tvorbu scénářů:**

Po návratu do seznamu laboratoří zkopírujte první laboratoř a popište ji podobně jako první laboratoř, jen s rozdílem zkratek směrovacího protokolu nebo QoS mechaniky. Příklad je uveden na obrázku [obr seznam labin]. Funkcí kopírování laboratoří si připravte i všechny zbylé scénáře stejně jako na obrázku. Celkový počet scénářů-laboratoří je 6.



Obr č.8: Seznam scénářů

V každém scénáři je nutno nainstalovat program Iperf na linuxová zařízení stejném způsobem, jak již bylo popsáno v předešlých krocích, taktéž je nutné nahrání konfigurací do všech směrovačů pro každý scénář. To proveďte podobně jako ve třetím kroku, dbejte na výběr správné konfigurace pro správný směrovač ve správném scénáři.

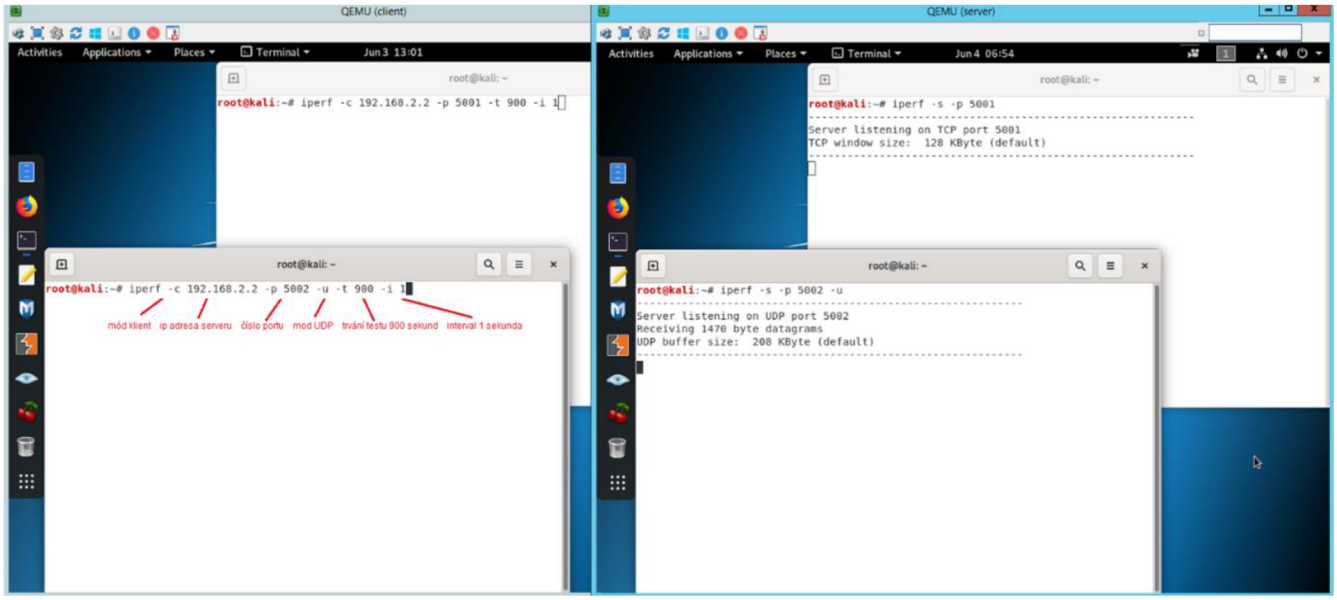
5. Měření laboratorní úlohy – studenti

Zadání:

Otestujte chování protokolů RIP a OSPF s mechanismy QoS FIFO, PQ a WFQ. Postupně proměřte 6 scénářů s různými konfiguracemi frontových mechanismů a směrovacích protokolů.

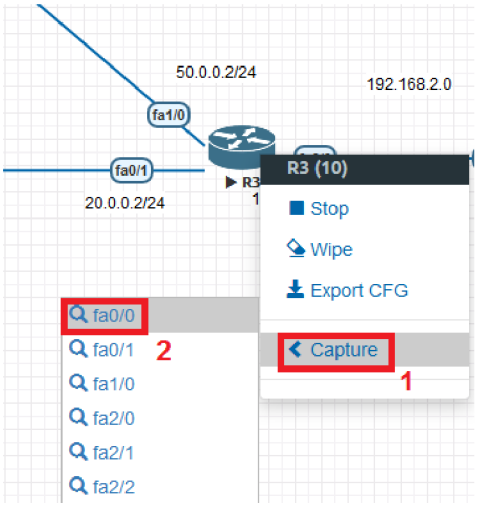
Průběh laboratorní úlohy:

Začněte otevřením scénářem **Routing RIP QoS FIFO**. V připravené laboratoři nejprve zapněte všechna zařízení. To provedete označením všech zařízení, kliknutím pravým tlačítkem myši na libovolné zařízení a vybráním možnosti **Start Select**. Dále se přihlaste na obě Linuxová zařízení, otevřete dva terminály na straně klienta i serveru a připravte si stranu serveru tak jako na obrázku [obr iperf připrava], můžete si předpřipravit příkazy pro testy TCP a UDP i na straně klienta.

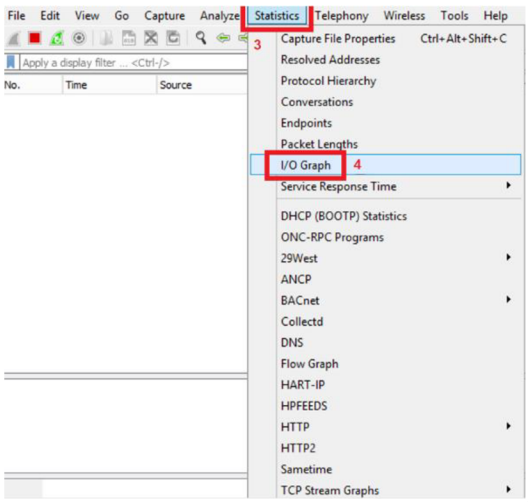


Obr č.9: Klient a server

Před samotným spuštěním testů za pomoci programu Iperf si ještě připravte program Wireshark pro záznam dat na směrovači R3, která budou vykreslena do grafů. Začněte kliknutím pravým tlačítkem myši na směrovač R3 a vyberte **Capture** na rozhraní **fa0/0**. Po otevření programu Wireshark spusťte grafické vykreslování přijatých dat podle obrázku.

