

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SIMULACE TRIPLE PLAY SLUŽEB V SÍTÍCH XG-PON V PROSTŘEDÍ
NS-3

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PATRIK CYMOREK

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SIMULACE TRIPLE PLAY SLUŽEB V SÍTÍCH XG-PON V PROSTŘEDÍ NS-3

SIMULATION OF TRIPLE PLAY SERVICES IN XG-PON NETWORKS IN NS-3

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PATRIK CYMOREK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. TOMÁŠ HORVÁTH

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Patrik Cymorek

ID: 154689

Ročník: 3

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Simulace Triple Play služeb v sítích XG-PON v prostředí NS-3

POKYNY PRO VYPRACOVÁNÍ:

V bakalářské práci se bude nutně seznámit se standardem XG-PON pro pasivní optické sítě a principem přenosu dat pro Triple Play služby. Dále v teoretické části bude popsán princip přenosu dat v sítích BitTorrent a multicastový přenos k vybraným koncovým stanicím.

V praktické části budou navrženy simulační modely pro služby Triple Play a přenos dat pomocí BitTorrent protokolu. Další simulační scénář bude tvořit simulace multicastového přenosu IPTV služby v síti XG-PON. Veškeré simulace budou realizovány v simulačním prostředí NS-3, které nativně nepodporuje BitTorrent protokol. V rámci zhodnocení výsledků provede student jejich diskuzi.

DOPORUČENÁ LITERATURA:

[1] JANEVSKI, Toni. NGN architectures, protocols, and services. pages cm. ISBN 978-111-8607-206.

[2] HENS, Francisco J a Jose

Termín zadání: 9.2.2015

Termín odevzdání: 2.6.2015

Vedoucí práce: Ing. Tomáš Horváth

Konzultanti bakalářské práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být 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.

ABSTRAKT

Cílem této práce bylo seznámit se se službami Triple Play a možnostmi simulace těchto služeb v prostředí NS-3. První kapitoly dokumentu se zabývají popisem optických sítí. V následující kapitole jsou rozebrány samotné služby Triple Play, jsou popisovány její součásti včetně jejich výhod a nevýhod. V závěru této kapitoly je popisován očekávaný vývoj služby Triple Play do budoucna. V závěru teoretické části je krátce obsažen popis zajištění kvality služeb a popis BitTorrent protokolu.

V praktické části je v prostředí NS-3 navržena topologie pro simulaci služeb Triple Play v sítích XG-PON. Na navržené topologii jsou simulovány jednotlivé Triple Play služby. Výsledky simulace jsou prezentovány graficky s pomocí souborů zaznamenávajících průběh simulace. V závěru praktické části jsou výsledky dosažené simulací srovnány s reálným chováním v síti.

KLÍČOVÁ SLOVA

Triple Play, IPTV, VoIP, pasivní optická síť, NS-3

ABSTRACT

The main goal of this paper was to familiarize ourselves with Triple Play service and to determine the possibility of simulating this service in NS-3. The first chapter of this document describes optical networks in detail. In the next chapter, we describe and analyze Triple Play service, including consideration of its advantages and disadvantages. Future development of the service is described at the end of this chapter. A short description of the Quality of Service and the BitTorrent protocol is included at the end of the theoretical section of the paper.

The topology that we used for the simulation of Triple Play service in XG-PON networks is presented in the practical part of this paper. We simulated Triple Play services in the topology that we designed. The results of the simulation are presented graphically with the help of files that were recorded during the simulation. Comparison of the simulated results with real behaviour in network is included at the end of the practical section of the paper.

KEYWORDS

Triple Play, IPTV, VoIP, passive optical network, NS-3

CYMOREK, Patrik *Simulace Triple Play služeb v sítích XG-PON v prostředí NS-3*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 62 s. Vedoucí práce byl Ing. Tomáš Horváth.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Simulace Triple Play služeb v sítích XG-PON v prostředí NS-3“ jsem vypracoval samostatně pod vedením vedoucího 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 bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Tomáši Horváthovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

Výzkum popsany v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....

(podpis autora)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Úvod	11
1 Optické sítě	12
2 Pasivní sítě PON	14
2.1 Skladba PON	14
2.2 Splitter	15
2.3 Provoz v pasivní optické síti	16
2.4 Vývoj pasivních optických sítí	17
2.4.1 APON/BPON	17
2.4.2 GPON	18
2.4.3 EPON	18
2.4.4 10G-EPON	18
2.4.5 NG-PON	19
3 Optické sítě FTTx	20
4 Triple Play	22
4.1 Problematika Triple Play	22
4.2 VoIP	22
4.2.1 Přenos signálu	23
4.2.2 Parametry sítě	24
4.3 IPTV	26
4.3.1 Rozdíl IPTV a internetového videa	26
4.3.2 Multicastový přenos IPTV	26
4.3.3 Video on Demand	27
4.3.4 Pay per View	28
4.4 Quadruple Play	28
5 QoS (Quality of Services)	29
5.1 IntServ	29
5.2 DiffServ	29
6 BitTorrent protokol	32
7 Implementace do NS-3	35
7.1 Použitý software	35
7.2 Práce v NS-3	36

7.3	Simulace	37
7.3.1	Webové stránky	38
7.3.2	PPV	40
7.3.3	VoD	41
7.3.4	BitTorrent	44
7.3.5	IPTV	47
7.3.6	Srovnání zobrazených výsledků	48
8	Závěr	54
	Literatura	56
	Seznam symbolů, veličin a zkratk	58
	Seznam příloh	61
A	Obsah přiloženého CD	62

SEZNAM OBRÁZKŮ

1.1	Rozdíl mezi AON a PON [14].	12
2.1	Skladba PON [2].	14
2.2	Struktura PLC a FBT rozbočovačů [2].	15
2.3	Princip downstreamu a upstreamu [4].	16
2.4	Multiplex 1G a 10G-PON [14].	19
3.1	Variety přípojek FTTx [17].	20
4.1	Digitalizace signálu [16].	23
4.2	Zdroje zpoždění [4].	25
6.1	Princip sdílení souborů pomocí BitTorrent protokolu [6].	34
7.1	Topologie simulace.	37
7.2	Komunikace mezi uzly Client_1 a WebServer.	39
7.3	Wireshark – záznam komunikace na WebServer.	39
7.4	Wireshark – záznam PPV komunikace.	40
7.5	PPV stream.	41
7.6	Throughput PPV pomocí Wiresharku.	41
7.7	Wireshark – záznam VoD komunikace 1.	42
7.8	Wireshark – záznam VoD komunikace 2.	42
7.9	VoD stream.	43
7.10	Throughput VoD pomocí Wiresharku.	43
7.11	Wireshark – záznam BitTorrent komunikace.	45
7.12	Aktualizace na tracker.	45
7.13	Wireshark – ukončení downloadu ze strany uzlu Source_B.	46
7.14	Vytížení linky mezi směrovačem R a OLT.	46
7.15	Počátek vysílání na uzel Client_3.	47
7.16	Změna multicastové skupiny pomocí NetAnimu.	48
7.17	Wireshark – změna multicastové skupiny, zdroj: onu_3-pcap-3-1.pcap.	48
7.18	Dekódování RTP protokolu.	49
7.19	VoD – odesílání dat správným směrem.	49
7.20	Throughput VoD bez použití logaritmické osy.	50
7.21	Zahlcení při VoD.	50
7.22	Obecný graf přenosové rychlosti udávaný bittorrent klientem.	51
7.23	Seznam připojených klientů udávaný bittorrent klientem.	51

SEZNAM TABULEK

2.1	Útlum splitteru v závislosti na dělicím poměru [2].	15
4.1	Požadavky SDTV a HDTV na přenosovou rychlost po kompresi [4]. .	27
5.1	DSCP pole kategorie Assured Forwarding [4].	30
5.2	Stupně priorit kategorie Class Selector [4].	31

ÚVOD

Již od poloviny 70. let minulého století je optické vlákno používáno jako fyzické médium k přenosům, a to nejen datovým. Od těchto dob byla optická vlákna zdokonalována, až se stala součástí vrcholu technologie. Díky skvělým vlastnostem tohoto média, jako je velká šířka pásma, nízký útlum, nízká hmotnost a rozměry, bezpečnost proti odposlechu a stále se zvyšujícím nárokům na přenosovou rychlost začala být optická vlákna nasazována (mimo jiná použití jako například optické senzory) jako médium v páteřních a metropolitních datových sítích. Se stále zvyšující se kvalitou a snižující se cenou jsou optická vlákna nasazována do přístupových sítí a vytlačují tak již překonané metalické vedení. V současné době jsou koncoví uživatelé připojováni optickými vlákny doslova až do domu.

Díky tomuto pokroku může být koncovým uživatelům poskytováno stále větší množství služeb s vysokou kvalitou. Tato práce je zaměřena na popis Triple Play služeb. Jak název napovídá, tato služba zahrnuje poskytování 3 služeb, a to přenosu dat, hlasu a videa skrze jedno připojení. Každá z těchto služeb má jiné požadavky pro zajištění odpovídající kvality služby. Pro zaručení kvality se využívá politiky QoS.

V prvních kapitolách této práce je uveden popis přehledu optických sítí, jejich výhody, princip a vývoj. Jsou také nastíněny varianty připojení koncových uživatelů FTTx. V následující kapitole je vysvětlen pojem Triple Play a jsou popsány služby, které zahrnuje. V kapitolách o VoIP a IPTV je uveden princip služeb, jejich výhody a nevýhody. Poslední část této kapitoly pojednává o výhledu do budoucnosti Triple Play služeb. Následující teoretická kapitola popisuje pojem kvalita služeb, proč se využívá a jak může být aplikována. V poslední kapitole je popsán BitTorrent protokol včetně průběhu a principu sdílení souborů.

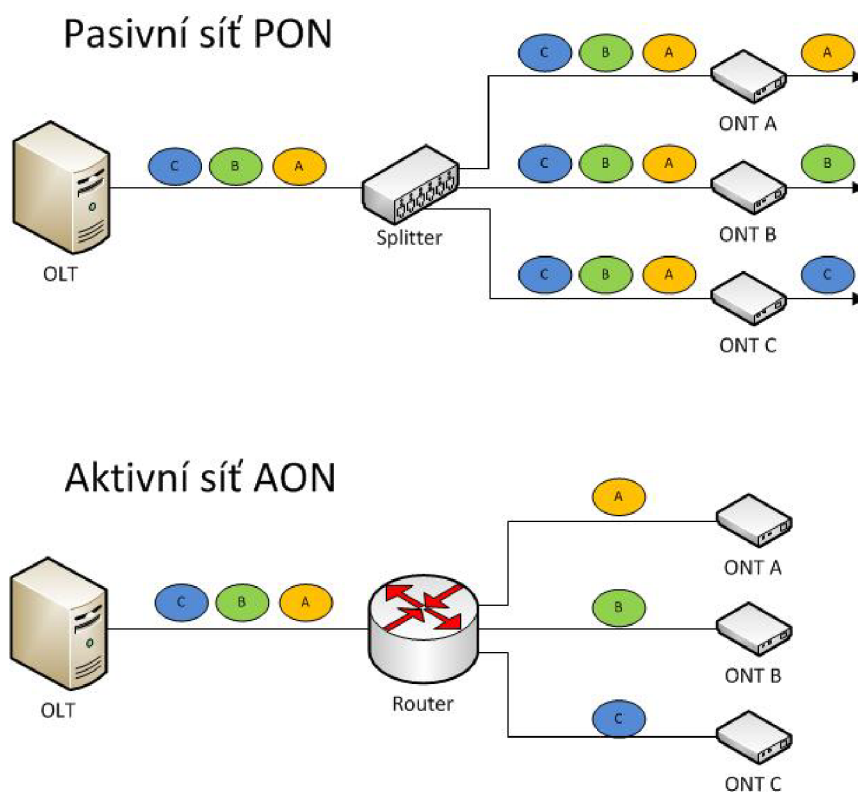
Následující kapitola práce se věnuje praktické implementaci Triple Play služeb v prostředí NS-3. Úvodem je zmíněn použitý software pro vytvoření simulace a základní objekty, se kterými je ve zdrojovém kódu pracováno. V následující podkapitole je popsána vytvořená topologie sloužící pro simulování služeb. V každé části podkapitoly je v krátkosti popsána služba, která je simulována, průběh simulace a specifické vlastnosti každé ze služeb, spolu s grafickými výstupy získanými pomocí použitého softwaru. Na závěr jsou získané výsledky diskutovány ve srovnání s reálným chováním sítě.

1 OPTICKÉ SÍTĚ

Díky neustálému rozšiřování, pokroku a nabídce služeb v oblasti telekomunikací dochází k nárůstu požadavků na přenosovou rychlost sítí, a to nejen páteřních či metropolitních ale také přístupových, které připojují koncové uživatele. Koncoví uživatelé v dnešní době požadují rychlosti v řádech desítek Mbit/s, organizace či větší komplexy dokonce v jednotkách Gbit/s. Tyto nároky již nejsou schopny splnit sítě metalické, proto dochází k zavádění sítí optických. Existují 2 základní typy optických sítí:

- Aktivní optické sítě – AON.
- Pasivní optické sítě – PON.

Mezi jejich hlavní rozdíly patří potřeba napájení routerů u sítí AON (u splitteru není potřeba) a množství informací vysílané na odchozí porty, kdy u splitteru je veškerý příchozí provoz odeslán na každé odchozí vlákno, na rozdíl od sítí AON, kdy jsou data směrována pouze na jeden výstup. Oba typy sítí je možno vidět na obr. 1.1.



Obr. 1.1: Rozdíl mezi AON a PON [14].

Sítě mohou být konstruovány dvěma různými strukturami:

- **P2P (Point-to-Point)** – Existuje přímé spojení mezi síťovým zařízením operátora OLT (Optical Line Terminal) a koncovým zařízením ONU (Optical Network Unit).
- **P2MP (Point-to-Multipoint)** – Mezi jednotkami OLT a ONU není přímé spojení jediným vláknem, ve struktuře jsou řazeny prvky (splittery), které rozdělují optický signál vstupního portu na porty výstupní.