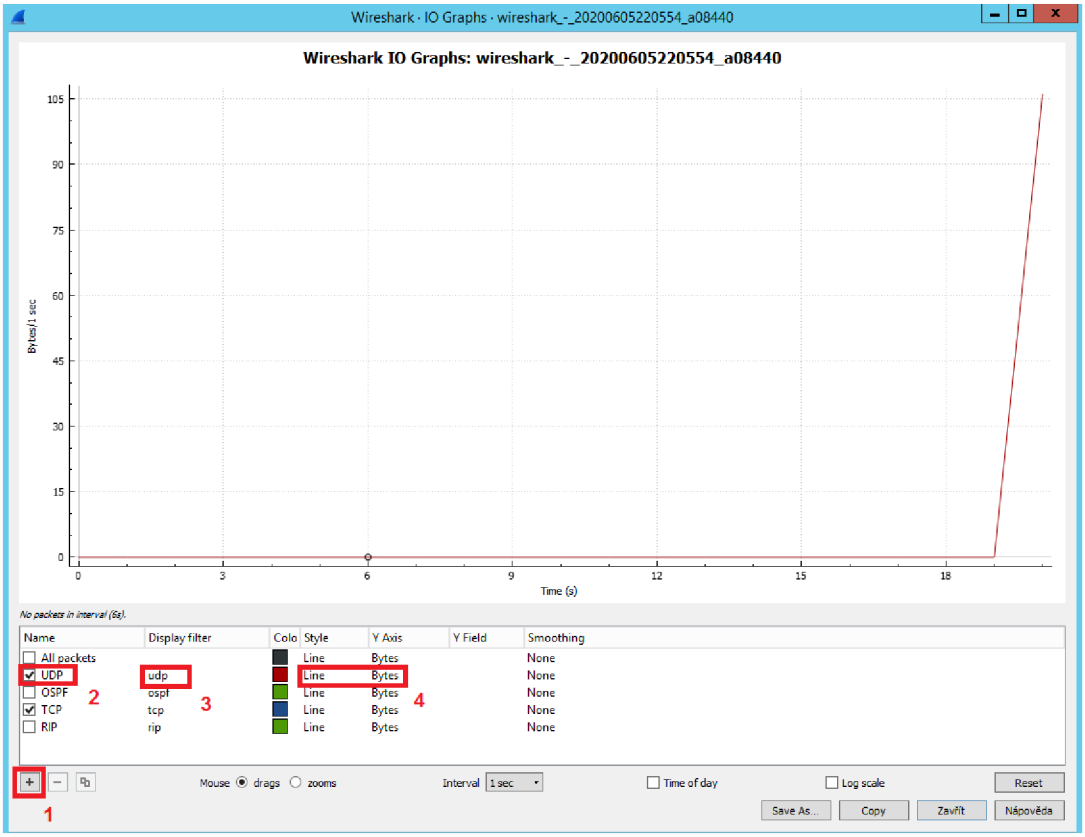


Obr č.10: Spuštěné Wiresharku



Obr č.11: Wireshark

Po otevření okna zaškrtněte zobrazení TCP a UDP, pokud v seznamu tato možnost není, přidejte ji následováním kroků na obrázku. Taktéž se ujistěte, že v seznamu jsou přítomny všechny možnosti vykreslení typů dat tak jako na obrázku a případně chybějící možnosti doplňte.



Obr č.12: I/O Graf

Nyní spusťte oba testy programu Iperf na klientovi. Řiďte se podle časové osy vykreslujícího grafu ve Wiresharku a zhruba v **120** sekundě vypněte směrovač **R2**, tím simulujte výpadek hlavní linky a sledujte průběh grafu. Po obnovení UDP a TCP spojení skrze náhradní linku (směrovače R4 a R5) v čase **700** sekund opět zapněte směrovač **R2**, čímž obnovíte hlavní linku, a počkejte na dokončení testu.

Dalším krokem si funkcí Print Screen uložte společný průběh TCP a UDP provozu, dále graf s průběhem všech paketů (All packets) a nakonec i průběh směrovacích dat RIP nebo OSPF v závislosti na testovaném scénáři. Dále si uložte výsledky z klienta a serveru, které poté zpracujte do tabulky v protokolu.

Po uložení všech potřebných dat vypněte všechna zařízení v síti podobným způsobem jako při jejich zapnutí, zavřete program Wireshark a vraťte se do seznamu laboratoří kliknutím na volbu **Close lab**

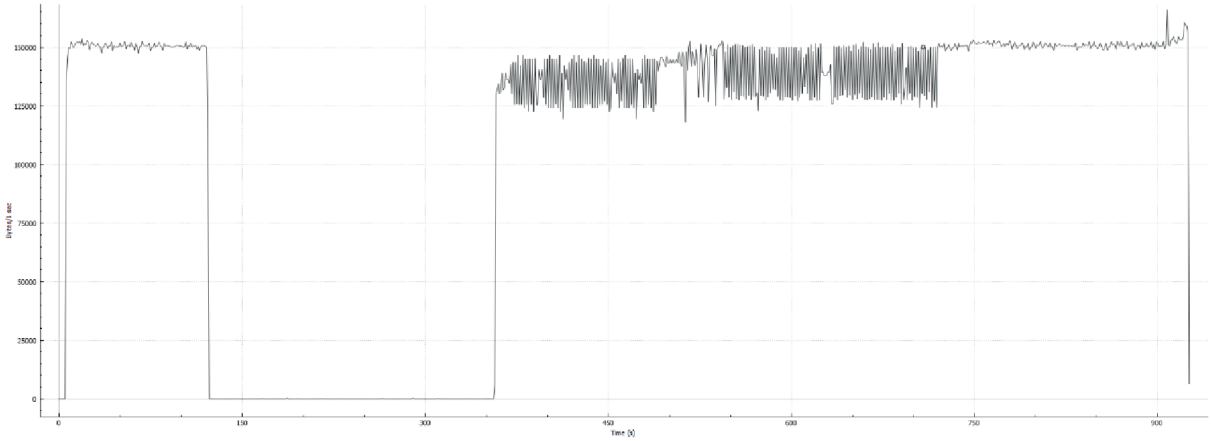
v levé svislé liště. Otevřete další laboratoř s názvem **Routing RIP QoS PQ** a opakujte měření, dále zopakujte měření i pro zbytek scénářů. Ve scénářích Routing OSPF QoS PQ a Routing OSPF QoS WFQ, lze zkrátit dobu testu na klientovi změnou indexu **– t** na hodnotu **600**. Ve zprávě o měření poté okomentujte všechny grafy a výsledky z programu Iperf vyneste do přehledné tabulky.

FEKT	Předmět	
	Jméno	
	Ročník	Studijní skupina
	Spolupracoval	Měřeno dne
Kontroloval	Hodnocení	Dne
Číslo úlohy	Název úlohy **Testování front QoS se směrovacími protokoly v emulátoru EVE-NG**	

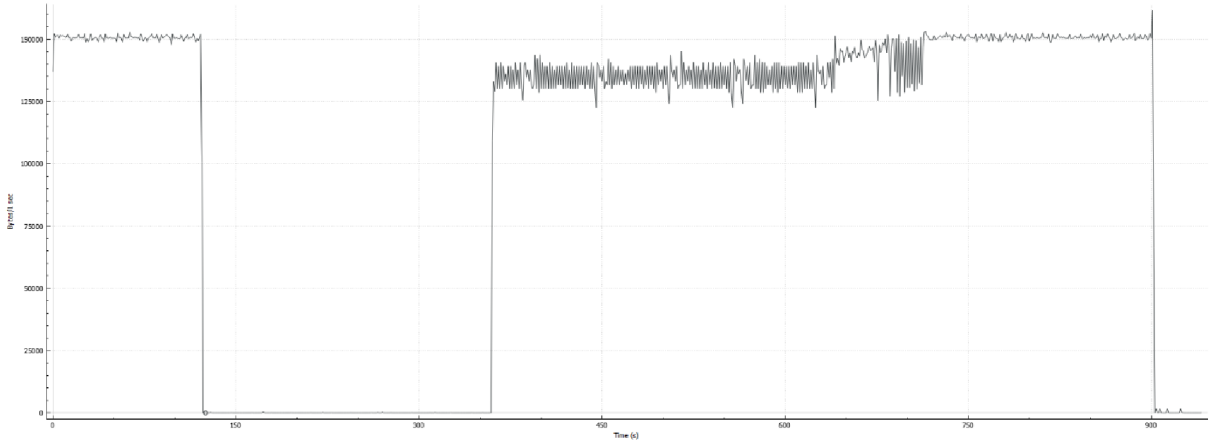
Zadání:

Otestujte chování protokolů RIP a OSPF s mechanismy QoS FIFO, PQ a WFQ. Postupně proměřte 6 scénářů s různými konfiguracemi frontových mechanismů a směrovacích protokolů.

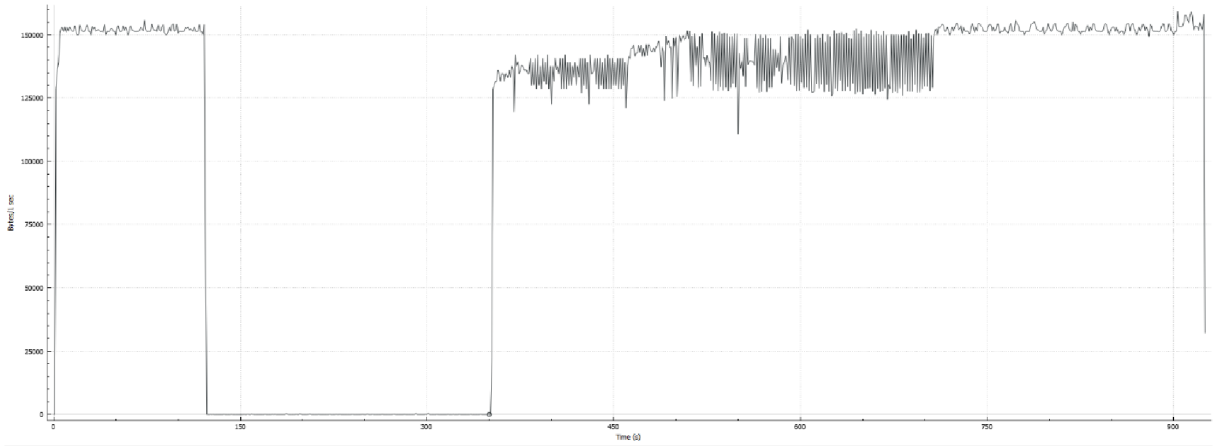
Naměřené hodnoty:



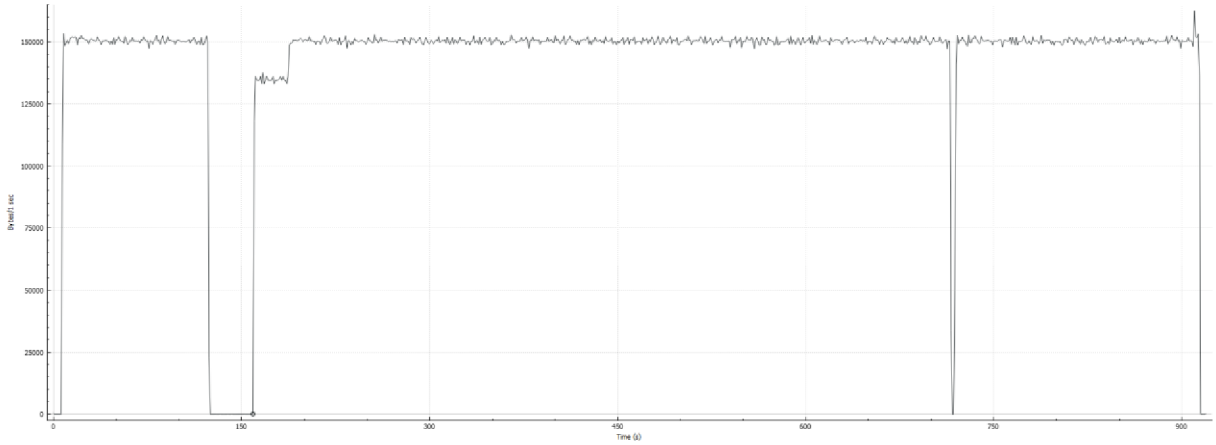
Graf. č. 1: Celkový příjem dat RIP FIFO



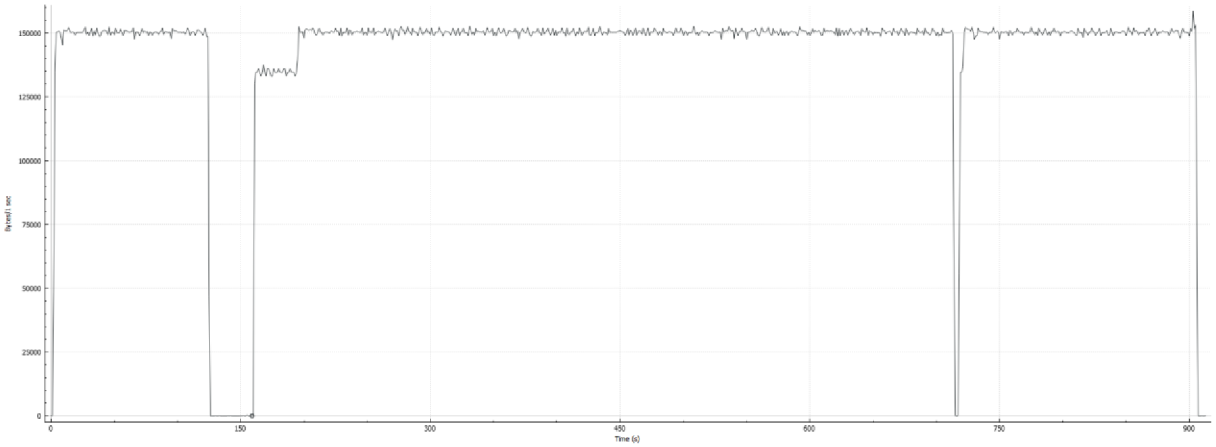
Graf. č. 2: Celkový příjem dat RIP PQ



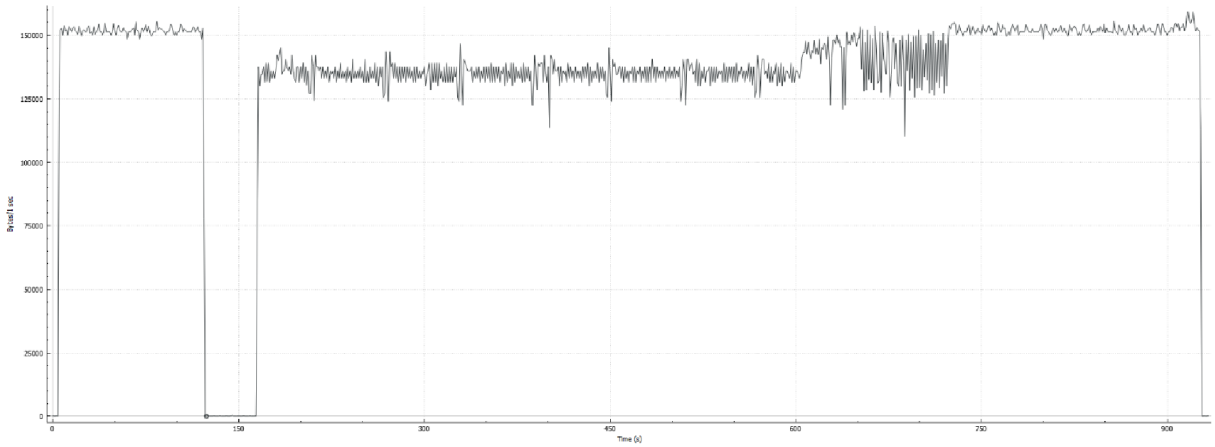
Graf. č. 3: Celkový příjem dat RIP WFQ



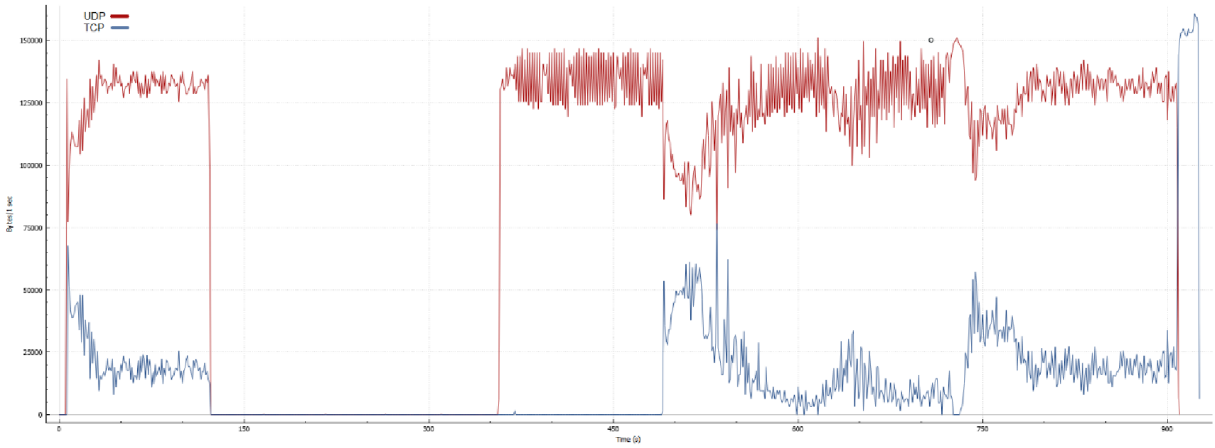
Graf. č. 4: Celkový příjem dat OSPF FIFO