Prvky tvořící přístupovou síť:

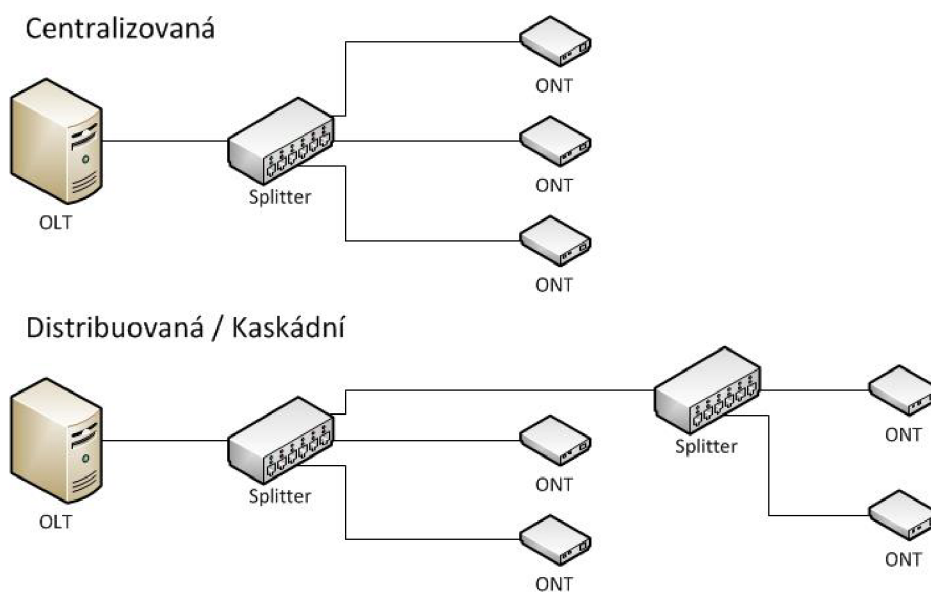
- **Optical line terminal (OLT)** – Optické linkové zařízení – představuje koncový bod (endpoint) ze strany poskytovatele, má funkci síťového rozhraní mezi přístupovou sítí a sítí poskytující telekomunikační služby. Poskytuje také možnost správy a dohledu nad sítí.
- **Optical network unit (ONU)** – Optická síťová jednotka – rozhraní mezi optickou (přístupovou) a metalickou (lokální) částí sítě, dochází v ní k transformaci příchozího optického signálu na signál elektrický. Za jednotku ONU je připojován větší počet účastníků.
- **Optical network terminal (ONT)** – Optické síťové zakončení – zařízení ukončující optickou linku, provádějící převod optického signálu na signál pro koncová zařízení (data, hlas, video). Principiálně je ONT stejné jako ONU, rozdíl v názvosloví je dán standardy.

2 PASIVNÍ SÍTĚ PON

Tato síťová infrastruktura je založena na použití pouze pasivních síťových prvků mezi OLT a ONU. Signál není během cesty obnoven, je pouze splitterem rozdělen na všechny jeho výstupní porty. Pro přístup k médiu je zavedeno časové dělení TDMA (Time Division Multiple Access). Z důvodů minimalizace nákladů na provoz ze strany operátora a co nejnižších výdajů za připojení do sítě ze strany uživatele jsou pasivní sítě mnohem více rozšířeny než AON [20] [21].

2.1 Skladba PON

Náhled na pasivní optickou síť, zobrazenou mimo jiné na obr. 2.1 je nutno začít u poskytovatele, z jehož centrály s OLT jsou vyvedena jednovláknová optická vlákna. S ohledem na geografické uspořádání síťových uživatelů jsou do cesty mezi poskytovatelem a koncové uživatele řazeny optické rozbočovače – splitters, které rozdělí příchozí signál v poměru 1:N v závislosti na počtu svých výstupních portů. Na výstupní porty splitterů jsou připojena nová optická vlákna, která vedou k dalším jednotkám. Samozřejmě je z hlediska efektivity možno řadit splitters kaskádově, jejich funkce zůstává stejná. Výhodnější je ovšem z hlediska útlumu a zásahů do celistvosti centralizovaná skladba, použití je ale závislé na mnoha dalších parametrech jako jsou: geografické rozmístění koncových uživatelů, maximální nutný dosah sítě aj.



Obr. 2.1: Skladba PON [2].

2.2 Splitter

Optický rozbočovač slouží v PON k rozdělení signálu pro více uživatelů ve směru sestupném, či sloučení příchozích signálů ve směru vzestupném. Jedná se o pasivní prvky, které mají pouze schopnost rozbočit nebo sloučit příchozí signál. Nejsou schopny tento signál jakkoliv upravovat.

Rozbočovače představují v rámci optické trasy největší útlum, proto byly stanoveny maximální mezní hodnoty útlumu v závislosti na dělicím poměru, které obsahuje tab. 2.1.

Tab. 2.1: Útlum splitteru v závislosti na dělicím poměru [2].

Dělicí poměr	1:4	1:8	1:16	1:32
Maximální hodnota vloženého útlumu [dB]	7,1	10,2	13,2	16,6

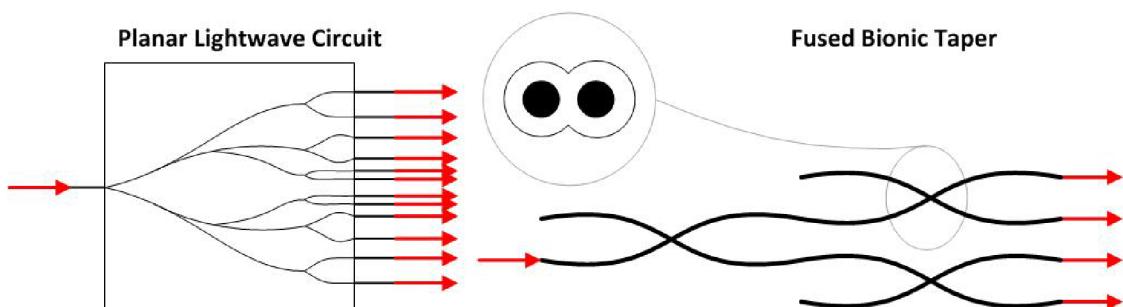
Dle výrobní technologie rozdělujeme rozbočovače do dvou skupin. Jejich struktura je zobrazeno na obr. 2.2 [2].

- **PLC (Planar Lightwave Circuit)**

PLC rozbočovače jsou vyráběny planárně, kdy je na křemíkovém substrátu vytvářena požadovaná struktura. Touto technologií jsou vytvářeny především rozbočovače s vysokým dělicím poměrem (až 1:32).

- **FBT (Fused Bionic Taper)**

FBT rozbočovače jsou vyráběny metodou slučování vláken [11]. Za vysokého tlaku a teploty se nataví pláště optických vláken, jádra vláken se téměř spojí. Rozbočovače vyrobené touto technologií mají menší počet výstupních portů.



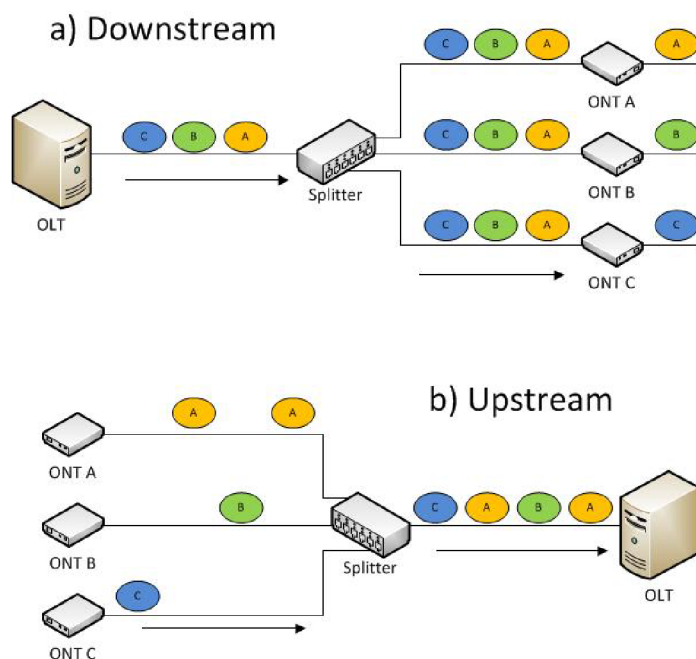
Obr. 2.2: Struktura PLC a FBT rozbočovačů [2].

2.3 Provoz v pasivní optické síti

Nejen v optické síti není žádoucí přenášet data pouze jedním směrem, je nutno komunikovat dvěma směry. Směr sestupný, kdy jsou data odesílána k uživateli se jmenuje downstream. Směr vzestupný, tedy od uživatelů do sítě se nazývá upstream. Pro zajištění duplexní komunikace po jednom vlákně se využívá vlnového multiplexování WDM (Wavelength Division Multiplexing).

V sestupném směru (obr. 2.3 (a)) jednotka OLT vysílá XGTC (XG-PON Transmission Convergence) rámce, které nesou informace pro uživatele. Jelikož během přenosu nejsou data nijak směrována, ale pouze rozbočena, musí být v záhlavích jednotlivých rámců LLID (Logical Link Identifier) identifikátory, označující pro kterou ONT jednotku jsou data určena (tuto informaci do rámce přidává OLT). Jednotky ONT na základě informací ze záhlaví vyčlení svá data, zbytek zahodí. V rámci bezpečnosti by mělo být zavedeno šifrování, z důvodu, že všechny informace z OLT jsou odesílány všem jednotkám ONT (ONU) v síti broadcastem. Zda bude šifrování zavedeno volí ISP (Internet Service Provider).

Vysílání rámců ve vzestupném směru (obr. 2.3 (b)) představuje značně větší problém než ve směru sestupném. Využívá se zde metody TDMA. Řízení přenosu připadá na jednotku OLT, která přiděluje jednotlivým ONT časové intervaly, ve kterých mohou vysílat. Pro zamezení kolizí, které mohou vzniknout kvůli různým vzdálenostem mezi OLT a ONT dochází k časové synchronizaci mezi těmito jednotkami [4].



Obr. 2.3: Princip downstreamu a upstreamu [4].

2.4 Vývoj pasivních optických sítí

Od dob, kdy se začaly PON sítě používat dochází ke stálému zdokonalování technologií. Ruku v ruce s pokrokem jde také vývoj protokolů a prvků, které sítě obsahují. Vývoj umožňuje stále zvyšování rychlostí připojení a nárůst počtu uživatelů v síti, se kterým je nutno také standardizovat různé pokrokové stupně. Standardy se liší zejména maximální přenosovou rychlostí, překlenutou vzdáleností a počtem připojených uživatelů.

2.4.1 APON/BPON

Historicky prvním standardem byl APON (ATM PON). Nese označení ITU–T G.983 a byl schválen v roce 1998 organizací ITU (International Telecommunication Union). Používaným protokolem byl ATM (Asynchronous Transfer Mode), který byl spojově orientován. Bylo tedy potřeba sestavit spojení mezi vysílací a přijímací stranou před započítáním samotného přenosu. Jedna síť připojila až 32 účastníků, kdy maximální překlenutou vzdáleností bylo 20 km.

Standard umožňoval použití symetrických nebo nesymetrických přenosových rychlostí. Symetrická varianta nabízela rychlost 155,52 Mbit/s v obou směrech. Nesymetrická varianta rychlosti:

- downstreamu (data k uživateli) – 622,08 Mbit/s,
- upstreamu (data od uživatele) – 155,52 Mbit/s.

Pro obousměrnou komunikaci pomocí dvou vláken bylo použito pásmo vlnových délek 1260–1360 nm s centrální vlnovou délkou 1310 nm [8].

BPON (Broadband PON) je nadstavbou APON. Standard byl upraven v roce 2001, nesl stejné označení a vycházel ze shodného protokolu.

Změnami, které přinesl bylo přidání rychlostní symetrické varianty 622,08 Mbit/s v obou směrech a nesymetrických variant (downstream/upstream):

- 1244,16 Mbit/s, 155,52 Mbit/s,
- 1244,16 Mbit/s, 622,08 Mbit/s.

Duplexní komunikace mohlo být dosaženo pomocí jednoho vlákna, používajícího vlnového multiplexu WDM. Upstream komunikace probíhala v pásmu 1260–1360 nm se střední vlnovou délkou 1310 nm. Pro downstream 155 Mbit/s a 622 Mbit/s bylo vyhrazeno pásmo 1480–1580 nm s necentrickou vlnovou délkou 1550 nm. Pro downstream 1,2 Gbit/s bylo vyhrazeno pásmo vlnových délek 1480–1500 nm [8].

2.4.2 GPON

Z důvodu nedostačující rychlosti byl následujícím ITU standardem Gigabit-capable passive optical networks (GPON). Jehož hlavním cílem je zavedení vyšších rychlostí než u předchozích standardů a zároveň navýšení dělícího poměru. V rámci GPON standardu existují dvě rychlostní kombinace:

- 2,4 Gbit/s downstream, 1,2 Gbit/s upstream,
- 2,4 Gbit/s downstream, 2,4 Gbit/s upstream.

Logická překlenutá vzdálenost byla stanovena na 60 km (sítě s rozšířeným dosahem), avšak maximální reálná fyzická odlehlost mezi OLT a ONU byla definována na 10 km nebo 20 km. Pro udržení rychlostí nad 1,25 Gbit/s je předpokládána maximální odlehlost 20 km. Maximální počet uživatelů v síti byl stanoven na 64, s pokrokem optických prvků se však očekávalo zvýšení na 128 uživatelů [9].

Novinkou v tomto standardu byla proměnná délka rámců díky zavedení metody GEM (GPON Encapsulation Mode). Pro zachování zpětné kompatibility bylo používáno také ATM [11].

2.4.3 EPON

Standard Ethernet Passive Optical Network je prvním, který nebyl navržen institucí ITU. O jeho zavedení se zasloužila instituce IEEE (Institute of Electrical and Electronics Engineers) [7].

Je podporováno zavedení dvou typů standardu EPON, tzv. short (krátkého dosahu) a long (dlouhého dosahu). Jmenovitá rychlost obou variant je 1,25 Gbit/s. Hlavním rozdílem je překlenutá vzdálenost, 10 km dosahu pro typ short a 20 km dosahu pro typ long (terminologie short a long byla převzata z [7]). Obě varianty podporují minimální vydělení 1:16, tedy připojení minimálně 16 účastníků. Duplexní komunikace je dosaženo pomocí vlnového multiplexu WDM po jednom vlákně. Použitá pásma vlnových délek:

- 1480–1500 nm pro downstream,
- 1260–1360 nm pro upstream.