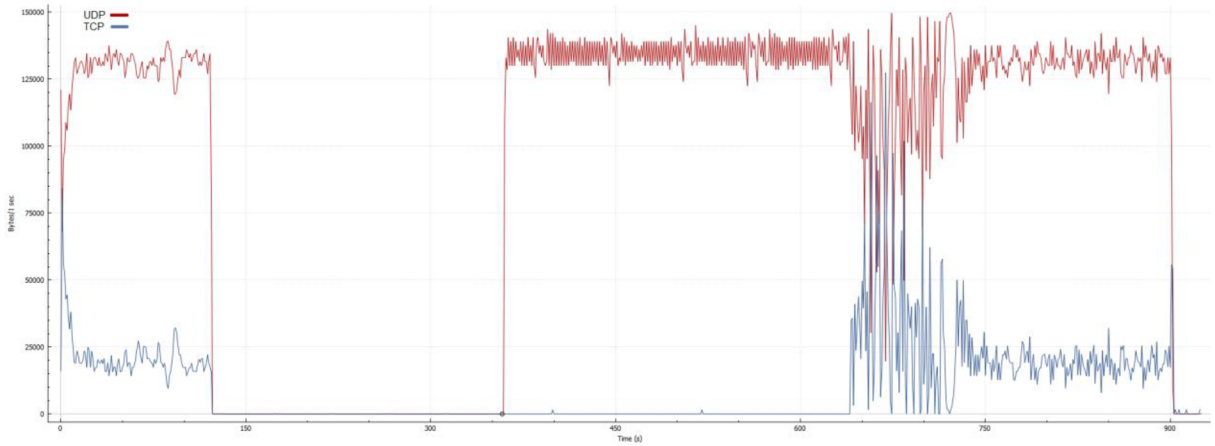
Graf. č. 5: Celkový příjem dat OSPF PQ



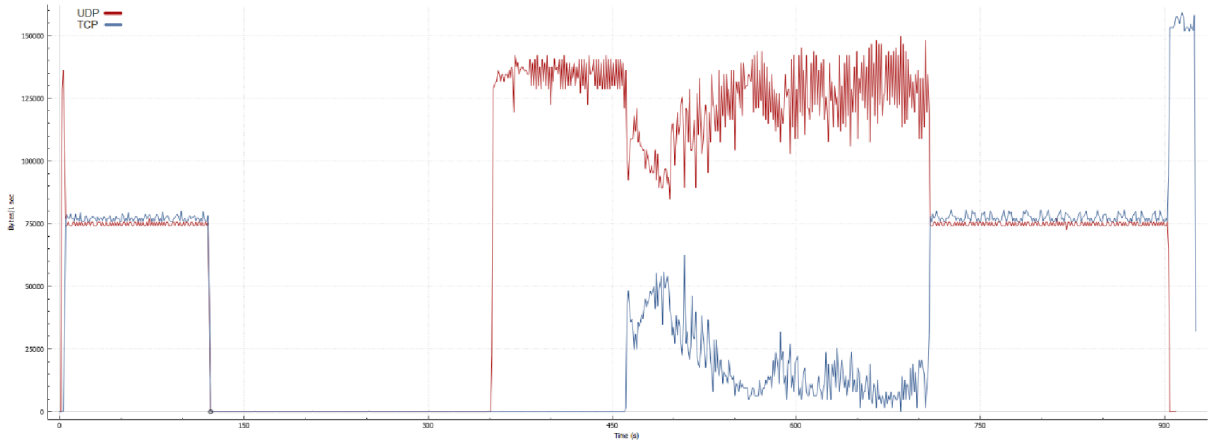
Graf. č. 6: Celkový příjem dat OSPF WFQ



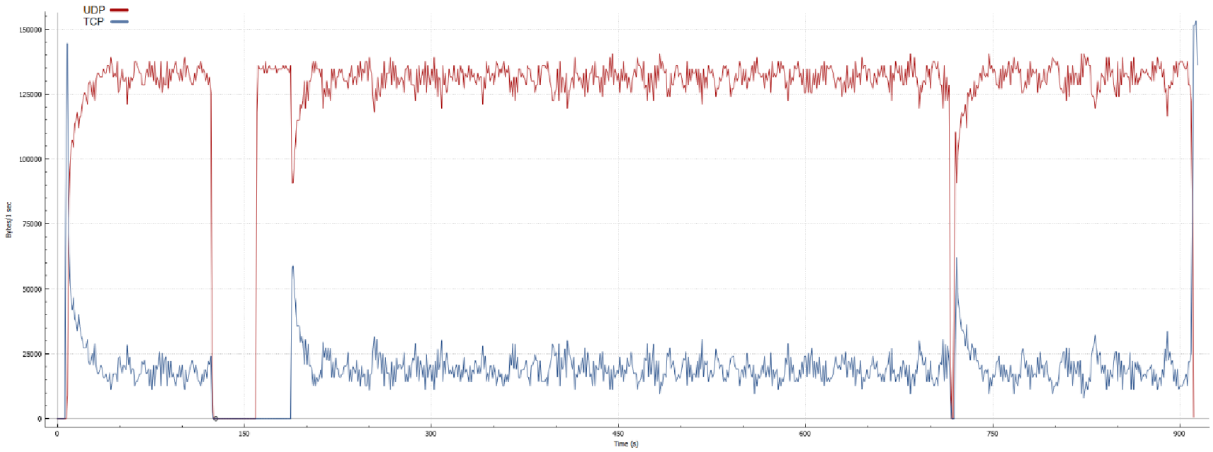
Graf. č. 7: Příjem UDP/TCP dat RIP FIFO