2.4.4 10G-EPON

10G-EPON je druhým standardem instituce IEEE. Jedná se o rychlejší variantu EPON. Rychlost downstreamu je v tomto standardu 10 Gbit/s, pro upstream existují varianty 1,25 Gbit/s nebo 10 Gbit/s. Vyhrazené pásmo vlnových délek pro downstream je 1575–1580 nm. Pásmo vlnových délek pro upstream 1 Gbit/s je stejné

jako u EPON, tedy 1260–1360 nm. Pro novou rychlost 10 Gbit/s se používá pásmo 1260–1280 nm. Ve standardu se objevuje více variant útlumu:

- PR10/PRX10 – dosah 10 km, rozbočení 1:16.
- PR20/PRX20 – dosah 20 km, rozbočení 1:16 nebo dosah 10 km, rozbočení 1:32.
- PR30/PRX30 – dosah 20 km, rozbočení 1:32.

První dvě výše uvedené varianty platí také pro EPON sítě. Označení PR znamená symetrickou variantu 10 Gbit/s, v případě PRX se jedná o variantu 10 Gbit/s downstream a 1,25 Gbit/s upstream [3]. PR a PRX reprezentují výkonový rozpočet (power budget) jednotlivých variant.

Samozřejmostí je kompatibilita s EPON, kdy je nutno z důvodu vyšších nároků na šířku pásma vyměnit jednotku OLT, poté dle potřeby měnit jednotky ONT. Avšak symetrické varianty 10 Gbit/s není hojně využíváno.

2.4.5 NG-PON

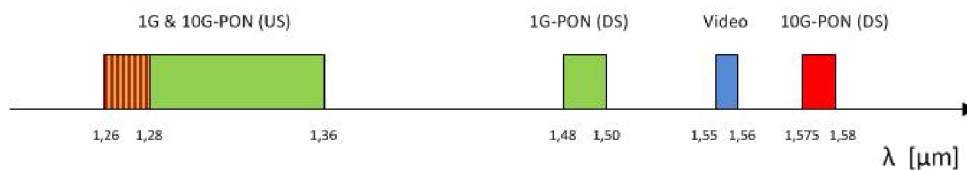
Se stále zvyšujícími se požadavky na síťovou rychlost vzniká standard NG-PON, který je následujícím z rodiny ITU-T a byl představen v roce 2010. Standard NG-PON je znám také pod názvem XG-PON [10]. V rámci standardu ITU-T G.987 vznikají dvě etapy projektu XG-PON sítí:

- NG-PON1: 10 Gbit/s downstream, 2,5 Gbit/s upstream,
- NG-PON2: 10 Gbit/s downstream, 10 Gbit/s upstream.

Stejně jako standardy zavedené před 10G-PON i pro tento standard platí, že je zpětně kompatibilní se staršími. Uživatelé nemusí měnit ONU, je nutné pouze vyměnit jednotku OLT, která je schopna obsluhovat síť kde koexistují standardy GPON i NG-PON. Pouze u NG-PON2 je nutno pořídit nové jednotky ONU (ONT) obsahující laditelné filtry pro specifické vlnové délky. Varianta NG-PON2 však není nyní příliš používaná. Pro nově zavedené rychlosti byla zavedena pásma vlnových délek:

- 1575–1580 nm pro downstream,
- 1260–1280 nm pro upstream.

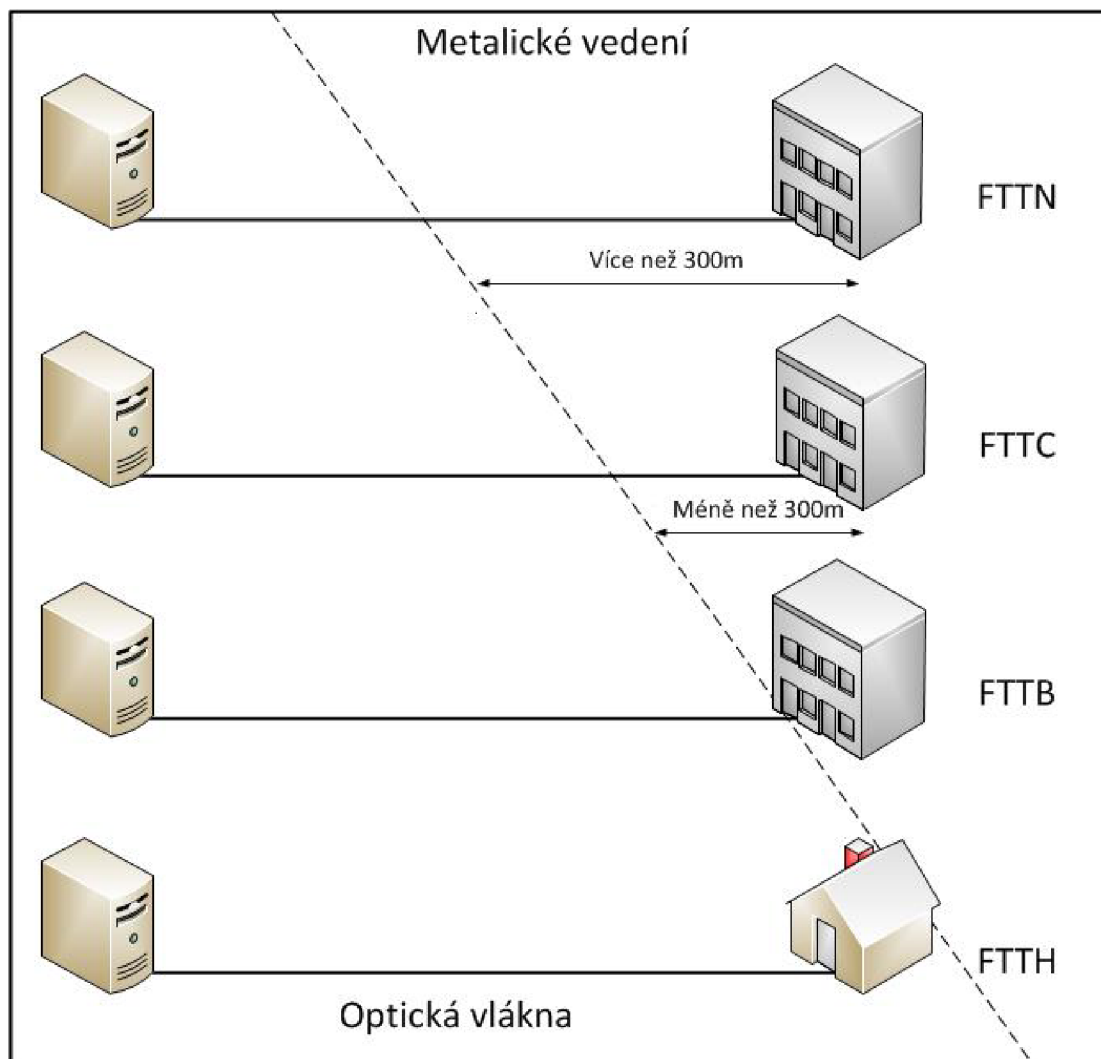
Používaná pásma standardy G-PON a 10G-PON je možno vidět na obr. 2.4. Dosah sítě ve standardu G.987 byl stanoven na 20 km a 60 km pro sítě dlouhého dosahu.



Obr. 2.4: Multiplex 1G a 10G-PON [14].

3 OPTICKÉ SÍTĚ FTT_x

Zkratka „Fibre to the x“ označuje skupinu možností, jak mohou být koncoví uživatelé připojeni pomocí optických vláken [17]. Písmeno „x“ označuje koncový bod, kam je optické vlákno zavedeno, tedy jak velkou vzdálenost k uživatelské přípojce je nutno překonat jinou technologií, zejména metalickými kabely [11]. Na obr. 3.1 je možno vidět různé varianty FTT_x.



Obr. 3.1: Varianty přípojek FTT_x [17].

- **FTTN**

Fibre to the node je variantou, kdy je koncové zařízení optické sítě umístěno do vzdálenosti 1500 m od koncových uživatelů. Koncová přípojka je pak realizována pomocí metalického vedení.

- **FTTC**

Varianta Fibre to the curb je velmi podobná FTTN, metalická část vedení je zde omezena do vzdálenosti 300 m.

- **FTTB**

Fibre to the building je první variantou, která počítá s umístěním koncového optického prvku přímo v budově, které je připojení poskytováno. Rozvody jsou dále realizovány metalicky strukturovanou kabeláží nebo bezdrátově. Varianta počítá s větším počtem připojených uživatelů v rámci budovy.

- **FTTH**

Připojení Fibre to the home zavádí optické vlákno až do domu. Zde se předpokládá jeden platící uživatel. Rozvody v domě již nejsou standardizovány a jsou realizovány dle požadavků uživatele (Ethernet či bezdrátově pomocí Wi-Fi).

- **FTTA**

Fiber to the antenna je variantou zakončení, kdy jsou optická vlákna použita mezi základnovou stanicí a anténní věží. Optická vlákna nahrazují koaxiální vedení z důvodu nutnosti vybudování více anténních věží, které pokrývají menší území při použití nejnovějších technologií.

Zde platí pravidlo: „Řetěz je silný tak, jak je silný jeho nejslabší článek“. To znamená, že při použití nevhodné technologie či prvků dochází ke snížení rychlosti připojení na nejvyšší hodnotu, kterou podporuje nejslabší prvek. Platí také fakt, že čím větší délka připojení je realizována pomocí metalického vedení, dochází ke zmenšení šířky pásma.

4 TRIPLE PLAY

Již v názvu je obsaženo, co je služba schopna poskytnout – tzv. „Trojitou hru“. Služba připojila k již zavedenému poskytování klasických datových služeb další dvě služby multimediální. Jedná se o služby hlasové (VoIP) a služby videa (IPTV). Cílem Triple Play není pouze přechod uživatelů z jiných forem poskytování TV a hlasových služeb, ale hlavním heslem Triple Play je: „Více služeb za nižší ceny“. Triple Play přináší mnoho výhod, zejména v oblasti TV služeb.

Některé z aplikací Triple Play [4]:

- přenos televizního signálu ve vysokém rozlišení (HDTV),
- placené události (PPV),
- video na vyžádání (VoD),
- videokonference,
- hlasový mail,
- rádiové vysílání,
- přenos souborů.

4.1 Problematika Triple Play

Služba je integrována do IP (Internet Protocol) přenosových sítí a přináší s sebou spoustu úskalí. V rámci přenosu dat je využíváno TCP (Transmission Control Protocol), spolehlivého protokolu. Nově zavedené služby však tohoto protokolu nemohou již ze své podstaty využít. Nově integrované služby (např. PPV) využívají nespolehlivého protokolu UDP (User Datagram Protocol), pro zaručení požadované kvality je třeba zvolit nastavení sítě tak, aby byly tyto služby doručeny v požadované kvalitě. Tohoto se docílí zavedením QoS (Quality of Service).

4.2 VoIP

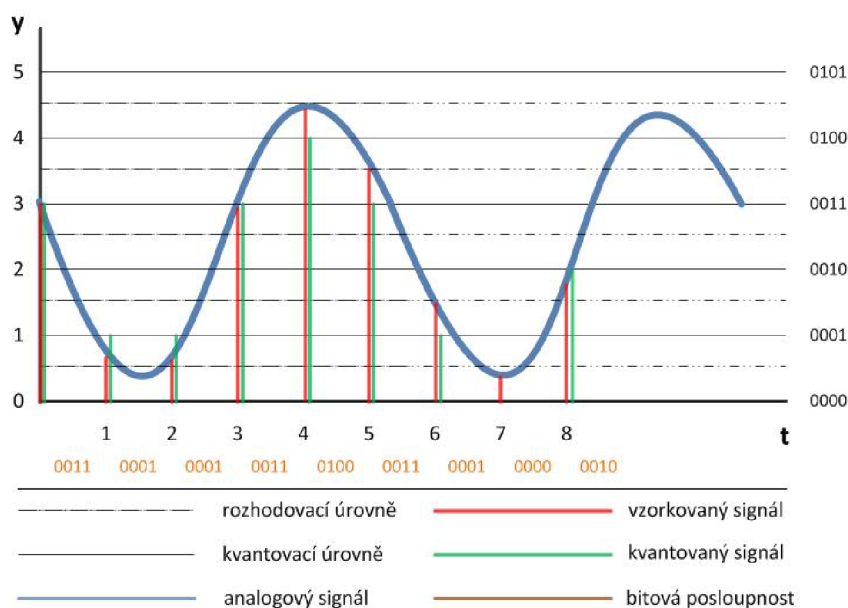
V minulosti se jako první telefony objevily analogové přístroje. Tyto přístroje přenášely hlas skrze vedení jako spojitý analogový signál pomocí frekvenčního dělení kanálu FDM (Frequency-Division Multiplexing). Dalším krokem evoluce telefonních přenosů byla digitalizace, která se objevila v 60. letech. Ta přinesla do oblasti přenosu hlasu velkou novinku v podobě převodu analogového signálu na signál s diskretními hodnotami. Přechod na digitální podobu signálu měl za následek například možnosti jako uchování záznamu, šifrování, kompresi atd.

V dnešní době přichází další revoluce: přechod telefonních hovorů do IP sítí. Ačkoliv tato revoluce není tak významná jako přechod z analogové do digitální komunikace, přináší sebou mnoho výhod jak pro poskytovatele služeb (zavedení nové služby do nabídky, zjednodušení správy), tak pro samotné uživatele (nižší náklady na provoz). Díky nízké ceně, vysoké dostupnosti a stále se zvyšující kvalitě se z VoIP (Voice over Internet Protocol) komunikace stává rychle rostoucí odvětví. Největší výhodou IP telefonování je možnost konvergence přenosu hlasových a datových služeb. Tato výhoda je ovšem také achylovou patou. Zavádění VoIP má také svá úskalí, která musí být řešena, aby výsledná služba měla požadovanou kvalitu. Hlavním problémem je, že většina současných sítí včetně Internetu není navržena pro tento typ komunikace, proto je nutno zavádět jistá pravidla – politiky QoS, kterých nebylo v klasické telefonní síti potřeba. V IP sítích je mnoho efektů, jako fronty, jitter (kolísání zpoždění), buffering (načítání), které způsobují vyšší end-to-end delay (zpoždění), které ve výsledku může být větší než v klasické telefonní síti [4].

4.2.1 Přenos signálu

Než může být vyslán signál, který zařízení zaznamená, musí dojít k níže uvedeným krokům [4]. Ty jsou zobrazeny na obr. 4.1.

1. Vzorkování: Analogový hlasový signál získaný mikrofonom musí být převeden na řadu diskrétních vzorků.
2. Kvantování: Vzorky jsou kvantovány na hladiny.
3. Kódování: Každý vzorek je převeden na kódované bitové slovo.



Obr. 4.1: Digitalizace signálu [16].

Výsledkem těchto operací je bitový tok, který je vyslán do sítě. S ohledem na efektivnost musí být vzorkování prováděno s co nejnižší frekvencí s ohledem na zachování požadované kvality. Také musí být dodržen Shannon-Kotělnikovův teorém, který říká, že přesná rekonstrukce spojitého, frekvenčně omezeného signálu z jeho vzorků je možná tehdy, pokud byla vzorkovací frekvence vyšší než dvojnásobek nejvyšší harmonické složky vzorkovaného signálu. Úkolem kódování je zajistit co nejkratší bitovou délku slova pro každý ze vzorků. Kvantování využívá nedokonalosti lidských vjemů, pro kvalitní přenos je však nutno zachovat potřebný stupeň kvality. Hlavními povětšinou kvalitativními znaky přenosu je zřetelnost a nezkrácenost přenášeného signálu. Příklady snahy o co nejvyšší efektivnost:

- Není přenášeno celé pásmo slyšitelné lidským uchem (20 Hz–20 kHz) ale pouze frekvence v rozmezí 300–3400 Hz, kdy se v tomto frekvenčním pásmu nachází asi 80 % informací z původního zvuku [4].
- Některé z používaných kodeků generují data pouze v okamžiku, kdy uživatel hovoří. Data nejsou generována v intervalech ticha.

4.2.2 Parametry sítě

Mezi parametry popisující výkon sítě patří:

- Signal-to-noise-ratio (SNR) – Odstup signál-šum.
- Bit Error Rate (BER) – Chybovost.
- End-to-end delay – Jednosměrné zpoždění přenosu.

Všechny tyto parametry udávají hodnoty o kvalitě přenášeného signálu, ale jedná se pouze o servisní hodnoty. Z těchto hodnot je možno usoudit, zda je přenos kvalitní, konečný výsledek je ale také závislý na vnímání uživatelů. Jinak řečeno, servisní hodnoty mohou být velmi kvalitní, ale bez přiřazení k poskytnuté službě nevíme, zda je možno přenos hodnotit jako kvalitní či nikoli.

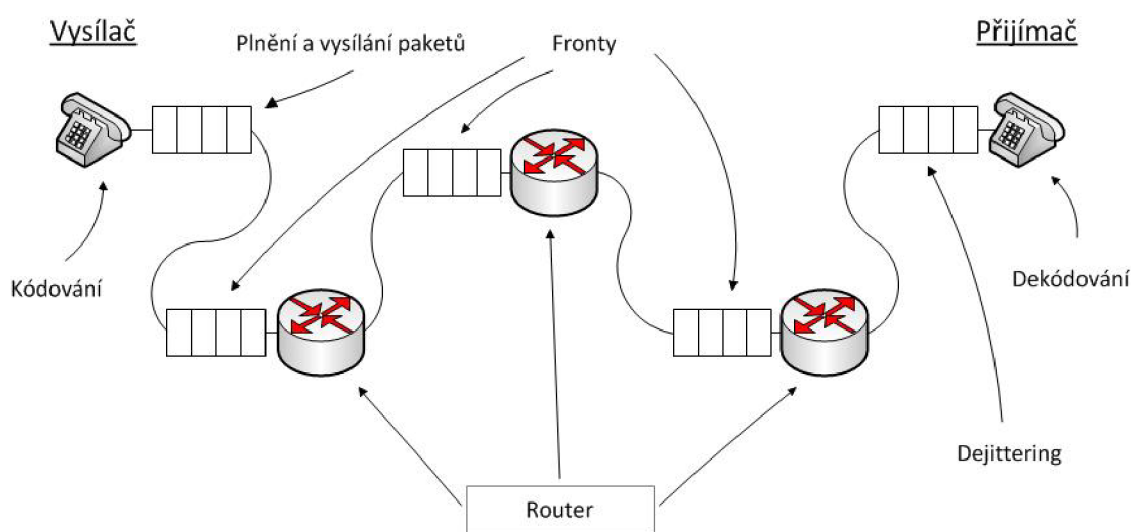
Hlavními parametry popisujícími přenos hlasu v IP sítích jsou:

- **Packet loss – ztrátovost paketů**
Čím je ztrátovost paketů vyšší, tím dochází k většímu zkreslení v hlasových aplikacích. Pokud je ztrátovost paketů nižší než 1 %, měl by být dopad na kvalitu hlasu minimální či žádný. Pro ošetření části komunikace, kdy byl paket ztracen je používáno PLC (Packet Loss Concealment). V případě PLC se jedná o algoritmus, odstraňující místa bez paketu přidáním tichého momentu, šumu v pozadí nebo nejčastěji opakovaným přehráním posledního přijatého paketu. PCL je schopno udržet kvalitu hovoru ve vysoké míře až do ztrátovosti 3 %. Vyšší procentuální ztrátovost se již projevuje jako citelné snížení kvality přenosu.

- **End-to-end delay – zpoždění přenosu**

Jeden z nejdůležitějších parametrů v rámci volání z pohledu uživatele. Z pohledu síťových parametrů se dá přirovnat k jednosměrnému zpoždění. Na rozdíl od přenosů po klasických telefonních linkách, na kterých probíhal pouze jeden druh komunikace, je v prostředí IP sítí udržení nízkého zpoždění jednou z největších výzev. V IP sítích probíhá paralelně více druhů komunikace, se zvyšujícím se zatížením roste zpoždění u všech druhů služeb, proto se u IP hovorů setkáváme s vyšším zpožděním.

Nejpodstatnější zdroje zpoždění je možno vidět na obr. 4.2.



Obr. 4.2: Zdroje zpoždění [4].

Doporučení G.114 vydané ITU-T zmiňuje zpoždění do 150 ms jako nepovšimnutelné. Při vyšších dobách zpoždění dochází ke zhoršení plynulé komunikace. Při pohledu na množství zdrojů je možno vidět, že udržení zpoždění pod 150 ms je velmi obtížné, zejména pokud se bude jednat o hovory na velkou vzdálenost [4]. End-to-end delay je také původcem Echa – nepříjemné vlastnosti, kdy dochází ke spojování signálů. Echo zůstává nepostřehnutu při nízkých zpožděních, avšak při zvyšujícím se zpožděním mezi originálním a replikovaným signálem dochází k zesílení echa na úroveň, kdy je již znatelné. Pro odstranění echa se v IP sítích, stejně jako v tradičních používají Echo cancellers. Echo cancellers jsou díky digitálnímu signálovému procesoru schopny rozpoznat opakující se signál a odečíst jej od signálu vysílaného.

4.3 IPTV

Internet Protocol Television neboli Televize přes vysokorychlostní síť je jednou z částí Triple Play [13]. IPTV je metoda doručení tradičních až stovek kanálů k zákazníkům pomocí privátních IP sítí. Služba IPTV se liší od jiných forem vysílání svou rozsáhlostí. Samostatné TV vysílání není strhujícím pokrokem, ale nabídka, která se pod pojmem IPTV skrývá je rozsáhlá. IPTV se liší také tím, že na rozdíl od klasického vysílání máme možnost komunikace zpět k poskytovateli, což je nezbytné pro mnoho služeb [13].

Některé ze služeb IPTV:

- video na vyžádání (VoD),
- placené události (PPV),
- IPTV,
- e-mail,
- iTV (Interactive television).

Přenos klasického vysílání v reálném čase probíhá pomocí multicastových skupin (přenos obsahu pro velký počet uživatelů), ovšem u služeb jako je VoD, PPV atd. dochází k vysílání v unicast režimu (je sestaveno individuální spojení od zdroje k divákovi) [15].

4.3.1 Rozdíl IPTV a internetového videa

Přenos pomocí IP sítí probíhá také u internetových videí, ovšem mezi těmito službami jsou podstatné rozdíly [15]:

1. IPTV

- IPTV sítě jsou budovány za účelem přenosu videa a poskytování dalších služeb.
- Je placenou službou.
- Služba reálného času – očekáváno malé zpoždění.

2. Internetové video

- Internetové video je přenášeno skrze již existující síť.
- Mnoho služeb je poskytováno zdarma.

4.3.2 Multicastový přenos IPTV

Jednou z odlišností, je-li pominuto médium, přes které je TV vysílána je také forma vysílání. IPTV není na rozdíl od jiných forem vysílána formou broadcastu (všesměrového vysílání) ale formou multicastu. Při broadcast vysílání dochází k multiplexu všech vysílaných kanálů, to je ovšem v rámci IP sítí nemožné z hlediska šířky pásma.

Požadované přenosové rychlosti pro vysílání IPTV za použití různých kodeků jsou uvedeny v tab. 4.1 [4].

Tab. 4.1: Požadavky SDTV a HDTV na přenosovou rychlost po kompresi [4].

	MPEG-2	MPEG-4	WM9
SDTV	3,5 Mbit/s	2–3,2 Mbit/s	2–3,2 Mbit/s
HDTV	15 Mbit/s	7,5–13 Mbit/s	7,5–13 Mbit/s

Vysílání probíhá za pomoci protokolu IGMP (Internet Group Management Protocol), kterým se uživatel zaregistruje do jedné z multicastových skupin. Každý kanál (program) má přiřazenu svou multicastovou adresu, na které je vysílán. Přiřazení do skupiny provádí uživatelská jednotka, průběh hlavních částí [15] [18]:

- Uživatel zvolí požadované vysílání nebo zapne zařízení.
- Protokol IGMP vysílá požadavek na jednotku poskytovatele o zařazení do nové multicastové skupiny a opuštění staré.
- Pokud má uživatel oprávnění (zaplacen daný kanál), je přiřazen do skupiny a jsou vysílána data.

Při změnách programu se objeví jedna ze slabin IPTV. Během přepínání kanálů nedochází k výběru jiné části přijímaných dat z vysílání jako v klasickém televizním signálu, ale musí dojít ke komunikaci mezi uživatelským přijímačem a jednotkou na straně poskytovatele. Komunikace není okamžitá, proto může přepínání kanálů trvat až jednotky sekund [13].

4.3.3 Video on Demand

Služba „Video na vyžádání“ je jedním z největších taháků Triple Play. Umožňuje uživateli, aby zvolil, jaký pořad na své obrazovce aktuálně uvidí. Stačí si pouze „objednat“ přenos [18]. Objednání služby může proběhnout dvěma způsoby:

1. Obsah je přehráván z VoD serveru.
2. Obsah je stažen do účastnického zařízení.

Obě tyto varianty poskytují uživateli možnosti typické pro klasické DVD (Digital Video Disc) přehrávače jako přehrát, pozastavit, posun vpřed nebo vzad.

Pokud uživatel využívá varianty přehrávání ze serveru, dochází k unicast události s TCP přenosem. V tomto případě musí být zabezpečeno, aby vysílání bylo doručeno v požadované kvalitě.

Při zvolení varianty stažení obsahu do zařízení na straně účastníka, dochází ke stahování obsahu metodou 'best-effort'. Pokud jsou během stahování přítomny další přenosy, je bráno v úvahu nastavení QoS a může docházet k jejich upřednostnění [4].

4.3.4 Pay per View

Další službou v rámci Triple Play je „Placená událost“. Jedná se o službu, která nabízí uživatelům možnost objednat si přenos, na který nemají v rámci své nabídky práva. Může se také jednat o jednorázovou sportovní nebo hudební událost.

Možnost objednání služby se nabízí několik dnů až týdnů před začátkem samotné události, objednat ji lze až do jejího zahájení. Po objednání a zadání bezpečnostního kódu uživatele dochází k zápisu uživatele do databáze uživatelů, kteří mají tuto událost přístupnu [15].

Poskytovatel v čase události vysílá pro všechny, kteří mají zapláceno a jsou připraveni vysílání přijímat. Nelze si tedy vybírat, kdy je žádaný obsah přehrán (na rozdíl od VoD).

4.4 Quadruple Play

Quadruple play je vizí do budoucna. Jedná se o variantu, kdy se k balíku Triple Play přidávají mobilní služby. Rozšíření služby dává smysl jelikož mobilní operátoři se snaží do svých nabídek přidat další služby. Cílem je, aby byli schopni konkurovat drátovým poskytovatelům svými mobilními zařízeními. Služby jako mobilní TV nebo multimediální konference se zdají býti strategickými službami.

Snahou operátorů je také zavedení mobilních hovorů do prostředí IP sítí [4]. K tomu mají přispět sítě WLAN (Wireless Local Area Networks). WLAN technologie je schopna poskytnout mnohem větší šířku pásma než jakékoli stávající sítě mobilní. I když WLAN sítě jsou schopny pokrýt pouze malé geografické oblasti, jsou instalovány díky jejich schopnosti poskytnout vysokorychlostní připojení. Klasická místa, kde jsou budovány tyto sítě jsou nádraží, nákupní centra, letiště apod. Hlavními problémy, které brání mobilním hovorům skrze IP sítě jsou:

- Neexistující zařízení, která by byla schopna provozovat hovory v mobilních i IP sítích v rámci jedné karty.
- Neexistující technika pro autentizaci, autorizaci a zpoplatnění uživatele.

5 QOS (QUALITY OF SERVICES)

IP sítě prošly úspěšným, více než 30 let trvajícím vývojem, mají mnoho předností jako flexibilitu a efektivnost, avšak nejsou schopny dodržet parametry jako malou ztrátovost, zpoždění, proměnlivost zpoždění. IP sítě poskytují služby technikou „best-effort“.