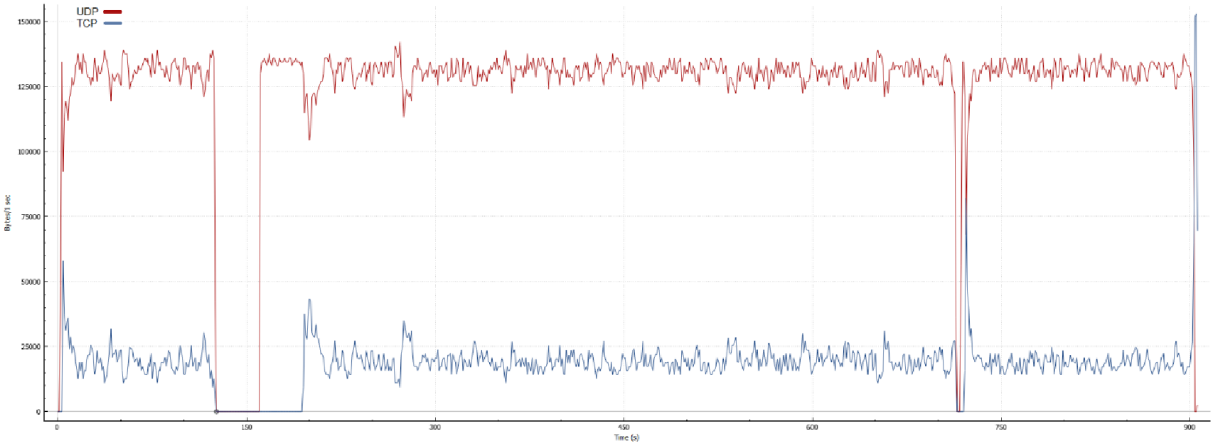
Graf. č. 8: Příjem UDP/TCP dat RIP PQ



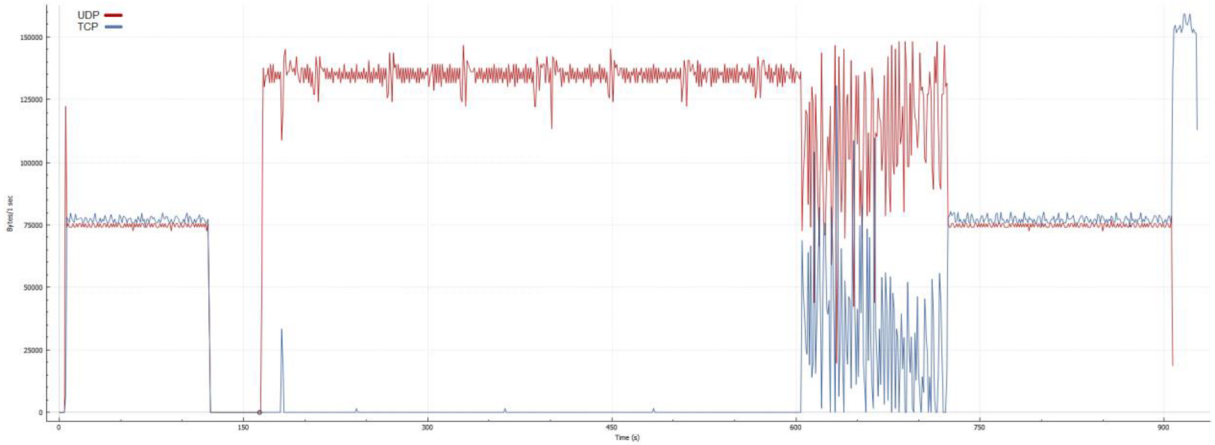
Graf. č. 9: Příjem UDP/TCP dat RIP WFQ



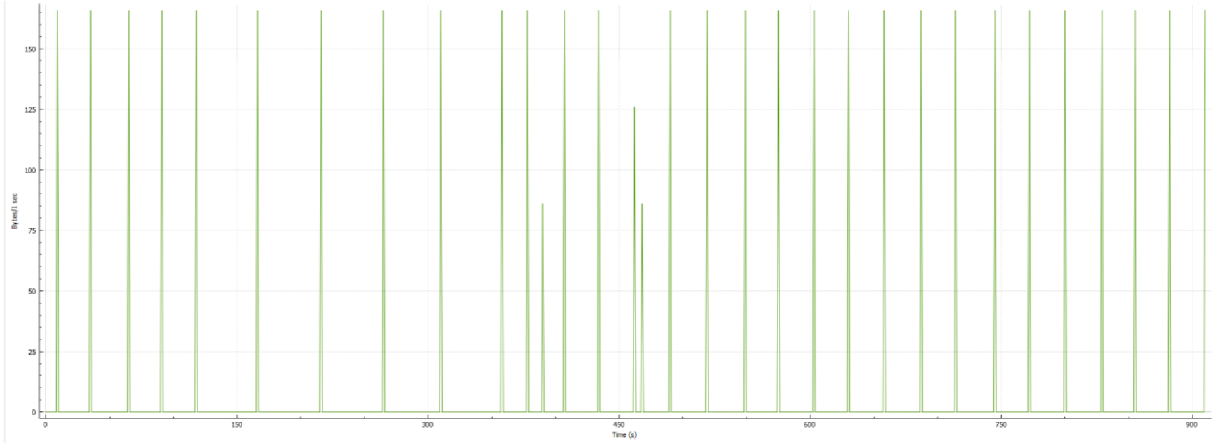
Graf. č. 10: Příjem UDP/TCP dat OSFP FIFO