QoS neboli kvalita služeb je nástroj pro zajištění prostředků pro různé typy aplikací tak, aby mohly být kvalitně využívány. Jedná se především o zajištění prostředků pro služby v reálném čase jako jsou například VoIP telefonní hovory, videokonference nebo IPTV. Každá z těchto služeb má jiné požadavky na síťové parametry. Dosažením požadované kvality služeb pomocí řízení zpoždění, proměnlivosti zpoždění, šířky pásma a ztrátovosti je základem pro úspěšnou komunikaci. Pro zajištění požadované kvality služeb byly vyvinuty dvě metody, které jsou v současných sítích využívány [4]:

- IntServ (Integrated Services – Integrované služby).
- DiffServ (Differentiated Services – Diferencované služby).

Každá z těchto metod využívá jiného principu pro zajištění QoS. Dokonalým řešením není ovšem ani jedna z těchto metod, proto se pro dosažení co nejlepších výsledků používá jejich kombinace.

5.1 IntServ

Integrované služby využívají pro zajištění kvality služeb předem nasmlouvaných požadavků. Od zdroje je skrz všechny uzly v síti vysílána zpráva s přesnými požadavky pro přenos směrem k cíli. Pokud jsou uzly sítě schopny poskytnout záruku dodržení požadavků, je sestaveno spojení s rezervovanými prostředky. Pokud nejsou uzly schopny požadavky splnit, musí dojít ke změně požadavků. Pro zajištění QoS, musí být metoda IntServ implementována ve všech uzlech po celou dobu komunikace. Rezervace zdrojů probíhá za pomoci protokolu RSVP (Resource Reservation Protocol).

Metoda je schopna dodržet požadavky přenosu, její efektivnost je ale z důvodu velké režie nižší. Proto dochází k využívání této metody především v sítích malého rozsahu, především v sítích firemních [4] [5].

5.2 DiffServ

Metoda diferencovaných služeb nepoužívá ve své podstatě rezervování kapacit jako IntServ, ale je postavena na myšlence kategorizace provozu. Třídy mají nastaveny

dle CoS (Class of Service) kvalitativní parametry přenosu [4]. Díky rozdělení toků u této metody není potřeba před spojením sestavovat spojení, nevzniká zde režijní provoz a není potřeba signalizačního protokolu.

K využívání DiffServ postačí mít správně definován typ služby a zapnutou podporu na všech routerech sítě. Samozřejmě je potřeba na všech routerech předem nadefinovat, jak má být se skupinami paketů zacházeno. Do jaké třídy paket patří a jak s ním má být zacházeno je značeno v poli ToS (Type of Service) pro IPv4 nebo v poli Traffic Class pro IPv6.

Pole ToS obsahuje 8 bitů, používáno je ovšem pouze bitů 0 až 5, bity 6 a 7 zůstávají nevyužity [5]. Bity 0–5 jsou obecně označovány jako DSCP (Differentiated Services Code Point). Dle pole DSCP se jednotlivé směrovače chovají. Toto chování je označováno jako PHB (Per Hop Behaviour) [1]. PHB je nastavováno provozovatelem sítě dle vlastního uvážení, pro správnou funkci by však mělo být nastaveno v celé síti stejně.

Velmi často je využíváno tří standardizovaných kategorií:

- **Expedited Forwarding (EF) – Urychlené posílání**

Pro kategorii je specifické nízké zpoždění, proměnlivost zpoždění a ztrátovost. Kategorie má nejvyšší prioritu, pakety jsou tedy v routerech upřednostňovány před ostatními. Do této kategorie jsou řazeny služby hlasu, videa a aplikací v reálném čase. Hodnota DSCP pole je 101110.

Existuje také kategorie VA (Voice Admit), která se shoduje s EF, je zde však využíváno dalších protokolů. Hodnota DSCP pole pro tuto kategorii je 101100.

- **Assured Forwarding (AF) – Zajištěné posílání**

V této kategorii existují 4 třídy, v každé ze tříd existují 3 možnosti (low, medium, high) pro nastavení zahazování paketů. Tabulka 5.1 ukazuje závislost DSCP pole na třídě a prioritě zahození. Celkem má kategorie 12 možností DSCP pole.

Tab. 5.1: DSCP pole kategorie Assured Forwarding [4].

DSCP Assured Forwarding	1.třída	2.třída	3.třída	4.třída
Nízká priorita zahození	100010	011010	010010	001010
Střední priorita zahození	100100	011100	010100	001100
Vysoká priorita zahození	100110	011110	010110	001110

- **Class Selector (CS)**

Kategorie s nejnižší prioritou, kde pole DSCP má tvar „xxx000“. V rámci kategorie Class selector je umožněno značit 7 stupňů priorit dle tab. 5.2.

Pokud je pole DSCP tvaru 000000 jedná se o službu typu **Best-effort**.

Tab. 5.2: Stupně priorit kategorie Class Selector [4].

Class Selector	DSCP pole
1.stupeň	111000
2.stupeň	110000
3.stupeň	101000
4.stupeň	100000
5.stupeň	011000
6.stupeň	010000
7.stupeň	001000

Značení paketů provádí hraniční router, může jej provádět i účastnické zařízení, jeho značení router může přijmout nebo přenastavit. V rámci sítě je s pakety zacházeno dle nastavení provozovatele. Pokud ovšem paket opouští síť provozovatele, musí na hraničním routeru dojít k přeznačení paketu. K tomuto účelu slouží Dohoda o úrovni služeb – SLA (Service Level Agreement), ve které je stanoveno jak mají být pakety přeznačeny a jak je s nimi dále zacházeno [18].

6 BITTORRENT PROTOKOL

BitTorrent protokol vyvinul Bram Cohen v roce 2001. Jedná se o protokol typu peer-to-peer neboli rovný s rovným, tedy protiklad sdílení dat pomocí modelu klient-server. Výhodami tohoto protokolu jsou:

- rozdělení nároků na uložení na celou síť,
- snížení nákladů pro poskytovatele dat,
- možnost návaznosti stahování.

Protokol nepracuje s daty jako s celkem, nýbrž provádí rozdělení celého souboru na bloky. Velikost bloků se pohybuje od desítek kB po maximálně 4 MB a závisí především na celkové velikosti souboru. Velké bloky je možno dělit na subčásti [6].

Základní pojmy

- **Tracker**

Jedná se o server zajišťující komunikaci mezi jednotlivými uživateli. Jeho úkolem je shromažďovat data o uživateli a jejich aktivitě. Klienti trackeru v intervalech oznamují, které bloky souboru vlastní a posílají také zprávy typu peer_id obsahující IP adresu a port na kterých jsou dostupní. Komunikace s trackerem probíhá architekturou klient-server, většinou na portu 6969. Tracker nevlastní žádná data, která jsou součástí užitečných dat.

- **Seed**

Jedná se o klienta, který vlastní všechny bloky požadovaných dat (celý soubor). Sám již nic nepožaduje, pouze nabízí data dalším klientům (uživateli). Komunikuje pouze s trackerem a čeká na příchozí spojení.

- **Leech**

Leech označuje uživatele, který nemá stažena kompletní data a stále stahuje. Dalším leech uživateli poskytuje již stažené bloky.

- **Peer**

Peer je souhrnným názvem pro všechny uživatele, nezáleží zda stahují nebo sdílí.

- **Swarm**

Skupina peerů podílejících se na sdílení.

Vytvoření torrent souboru a publikace

Soubor *.torrent*

Vytvoření podporují BitTorrent klienti. V rámci tvorby souboru je nutno zadat soubor nebo složku, která má být zveřejněna a adresu trackeru, která vypadá v obecném formátu `udp://adresa_trackeru.doména:80/announce`. Pro sdílení souboru pouze s jedním uživatelem, může být použito jako trackeru vlastního PC. Požadavkem ovšem je veřejná IP adresa.

Prvotní seedování

Po vytvoření příslušného *.torrent* souboru musí dojít k jeho kontrole. Uživatel, který soubor publikuje otevře nově vytvořený *Soubor.torrent*. BitTorrent klient provede kontrolu souboru, který již uživatel vlastní. Pokud se tyto soubory shodují, stane se uživatel (tvůrce) prvotním neboli initial seederem.

Stahování a sdílení

Získ metadat

Uživatel, snažící se o stažení souboru, vyhledá na webu soubor s příponou *.torrent* obsahující metadata k požadovanému souboru. BitTorrent klient po otevření získá ze souboru data jako jméno, velikost, URL trackeru, velikost bloku a binárně uložené kontrolní sumy bloků. Příklad úvodní textové části:

```
d8:announce44:udp://tracker.openbittorrent.com:80/announce10:created
by13:uTorrent/330013:creation datei1426335957e8:encoding5:UTF-84:
infod6:lengthi2705296e4:name13:Wireshark.exe12:piece lengthi65536e6:
pieces840:
```

URL trackeru:	udp://tracker.openbittorrent.com:80/announce,
vytvořil:	uTorrent verze 3.3,
vytvořeno:	14.3.2015 13:25,
kódování:	UTF-8,
délka v B:	2705296,
jméno:	Wireshark.exe,
délka bloku v B:	65536,
počet bloků:	40.

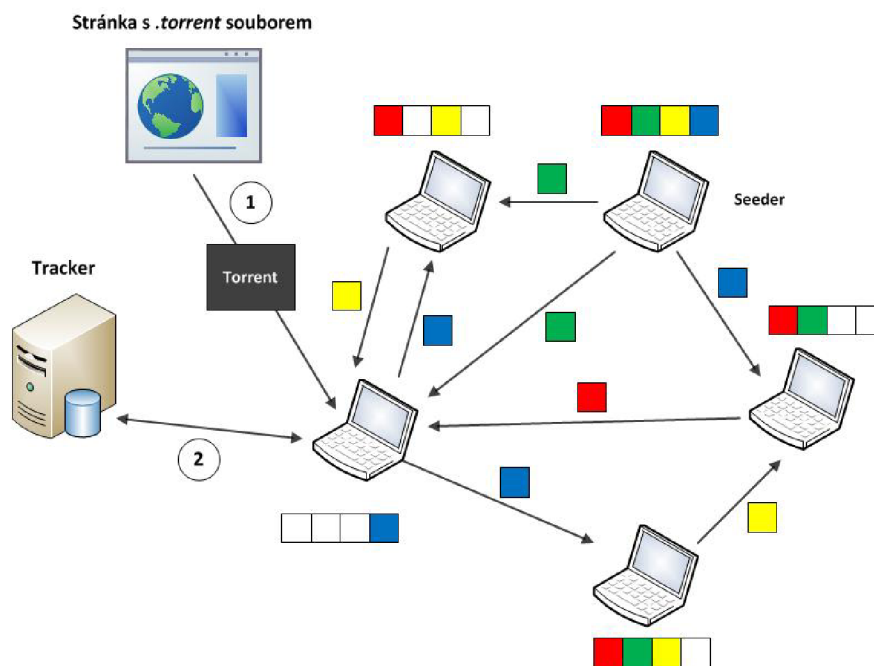
Stahování

Zpočátku stahování kontaktuje klient tracker. Tracker v odpovědi poskytne klientovi informace o několika dalších peerech (v rámci DHT (Distributed Hash Table) tabulky) podílejících se na přenosu souboru. Klient se díky těmto informacím k peerům připojí a získá od nich seznam bloků, které následně požaduje, jak je naznačeno na obr. 6.1.

Pořadí požadovaných bloků je z důvodu zvýšení šance na sdílení mezi leechery náhodné. V současné době se používají dvě strategie:

- **Random first** – technika náhodného výběru, nutnost získání jakéhokoliv bloku.
- **Rarest first** – technika získávání bloků od nejvzácnějších, které připojení peeri mají. Zajišťuje stahování nejběžnějších bloků jako posledních.

Klient v intervalech také kontaktuje tracker o blocích, které již získal. Vlivem postupného získávání bloků a informování trackeru dochází k postupnému nárůstu rychlosti stahování. Ze začátku stahování mohou klienti ostatní klienty podvést, když na žádosti o bloky žádné nevysílají nebo vysílají náhodná data, která si peer nevyžádal. Klienti jsou ovšem tyto praktiky schopni odhalit a na žádosti o další bloky neodpovídají (snaží se vyměňovat své bloky alespoň v poměru 1:1) [12]. Pro zamezení stavu, kdy by klient na své žádosti žádné další bloky nedostal, získává klient z trackeru seznam dalších peerů.



Obr. 6.1: Princip sdílení souborů pomocí BitTorrent protokolu [6].

7 IMPLEMENTACE DO NS-3

Tato část práce je věnována především praktické implementaci a simulaci Triple Play služeb v prostředí NS-3 (Network Simulator 3) ve verzi 3.21. Dále je popsán software, který byl použit pro vytvoření zdrojového kódu a interpretaci výsledků a také samotné prostředí NS-3 včetně základních pojmů pro práci v tomto prostředí. V podkapitole 7.3 jsou rozebrány části simulace, jejich zdrojové kódy a grafické výstupy pomocí softwaru NetAnim a trasovacího programu Wireshark.

7.1 Použitý software

Eclipse Luna

Eclipse je vývojové prostředí primárně určené pro Javu. V současné době se jedná již o open-source, který byl dříve vyvíjen společností IBM.

Eclipse je nástroj vhodný k vývoji softwaru jakéhokoliv rozsahu. Zdrojem zásuvných modulů je tzv. Eclipse Plugin Central. Zde je možno nalézt zásuvné modely pro jazyky C/C++, Python a jiné.

NS-3

NS-3 je volně šířitelný software (freeware). Jedná se o diskretní síťový simulátor, především pro učební a vývojové použití.

Cílem projektu NS-3 je vytvořit celistvý simulační program, který obsahuje dokumentaci, je snadný z hlediska použití a splňuje požadavky komunity využívající tento program jako: snadnou konfiguraci simulací, sledování a analýzu výsledných dat. Pro snadnou práci v prostředí existuje také rozsáhlá nápověda na adrese <http://www.nsnam.org/> [19].

Z důvodu, že se jedná o neplacené, volně šířitelné vývojové prostředí je práce na vývoji rozšířena také mezi veřejné uživatele a vývojáře. Každé 3 měsíce je vydána nová stabilní verze, která je kvalitně zdokumentovaná, ověřená a udržovaná.

NetAnim

Animace je důležitou součástí síťových simulátorů. V rámci práce na tomto dokumentu bylo využíváno simulačního softwaru NetAnim, který je schopen vizualizovat ASCII soubor ve formátu .xml.

Vstupní soubor pro NetAnim je generován ve zdrojovém kódu obecným příkazem:

```
AnimationInterface anim ("animation.xml");
```

Wireshark

Volně šířitelný program pro zachytávání a analýzu síťového provozu, dříve známý jako Ethereal. Program je využíván především při řešení problému v počítačových sítích, při vývoji software nebo pro praktickou ukázkou teoretických znalostí v rámci vzdělávání.

Pro zjednodušení analýzy dat je možno v programu vytvářet filtry získaných dat (jak dle použitého protokolu tak například i dle zdrojové či cílové adresy), označit a sledovat uživatelem vybranou komunikaci.

7.2 Práce v NS-3

Zdrojový kód je pro prostředí NS-3 psán v jazyce C++, jedná se tedy o objektové programování. Základními objekty v prostředí NS-3 jsou:

- **Node**
Základní prvek, představující funkční uzel sítě. Pro pochopení se může jednat o PC s vlastní pamětí, procesorem, vstupními a výstupními rozhraními.
- **Application**
Aplikace představující generátor nebo příjemce dat. Aplikace běží na konkrétních uzlech sítě (node). Příklady aplikací, užitých také v rámci této práce jsou: OnOff, PacketSink, UdpEchoClient, UdpEchoServer.
- **NetDevice**
Síťové zařízení (síťová karta), která je vložena do rozhraní uzlu.
- **Channel**
Kanál – spojení mezi síťovými zařízeními (kartami). Příklad kanálu: PointToPoint.

Příklad, jak vytvářet uzly, aplikace a kanály je možno vidět v dále vloženém kódu.

Pro různé objekty, existují v rámci NS-3 kontejnery. V těchto objektových kontejnerech jsou vytvářeny uzly nebo rozhraní, které jsou dále použity ve specifické části simulace, jako například NodeContainer v dále umístěném kódu, obsahující uzly pro vytvoření klientské části simulované sítě označeny jako clientNodes. Příklady kontejnerů:

- NodeContainer,
- NetDeviceContainer,
- Ipv4InterfaceContainer.

Pro každou aplikaci nebo třídu existuje v prostředí NS-3 Helper. Pomocí Helperů je uživateli umožněno definovat vlastnosti aplikací nebo tříd, jak je možno vidět v kódu níže. Zavedení helperů pomáhá minimalizovat délku kódu a zjednodušit kód. Příklad helperů:

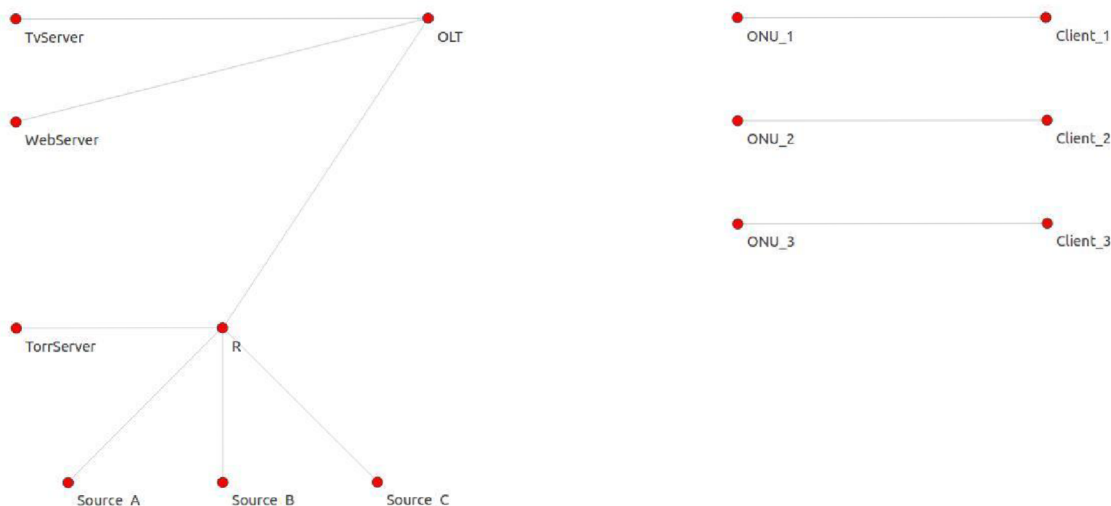
- InternetStackHelper,
- Ipv4AddressHelper,
- OnOffHelper.

```
NodeContainer clientNodes , serverNodes ;
clientNodes . Create ( nOnus ) ;
serverNodes . Create ( nOnus ) ;

PointToPointHelper pointToPoint ;
pointToPoint . SetDeviceAttribute ( "DataRate" , StringValue ( "100Mbps" ) ) ;
pointToPoint . SetChannelAttribute ( "Delay" , StringValue ( "2ms" ) ) ;
```

7.3 Simulace

Pro vlastní simulaci Triple Play služeb byla v prostředí NS-3 vytvořena topologie zobrazená na obr. 7.1.



Obr. 7.1: Topologie simulace.

Topologie se skládá ze serverové, přístupové, klientské a internetové části.

Serverová část je tvořena uzlem TvServer, který představuje v jednotlivých částech simulace server služby Pay per View, Video on Demand a IPTV. Druhým uzlem je WebServer, představující webový server poskytovatele služeb.

Klientskou část tvoří uzly optických síťových jednotek ONU a účastnické uzly Client_1, Client_2 a Client_3.

Poslední simulovanou částí ze strany poskytovatele služeb je modul XG-PON představující optickou přístupovou síť pro připojení klientů.

Internetovou část simulace tvoří hraniční router poskytovatele služeb, uzel Torr-Server představující torrentový server včetně trackeru a tři uzly (Source_A, Source_B, Source_C) představující zdroje/příjemce komunikace mimo část sítě poskytovatele služeb. Tato část simulované sítě je zjednodušena do verze, která je dostačující pro zobrazení principu BitTorrent protokolu.

Hlavním zdrojem dat pro Wireshark bylo zvoleno rozhraní jednotky OLT v optické přístupové síti, jehož výstupem je soubor Bachelor-xgpon-0-0.pcap.

7.3.1 Webové stránky

Čas simulace 0–3 s.

První částí simulace je komunikace na principu klient-server. Client_1 komunikuje na WebServer (v tomto případě poskytovatele služeb, například za účelem kontroly zaplacení účtu za připojení). Komunikace probíhá pomocí HTTP (Hypertext Transfer Protocol) protokolu na portu 80, jak je patrné z níže uvedeného kódu.

```
uint16_t serverPort=80;

ApplicationContainer httpApp;
OnOffHelper httpOnOff ("ns3::TcpSocketFactory",InetSocketAddress
(Ipv4Address(p2pSInterfaces[1].GetAddress(0)),serverPort));
std::cout <<"OnOff_␣http_␣to_␣server_␣at_␣address"<< " :_␣"
<<p2pSInterfaces[1].GetAddress(0)<< " ."<<std::endl;

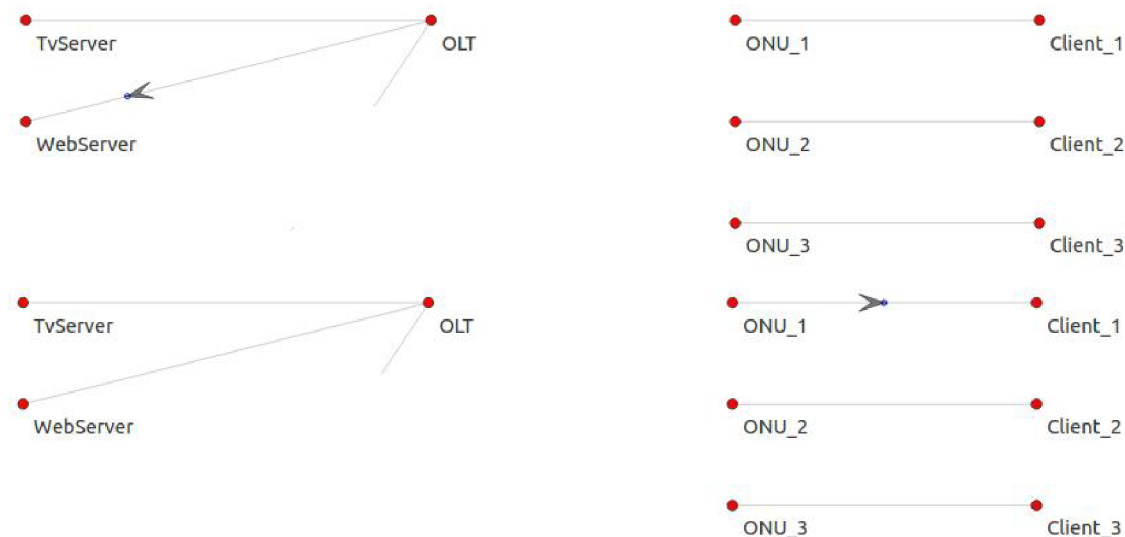
httpOnOff.SetAttribute("OnTime",StringValue("ns3::Constant
RandomVariable[Constant=0.02]"));
httpOnOff.SetAttribute("OffTime",StringValue("ns3::Constant
RandomVariable[Constant=0.15]"));
httpOnOff.SetAttribute("DataRate",DataRateValue(DataRate("3Mb/s")));
httpOnOff.SetAttribute("PacketSize",UIntegerValue(1400));
httpOnOff.SetAttribute("MaxBytes",UIntegerValue(0));

httpApp.Add(httpOnOff.Install(clientNodes.Get(0)));
httpApp.Start(Seconds(httpStart));
httpApp.Stop(Seconds(httpStop));
```

Na vybrané části kódu je také aplikace *httpApp*, která vytváří komunikaci z uzlu Client_1 (*clientNodes.Get(0)*) na server (*p2pSInterfaces[1].GetAddress(0)*) na port

(*serverPort*). Port 80 pro tuto službu byl vybrán na základě definovaných portů organizací IANA (Internet Assigned Numbers Authority). Na straně serveru musí existovat příjemce paketů (sink), v tomto případě pod názvem *httpSinkApp*, na portu 80.

Po skončení simulace je možno si zobrazit animaci pomocí programu NetAnim, jenž je doprovodným softwarem k NS-3. Z uvedeného obr. 7.2 je patrný přenos paketů napříč navrženou sítí.



Obr. 7.2: Komunikace mezi uzly Client_1 a WebServer.

Pro detailnější rozbor komunikace mezi klientem a serverem slouží vygenerovaný soubor Bachelor-xgpon-0-0.pcap. Obsah tohoto souboru zachycuje obr. 7.3, ze kterého jsou patrné IP adresy klienta (160.0.0.1) a serveru (172.0.0.1).

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	160.0.0.1	172.0.0.1	TCP	56	(TCP Port numbers reused) 49153->80 [SYN] seq=4294965896 win=32768 Len=0 w...
2	0.004075	172.0.1.1	160.0.0.1	TCP	56	80->49153 [SYN, ACK] Seq=4294967295 Ack=4294965897 Win=32768 Len=0 WS=4 TS...
3	0.011500	160.0.0.1	172.0.1.1	TCP	52	49153->80 [ACK] Seq=4294965897 Ack=0 win=131072 Len=0 Tsva=10 TSecr=5
4	0.154750	160.0.0.1	172.0.1.1	TCP	1452	49153->80 [ACK] Seq=4294965897 Ack=0 win=131072 Len=1400 Tsva=153 TSecr=5
5	0.158950	172.0.1.1	160.0.0.1	TCP	52	80->49153 [ACK] Seq=0 Ack=1 win=129672 Len=0 Tsva=160 TSecr=153
6	0.166250	160.0.0.1	172.0.1.1	IPv4	1500	Fragmented IP protocol (proto=TCP 6, off=0, ID=0003) [Reassembled in #7]
7	0.166250	160.0.0.1	172.0.1.1	TCP	72	49153->80 [ACK] Seq=1 Ack=0 win=131072 Len=1500 Tsva=165 TSecr=160
8	0.166250	160.0.0.1	172.0.1.1	IPv4	1500	Fragmented IP protocol (proto=TCP 6, off=0, ID=0004) [Reassembled in #9]
9	0.166250	160.0.0.1	172.0.1.1	TCP	72	49153->80 [ACK] Seq=1501 Ack=0 win=131072 Len=1500 Tsva=165 TSecr=160
10	0.170575	172.0.1.1	160.0.0.1	TCP	52	80->49153 [ACK] Seq=0 Ack=3001 win=129572 Len=0 Tsva=172 TSecr=165
11	0.177750	160.0.0.1	172.0.1.1	IPv4	1500	Fragmented IP protocol (proto=TCP 6, off=0, ID=0005) [Reassembled in #12]
12	0.177750	160.0.0.1	172.0.1.1	TCP	72	49153->80 [ACK] Seq=3001 Ack=0 win=131072 Len=1500 Tsva=176 TSecr=172
13	0.177750	160.0.0.1	172.0.1.1	TCP	1152	49153->80 [ACK] Seq=4501 Ack=0 win=131072 Len=1100 Tsva=176 TSecr=172
14	0.182075	172.0.1.1	160.0.0.1	TCP	52	80->49153 [ACK] Seq=0 Ack=5601 win=129972 Len=0 Tsva=183 TSecr=176
15	0.323125	160.0.0.1	172.0.1.1	TCP	1452	49153->80 [ACK] Seq=5601 Ack=0 win=131072 Len=1400 Tsva=322 TSecr=183
16	0.326125	160.0.0.1	172.0.1.1	TCP	1452	49153->80 [ACK] Seq=7001 Ack=0 win=131072 Len=1400 Tsva=326 TSecr=183
17	0.330325	172.0.1.1	160.0.0.1	TCP	52	80->49153 [ACK] Seq=0 Ack=8401 win=129672 Len=0 Tsva=332 TSecr=326
18	0.331250	160.0.0.1	172.0.1.1	TCP	1452	49153->80 [ACK] Seq=8401 Ack=0 win=131072 Len=1400 Tsva=329 TSecr=183
130	2.711000	160.0.0.1	172.0.1.1	TCP	1452	49153->80 [ACK] Seq=113401 Ack=0 win=131072 Len=1400 Tsva=2709 TSecr=2551
131	2.714000	160.0.0.1	172.0.1.1	TCP	1452	49153->80 [ACK] Seq=114801 Ack=0 win=131072 Len=1400 Tsva=2713 TSecr=2708
132	2.715200	172.0.1.1	160.0.0.1	TCP	52	80->49153 [ACK] Seq=0 Ack=114801 win=129672 Len=0 Tsva=2716 TSecr=2709
133	2.717000	160.0.0.1	172.0.1.1	TCP	1452	49153->80 [ACK] Seq=116201 Ack=0 win=131072 Len=1400 Tsva=2717 TSecr=2708
134	2.721200	172.0.1.1	160.0.0.1	TCP	52	80->49153 [ACK] Seq=0 Ack=117601 win=129672 Len=0 Tsva=2722 TSecr=2717
135	2.870875	160.0.0.1	172.0.1.1	TCP	1452	49153->80 [ACK] Seq=117601 Ack=0 win=131072 Len=1400 Tsva=2874 TSecr=2722
136	2.876000	160.0.0.1	172.0.1.1	TCP	1452	49153->80 [ACK] Seq=119001 Ack=0 win=131072 Len=1400 Tsva=2874 TSecr=2722
137	2.879000	160.0.0.1	172.0.1.1	TCP	1452	49153->80 [ACK] Seq=120401 Ack=0 win=131072 Len=1400 Tsva=2878 TSecr=2722
138	2.880200	172.0.1.1	160.0.0.1	TCP	52	80->49153 [ACK] Seq=0 Ack=120401 win=129672 Len=0 Tsva=2881 TSecr=2874
139	2.882000	160.0.0.1	172.0.1.1	TCP	1452	49153->80 [ACK] Seq=121801 Ack=0 win=131072 Len=1400 Tsva=2882 TSecr=2722
140	2.886200	172.0.1.1	160.0.0.1	TCP	52	80->49153 [ACK] Seq=0 Ack=123201 win=129672 Len=0 Tsva=2887 TSecr=2882
141	2.887125	160.0.0.1	172.0.1.1	TCP	1452	49153->80 [ACK] Seq=123201 Ack=0 win=131072 Len=1400 Tsva=2883 TSecr=2722
142	2.890125	160.0.0.1	172.0.1.1	TCP	1452	49153->80 [ACK] Seq=124601 Ack=0 win=131072 Len=1400 Tsva=2889 TSecr=2881
143	2.894325	172.0.1.1	160.0.0.1	TCP	52	80->49153 [ACK] Seq=0 Ack=126001 win=129672 Len=0 Tsva=2896 TSecr=2889
144	2.998325	172.0.1.1	160.0.0.1	TCP	52	80->49153 [FIN, ACK] Seq=0 Ack=126001 win=131072 Len=0 Tsva=3000 TSecr=28
145	3.001500	160.0.0.1	172.0.1.1	TCP	52	49153->80 [FIN, ACK] Seq=126001 Ack=0 win=131072 Len=0 Tsva=3000 TSecr=28
146	3.004500	160.0.0.1	172.0.1.1	TCP	52	49153->80 [ACK] Seq=126002 Ack=1 win=131072 Len=0 Tsva=3004 TSecr=3000
147	3.005575	172.0.1.1	160.0.0.1	TCP	52	80->49153 [ACK] Seq=1 Ack=126002 win=131072 Len=0 Tsva=3007 TSecr=3000