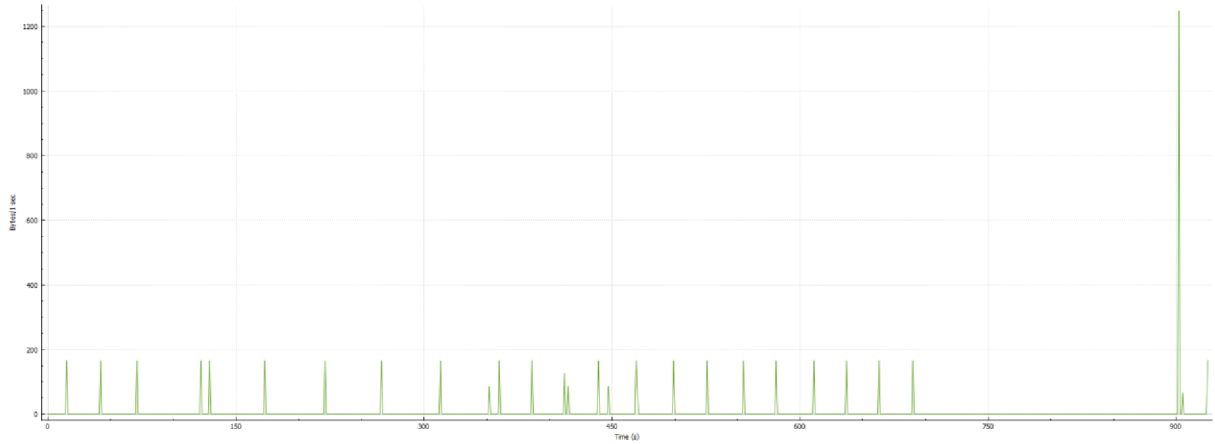
Graf. č. 11: Příjem UDP/TCP dat OSFP PQ



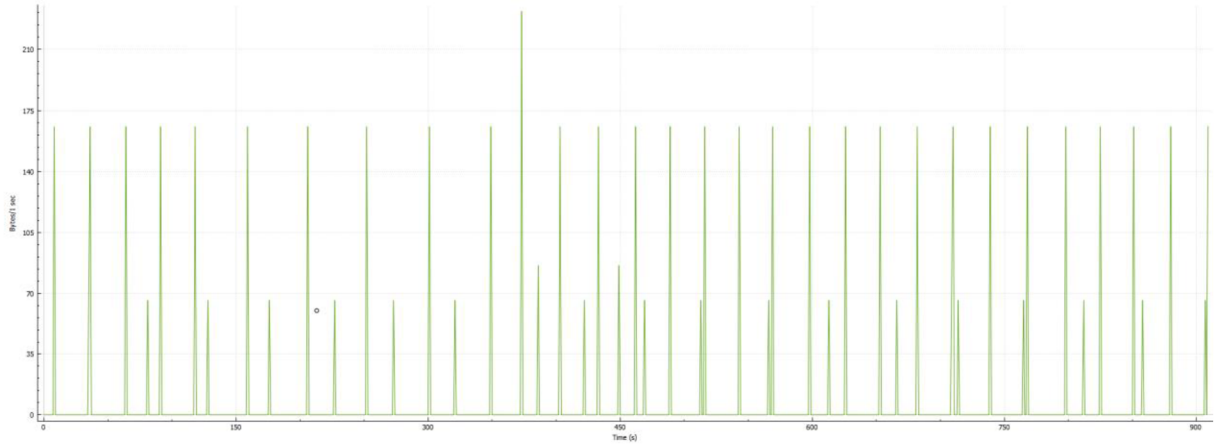
Graf. č. 12: Příjem UDP/TCP dat OSFP WFQ



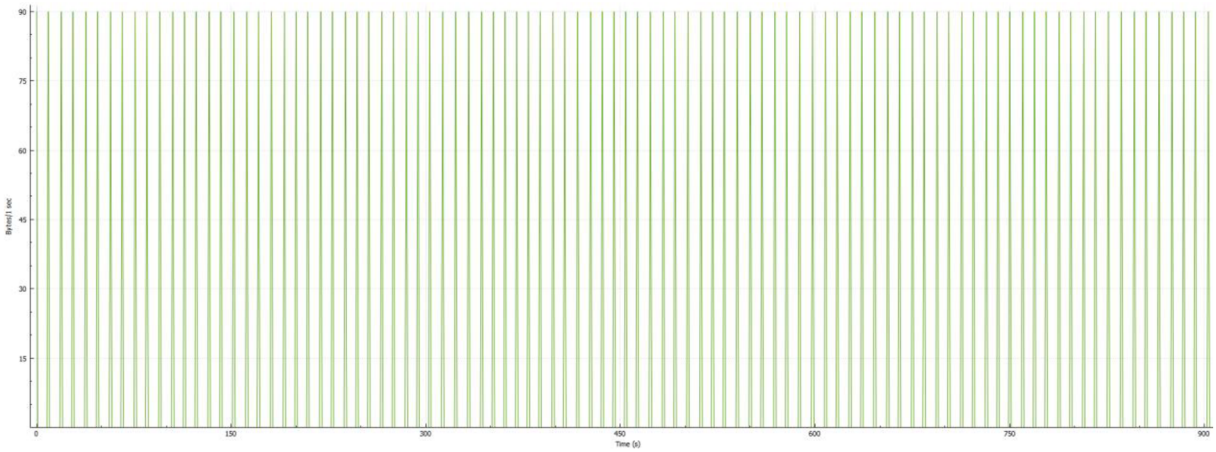
Graf. č. 13: Příjem směrovacích dat RIP FIFO



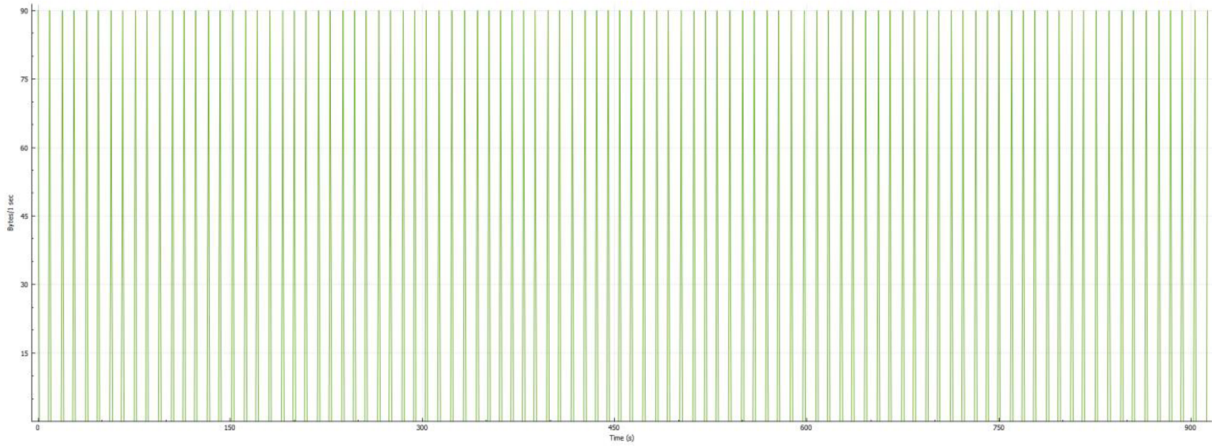
Graf. č. 14: Graf. č. 13: Příjem směrovacích dat RIP PQ