Obr. 7.3: Wireshark – záznam komunikace na WebServer.

Princip přenosu pomocí TCP protokolu spoléhá zpočátku na tzv. Three-Way Handshake, který slouží pro navázání spojení mezi entitami (pakety 1, 2 a 3). Ukončení komunikace představují pakety číslo 144–147, kdy v rámci prostředí NS-3 ukončují spojení obě komunikující strany zároveň. Reálně je iniciátorem ukončení spojení klient, kdy jako první odešle zprávu FIN, na kterou server odpoví zprávou ACK. Poté server odesílá vlastní zprávu FIN a očekává potvrzení zprávou ACK od klienta.

7.3.2 PPV

Čas simulace 4–5,5 s.

Služba Pay per View poskytuje divákovi možnost shlédnout jednorázovou událost, kterou nemá předplacenu nebo která je nabízena mimo nabídku. Hlavními částmi této služby je její objednání v časovém odstupu před vysíláním a samotné vysílání.

Objednání služby v rámci této simulace probíhá pomocí televizního průvodce (může probíhat také skrze operátora nebo automatizovaný telefonní systém). Na obr. 7.4 je možno v čase 4–4,5s vidět TCP komunikaci mezi klientem (Client_1) a TvServerem, který obsluhuje objednávky těchto služeb na portu 80. Po objednání vybrané události je klient zařazen do seznamu uživatelů, ke kterým má být přenos v daný vysílací čas doručen.

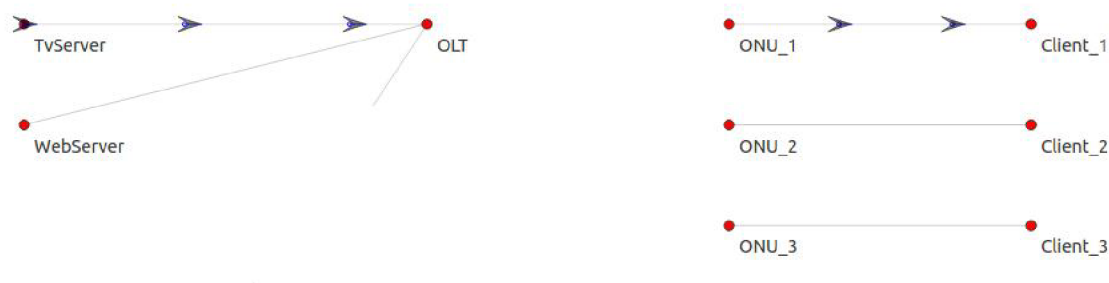
No.	Time	Source	Destination	Protocol	Length	Info
148	3.999875	160.0.0.1	172.0.0.1	TCP	56	[TCP Port numbers reused] 49154->80 [SYN] Seq=4294966272 Win=32768 Len=0 W
149	4.003950	172.0.0.1	160.0.0.1	TCP	56	80->49154 [SYN, ACK] Seq=4294966273 Ack=4294966273 Win=32768 Len=0 WS=4 TS
150	4.011375	160.0.0.1	172.0.0.1	TCP	52	49154->80 [ACK] Seq=4294966273 Ack=0 Win=131072 Len=0 TSval=4010 TSecr=400
151	4.054750	160.0.0.1	172.0.0.1	TCP	1076	49154->80 [ACK] Seq=4294966273 Ack=0 Win=131072 Len=1024 TSval=4054 TSecr=
152	4.058950	172.0.0.1	160.0.0.1	TCP	52	80->49154 [ACK] Seq=0 Ack=1 Win=130048 Len=0 TSval=4060 TSecr=4054
153	4.066250	160.0.0.1	172.0.0.1	IPV4	1500	Fragmented IP protocol (protocol 6, offset 0, ID=0003) [reassembled in #154
154	4.066250	160.0.0.1	172.0.0.1	TCP	72	49154->80 [ACK] Seq=1 Ack=0 Win=131072 Len=1500 TSval=4065 TSecr=4060
155	4.066250	160.0.0.1	172.0.0.1	TCP	600	49154->80 [ACK] Seq=1501 Ack=0 Win=131072 Len=548 TSval=4065 TSecr=4060
156	4.070450	172.0.0.1	160.0.0.1	TCP	52	80->49154 [ACK] Seq=0 Ack=2049 Win=130524 Len=0 TSval=4072 TSecr=4065
157	4.077750	160.0.0.1	172.0.0.1	TCP	1076	49154->80 [ACK] Seq=2049 Ack=0 Win=131072 Len=1024 TSval=4076 TSecr=4072
158	4.121125	160.0.0.1	172.0.0.1	TCP	1076	49154->80 [ACK] Seq=3073 Ack=0 Win=131072 Len=1024 TSval=4120 TSecr=4072
159	4.125325	172.0.0.1	160.0.0.1	TCP	52	80->49154 [ACK] Seq=0 Ack=4097 Win=130048 Len=0 TSval=4127 TSecr=4120
160	4.126250	160.0.0.1	172.0.0.1	TCP	1076	49154->80 [ACK] Seq=4097 Ack=0 Win=131072 Len=1024 TSval=4124 TSecr=4072
161	4.129250	160.0.0.1	172.0.0.1	TCP	1076	49154->80 [ACK] Seq=5121 Ack=0 Win=131072 Len=1024 TSval=4128 TSecr=4072
162	4.133450	172.0.0.1	160.0.0.1	TCP	52	80->49154 [ACK] Seq=0 Ack=6145 Win=130048 Len=0 TSval=4135 TSecr=4128
163	4.134375	160.0.0.1	172.0.0.1	TCP	1076	49154->80 [ACK] Seq=6145 Ack=0 Win=131072 Len=1024 TSval=4132 TSecr=4127
200	4.485625	160.0.0.1	172.0.0.1	TCP	1076	49154->80 [ACK] Seq=31745 Ack=0 Win=131072 Len=1024 TSval=4485 TSecr=4424
201	4.489825	172.0.0.1	160.0.0.1	TCP	52	80->49154 [ACK] Seq=0 Ack=32769 Win=130048 Len=0 TSval=4491 TSecr=4485
202	4.490750	160.0.0.1	172.0.0.1	TCP	1076	49154->80 [ACK] Seq=32769 Ack=0 Win=131072 Len=1024 TSval=4489 TSecr=4483
203	4.500125	160.0.0.1	172.0.0.1	TCP	52	49154->80 [FIN, ACK] Seq=33793 Ack=0 Win=131072 Len=0 TSval=4500 TSecr=449
204	4.504200	172.0.0.1	160.0.0.1	TCP	52	80->49154 [FIN, ACK] Seq=0 Ack=33794 Win=131072 Len=0 TSval=4505 TSecr=450
205	4.511625	160.0.0.1	172.0.0.1	TCP	52	49154->80 [ACK] Seq=33794 Ack=1 Win=131072 Len=0 TSval=4510 TSecr=4505
206	4.999200	172.0.0.1	160.0.0.1	UDP	1052	Source port: 49153 Destination port: 6970
207	4.999950	172.0.0.1	160.0.0.1	UDP	1052	Source port: 49153 Destination port: 6970
208	5.000825	172.0.0.1	160.0.0.1	UDP	1052	Source port: 49153 Destination port: 6970
209	5.001575	172.0.0.1	160.0.0.1	UDP	1052	Source port: 49153 Destination port: 6970
210	5.002450	172.0.0.1	160.0.0.1	UDP	1052	Source port: 49153 Destination port: 6970
211	5.003200	172.0.0.1	160.0.0.1	UDP	1052	Source port: 49153 Destination port: 6970
212	5.004075	172.0.0.1	160.0.0.1	UDP	1052	Source port: 49153 Destination port: 6970

Obr. 7.4: Wireshark – záznam PPV komunikace.

Na obr. 7.4 je dále možno vidět, že objednaná událost je streamována ke klientovi od času 5 s. V rámci této simulace je použit jako cílový port 6970, pro který nebyl organizací IANA definován používaný protokol. Reálný přenos ovšem využívá RTP (Real-time Transport Protocol) protokolu. Pro tento protokol není port rezervován. Důvodem je, že port je sjednáván dynamicky za pomoci SDP (Session Description Protocol) protokolu, který se používá pro popis spojení.

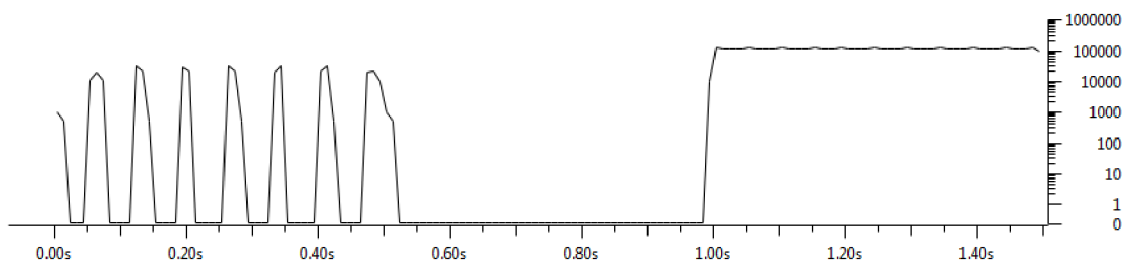
Pro zobrazení reálného RTP protokolu ve Wiresharku je nutno provést změny, které jsou zmíněny v kapitole 7.3.6.

Grafické zobrazení komunikace je možno vidět na obr. 7.5. Obrázek zachycuje pakety vysílané z TvServeru na klienta Client_1.



Obr. 7.5: PPV stream.

Graf propustnosti v části simulace PPV je možno vidět na obr. 7.6. Jedná se pouze o čas běhu simulace v rámci PPV, proto je časové označení horizontální osy 0–1,5 s. Vertikální osa je kvantována v bit/tick a tick = 0,01 s. Komunikace 0–0,5 s představuje objednání služby a čas 1–1,5 s zobrazuje stream ze serveru ke klientovi. Pro lepší zobrazení části, kdy komunikace funguje na modelu klient-server byla zvolena logaritmická vertikální osa, která je příčinou chybějící počáteční části grafu.



Obr. 7.6: Throughput PPV pomocí Wiresharku.

7.3.3 VoD

Čas simulace 6–7 s.

Služba Video on Demand je v rámci této práce demonstrována způsobem, kdy je vyžádaný obsah streamován ze serveru směrem ke klientovi.

Prvním bodem reálné komunikace i této části simulace je výběr obsahu, který klient požaduje. Pro výběr z nabídky se klient připojí na server, se kterým komunikuje modelem klient-server. Zachycenou komunikaci zobrazuje obr. 7.7, kde pakety číslo 816–826 představují tři fáze komunikace, a to její sestavení, průběh a ukončení.

Druhým bodem je samotný přenos obsahu – stream. Na rozdíl od PPV dochází ke streamování vybraného obsahu okamžitě po objednání. Na již zmíněném obr. 7.7 jsou zobrazeny také zachycené pakety č. 827–829 navazující unicastové spojení mezi serverem a klientem.