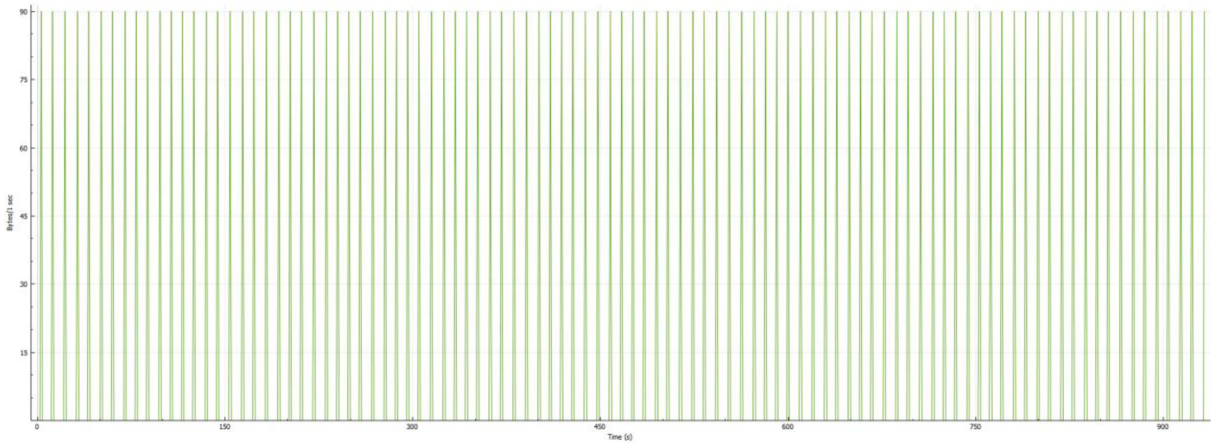
Graf. č. 15: Příjem směrovacích dat RIP WFQ



Graf. č. 16: Příjem směrovacích dat OSPF FIFO



Graf. č. 17: Příjem směrovacích dat OSPF PQ



Graf. č. 18: Příjem směrovacích dat OSPF WFQ

Tab. č. 1: Výsledky z programu Iperf

		TCP		UDP			
		Transfer [MBytes]	Bandwidth [Kbits/sec]	Transfer [MBytes]	Bandwidth [Kbits/sec]	Jitter [ms]	Lost/Total [%]
OSPF	FIFO	15,2	141	105	973	2,285	7,1
	PQ	14,5	135	104	973	2,489	7,1
	WFQ	26,7	243	87,8	817	5,610	22
RIP	FIFO	12,0	34,5	79,2	736	1,907	30
	PQ	10,8	85,0	79,9	743	1,990	29
	WFQ	25,2	257	63,4	590	9,131	44

Závěr:

Prodleva přepnutí na záložní linku u protokolu RIP trvala zhruba 240 sekund, důvodem tak velké prodlevy je rozesílání aktualizovaných tabulek každých 30 vteřin. Prodleva u OSPF protokolu trvala 35 sekund u scénářů FIFO a PQ, u scénáře WFQ pak 38 sekund. Je tedy patrné, že protokol OSPF reaguje na změny v síti mnohem rychleji. Z první šestice grafů je též patrná větší stabilita datového toku při použití směrovacího protokolu OSPF, a to nejspíše kvůli tomu, že tento protokol nezatěžuje tak výrazně síť jako protokol RIP, který každých 30 sekund rozesílá kompletní směrovací tabulky.

Další šestice grafů znázorňující průběhy UDP/TCP dat všech scénářů taktéž potvrzují rychlejší přepnutí datových toků na vedlejší linku a stabilnější tok dat u scénářů s OSPF protokolem. V těchto grafech je však patrné pomalejší obnovení TCP, nejspíše kvůli jeho způsobu fungování. To je nejvíce vidět ve scénářích OSPF WFQ a RIP FIFO. Grafy scénářů s WFQ nejprve vykazují obvyklé fungování mechaniky QoS, ovšem po výpadku se mechanika WFQ ve scénáři s protokolem RIP začala chovat podobně jako FIFO fronta se stejným typem protokolu. Ve scénáři s WFQ mechanikou a OSPF protokolem byl na dlouhou dobu upřednostněn UDP přenos podobně jako u scénáře RIP protokolu a mechaniky PQ. Z těchto důvodů by bylo vhodné dále prozkoumat mechaniku WFQ. Grafy ovšem celkově potvrzují teorii o všech mechanikách QoS i směrovacích protokolech.

Poslední šestice grafů se zabývá příjmem směrovacích dat, z těchto grafů vyplívá větší náročnost a náchylnost protokolu RIP na negativní vlivy QoS. Ve scénáři RIP PQ je možné vidět úsek, kdy směrovač přestal přijímat směrovací data, což by mohlo mít za následek omezení fungování protokolu RIP. Směrovací protokol OSPF je ve všech scénářích prakticky stejný a lze tedy usoudit, že je odolnější vůči nežádoucím projevům QoS.

Výsledky z programu Iperf byly vyznačeny do tabulky. Z těchto hodnot je patrné, že scénáře s mechanikou WFQ řídily provoz spravedlivěji než ostatní mechaniky, tuto domněnku podporuje i vysoká ztráta UDP dat. Ve scénářích s RIP protokolem došlo celkově ke větším ztrátám paketů, způsobené velkou opožděnou reakcí směrovacího protokolu. Mezi mechanikami FIFO a PQ není podle tabulky moc velký rozdíl, nejlepší kombinací směrovacího protokolu a QoS mechaniky je nejspíše OSPF s WFQ. Ovšem vždy záleží na konkrétním případu přenosu informací a podmínek, například kdybychom chtěli přenést nejvíce UDP dat, jeví se podle tabulky kombinace OSPF a FIFO jako lepší varianta.