No.	Time	Source	Destination	Protocol	Length	Info
816	6.004000	160.0.1.1	172.0.0.1	TCP	56	TCP Port numbers reused 49153-80 [SYN] Seq=4294966784 Win=32768 Len=0
817	6.003075	172.0.0.1	160.0.1.1	TCP	56	80->49153 [SYN, ACK] Seq=4294967295 Ack=4294966785 Win=32768 Len=0 WS=4 TS
818	6.012500	160.0.1.1	172.0.0.1	TCP	52	49153->80 [ACK] Seq=4294966785 Ack=0 Win=131072 Len=0 TSval=6011 TSecr=600
819	6.068625	160.0.1.1	172.0.0.1	TCP	564	49153->80 [ACK] Seq=4294966785 Ack=0 Win=131072 Len=512 TSval=6067 TSecr=6
820	6.072700	160.0.1.1	160.0.1.1	TCP	52	80->49153 [ACK] Seq=0 Ack=1 Win=130560 Len=0 TSval=6074 TSecr=6067
821	6.116250	160.0.1.1	172.0.0.1	TCP	564	49153->80 [ACK] Seq=1 Ack=0 Win=131072 Len=512 TSval=6114 TSecr=6074
822	6.183000	160.0.1.1	172.0.0.1	TCP	564	49153->80 [ACK] Seq=513 Ack=0 Win=131072 TSval=6181 TSecr=6074
823	6.187075	172.0.0.1	160.0.1.1	TCP	52	80->49153 [ACK] Seq=0 Ack=1025 Win=130560 Len=0 TSval=6188 TSecr=6181
824	6.200875	160.0.1.1	172.0.0.1	TCP	52	49153->80 [FIN, ACK] Seq=1025 Ack=0 Win=131072 Len=0 TSval=6200 TSecr=6188
825	6.204950	172.0.0.1	160.0.1.1	TCP	52	80->49153 [FIN, ACK] Seq=0 Ack=1026 Win=131072 Len=0 TSval=6206 TSecr=6200
826	6.212375	160.0.1.1	172.0.0.1	TCP	52	49153->80 [ACK] Seq=1026 Ack=1 Win=131072 Len=0 TSval=6211 TSecr=6206
827	6.218325	172.0.0.1	160.0.1.1	TCP	56	TCP Port numbers reused 49153-6000 [SYN] Seq=4294966784 Win=32768 Len=0
828	6.226000	160.0.1.1	172.0.0.1	TCP	56	6000->49153 [SYN, ACK] Seq=4294967295 Ack=4294966785 Win=32768 Len=0 WS=4
829	6.230075	172.0.0.1	160.0.1.1	TCP	52	49153->6000 [ACK] Seq=4294966785 Ack=0 Win=131072 Len=0 TSval=6231 TSecr=6
830	6.424500	160.0.1.1	172.0.0.1	UDP	328	Source port: 49153 Destination port: 554
831	6.429700	172.0.0.1	160.0.1.1	X11	564	49153->6000 [ACK] Seq=4294966785 Ack=0 Win=131072 Len=512 TSval=6431 TSecr
832	6.430075	172.0.0.1	160.0.1.1	X11	564	49153->6000 [ACK] Seq=1 Ack=0 Win=131072 Len=512 TSval=6431 TSecr=6224
833	6.436000	160.0.1.1	172.0.0.1	TCP	52	6000->49153 [ACK] Seq=0 Ack=1 Win=130560 Len=0 TSval=6435 TSecr=6431
834	6.440200	172.0.0.1	160.0.1.1	IPv4	1500	Fragmented IP protocol (proto=TCP 6, off=0, ID=0008) [Reassembled in #835
835	6.440200	172.0.0.1	160.0.1.1	X11	72	49153->6000 [ACK] Seq=513 Ack=0 Win=131072 Len=1500 TSval=6441 TSecr=6435
836	6.447500	160.0.1.1	172.0.0.1	TCP	52	6000->49153 [ACK] Seq=0 Ack=2013 Win=129572 Len=0 TSval=6446 TSecr=6441
837	6.451700	172.0.0.1	160.0.1.1	IPv4	1500	Fragmented IP protocol (proto=TCP 6, off=0, ID=0009) [Reassembled in #838
838	6.451700	172.0.0.1	160.0.1.1	X11	72	49153->6000 [ACK] Seq=2013 Ack=0 Win=131072 Len=1500 TSval=6453 TSecr=6446

Obr. 7.7: Wireshark – záznam VoD komunikace 1.

Spojení v tomto případě navazuje server, který následně čeká na příchozí paket se zprávou play pro zahájení streamování. Reálnou komunikaci ovšem navazuje klient, za pomoci TCP transportního protokolu, díky jemuž má také prostřednictvím RTSP (Real Time Streaming Protocol) protokolu možnost vysílaný stream ovlivňovat příkazy podobnými příkazům přehrávače jako například play, pause atd. Pro co nejněvčetnější zobrazení principu ovládní RTSP protokolem je v rámci simulace ze strany klienta použita komunikace pomocí protokolu UDP namísto reálně používaného TCP. Příkaz play pro zahájení streamování je zachycen jako paket s pořadovým číslem 830. Zde se projeví důvod volby navázání komunikace ze strany serveru. Pokud by bylo zvoleno reálné navázání komunikace ze strany klienta, došlo by po obdržení výše zmíněného paketu k vysílání paketů ze strany klienta k serveru, což je v rámci této simulace nežádoucí. Tato situace je dále zmíněna v kapitole 7.3.6.

Na následujícím obr. 7.8 je zobrazena část zachycené komunikace, kdy se klient rozhodl pozastavit streamování, vyslal tedy paket s číslem 1350 s příkazem pause.

No.	Time	Source	Destination	Protocol	Length	Info
1349	6.615325	172.0.0.1	160.0.1.1	X11	564	49153->6000 [ACK] Seq=231425 Ack=0 Win=131072 Len=512 TSval=6616 TSecr=661
1350	6.615375	160.0.1.1	172.0.0.1	UDP	328	Source port: 49153 Destination port: 554
1351	6.615375	160.0.1.1	172.0.0.1	TCP	52	6000->49153 [ACK] Seq=0 Ack=224769 Win=130560 Len=0 TSval=6615 TSecr=6611
1352	6.615700	172.0.0.1	160.0.1.1	X11	564	49153->6000 [ACK] Seq=231937 Ack=0 Win=131072 Len=512 TSval=6617 TSecr=661
1353	6.616075	172.0.0.1	160.0.1.1	X11	564	49153->6000 [ACK] Seq=232449 Ack=0 Win=131072 Len=512 TSval=6617 TSecr=661
1354	6.616250	160.0.1.1	172.0.0.1	TCP	52	6000->49153 [ACK] Seq=0 Ack=225793 Win=130560 Len=0 TSval=6616 TSecr=6612
1355	6.616450	172.0.0.1	160.0.1.1	X11	564	49153->6000 [ACK] Seq=232961 Ack=0 Win=131072 Len=512 TSval=6618 TSecr=661
1356	6.616950	172.0.0.1	160.0.1.1	X11	564	49153->6000 [ACK] Seq=233473 Ack=0 Win=131072 Len=512 TSval=6618 TSecr=661
1357	6.617125	160.0.1.1	172.0.0.1	TCP	52	6000->49153 [ACK] Seq=0 Ack=226817 Win=130560 Len=0 TSval=6617 TSecr=6612
1358	6.617325	172.0.0.1	160.0.1.1	X11	564	49153->6000 [ACK] Seq=233985 Ack=0 Win=131072 Len=512 TSval=6619 TSecr=661
1359	6.617700	172.0.0.1	160.0.1.1	X11	564	49153->6000 [ACK] Seq=234497 Ack=0 Win=131072 Len=512 TSval=6619 TSecr=661
1360	6.618000	160.0.1.1	172.0.0.1	TCP	52	6000->49153 [ACK] Seq=0 Ack=227841 Win=130560 Len=0 TSval=6618 TSecr=6613
1361	6.618075	172.0.0.1	160.0.1.1	X11	564	49153->6000 [ACK] Seq=235009 Ack=0 Win=131072 Len=512 TSval=6619 TSecr=661
1362	6.618875	160.0.1.1	172.0.0.1	TCP	52	6000->49153 [ACK] Seq=0 Ack=228865 Win=130560 Len=0 TSval=6619 TSecr=6614
1363	6.619750	160.0.1.1	172.0.0.1	TCP	52	6000->49153 [ACK] Seq=0 Ack=229889 Win=130560 Len=0 TSval=6619 TSecr=6615
1364	6.620625	160.0.1.1	172.0.0.1	TCP	52	6000->49153 [ACK] Seq=0 Ack=230913 Win=130560 Len=0 TSval=6620 TSecr=6616
1365	6.621500	160.0.1.1	172.0.0.1	TCP	52	6000->49153 [ACK] Seq=0 Ack=231937 Win=130560 Len=0 TSval=6621 TSecr=6616
1366	6.622375	160.0.1.1	172.0.0.1	TCP	52	6000->49153 [ACK] Seq=0 Ack=232961 Win=130560 Len=0 TSval=6622 TSecr=6617
1367	6.623250	160.0.1.1	172.0.0.1	TCP	52	6000->49153 [ACK] Seq=0 Ack=233985 Win=130560 Len=0 TSval=6623 TSecr=6618
1368	6.624125	160.0.1.1	172.0.0.1	TCP	52	6000->49153 [ACK] Seq=0 Ack=235009 Win=130560 Len=0 TSval=6623 TSecr=6619
1369	6.824750	160.0.1.1	172.0.0.1	UDP	328	Source port: 49154 Destination port: 554
1370	6.824750	160.0.1.1	172.0.0.1	TCP	52	6000->49153 [ACK] Seq=0 Ack=235521 Win=131072 Len=0 TSval=6824 TSecr=6619

Obr. 7.8: Wireshark – záznam VoD komunikace 2.

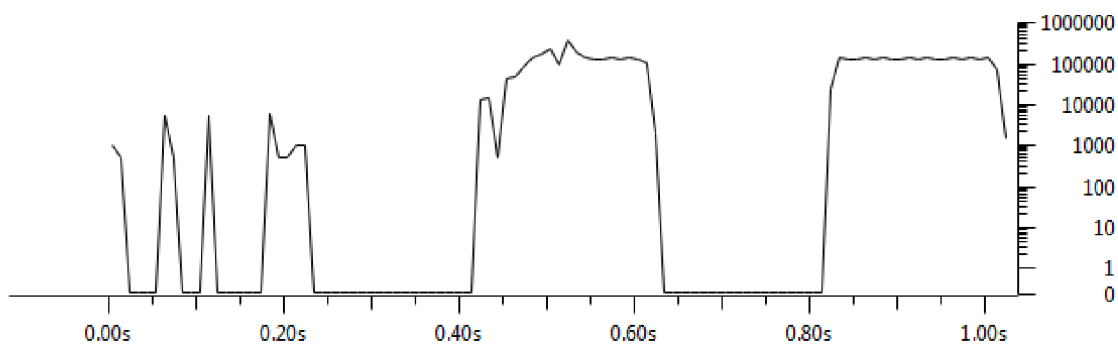
Z obr. 7.8 je patrné, že po zachycení paketu bylo ze serveru odesláno ještě dalších několik paketů. V potaz ovšem musí být bráno, že paket byl zachycen na rozhraní jednotky OLT v rámci sítě XG-PON a bude tedy ještě 2 ms trvat, než je přijat serverem, kde musí být následně zpracován a až poté je stream přerušen. Na zmíněném obrázku je také zobrazen další paket vyslaný klientem pomocí UDP protokolu, kdy se klient rozhodne pokračovat ve sledování streamovaného obsahu.

Průběh komunikace zobrazený pomocí doprovodného programu NetAnim je k vidění na obr. 7.9, kde server odesílá pakety, jejichž přijetí klient potvrzuje.



Obr. 7.9: VoD stream.

Vytížení linky během simulace VoD je zobrazeno na obr. 7.10. V první části grafu je vykresleno vytížení během objednání obsahu následované vytížením během samotného streamování, kde je viditelné nulové vytížení v době, kdy je stream pozastaven. Obdobně jako na obr. 7.6 se horizontální časová osa vztahuje pouze k dané části simulace, vertikální osa je kvantována v bit/tick a tick = 0,01 s.



Obr. 7.10: Throughput VoD pomocí Wiresharku.

7.3.4 BitTorrent

Čas simulace 9,1–12 s.

Pro simulaci BitTorrent protokolu v prostředí NS-3 neexistuje ve verzi 3.21 již vytvořený modul. Simulace funkce tohoto protokolu je proto realizována větším množstvím OnOff aplikací představujících zdroje/cíle komunikace, jejichž popis je obsažen v kapitole 6.

Simulace z velké části využívá internetovou část topologie, pro zobrazení principu a výsledných dat jsou proto pomocí níže uvedeného kódu generovány výstupy SourceA-pcap-9-1.pcap a SourceB-pcap-9-2.pcap.

```
NetDeviceContainer SourceAtrack ;
SourceAtrack . Add ( p2pOutDev1 [ 0 ] . Get ( 0 ) ) ;
NetDeviceContainer SourceBtrack ;
SourceBtrack . Add ( p2pOutDev1 [ 1 ] . Get ( 0 ) ) ;

pointToPoint . EnablePcap ( " SourceA-pcap " , SourceAtrack ) ;
pointToPoint . EnablePcap ( " SourceB-pcap " , SourceBtrack ) ;
```

Před samotným začátkem této části simulace jsou uzly podílející se na komunikaci inicializovány takto:

- Source_A – seed.
- Source_B – leech – vlastní velkou část dat.
- Source_C – leech – vlastní malou část dat.

Prvním krokem ze strany klienta pro připojení k peer-to-peer sdílení je nutnost získu *.torrent* souboru obsahujícího metadata k požadovanému souboru. Zisk metadata je prezentován v simulaci komunikací klienta (Client_3) na torrentový server (Torr-Server) v rozmezí 9,1–9,9 s. V tomto časovém rozmezí dojde k navázání komunikace mezi klientem a serverem v modelu klient-server, k zisku metadata a zisku seznamu peerů z trackeru. V internetové části simulace již mezi sebou komunikují aktivní uzly Source_A a Source_B, kdy seed poskytuje data leech uzlu Source_B. V této fázi získává klient z trackeru tato data:

- Source_A – IP: 100.0.0.2, port: 6882.
- Source_B – IP: 110.0.0.2, port: 6883.

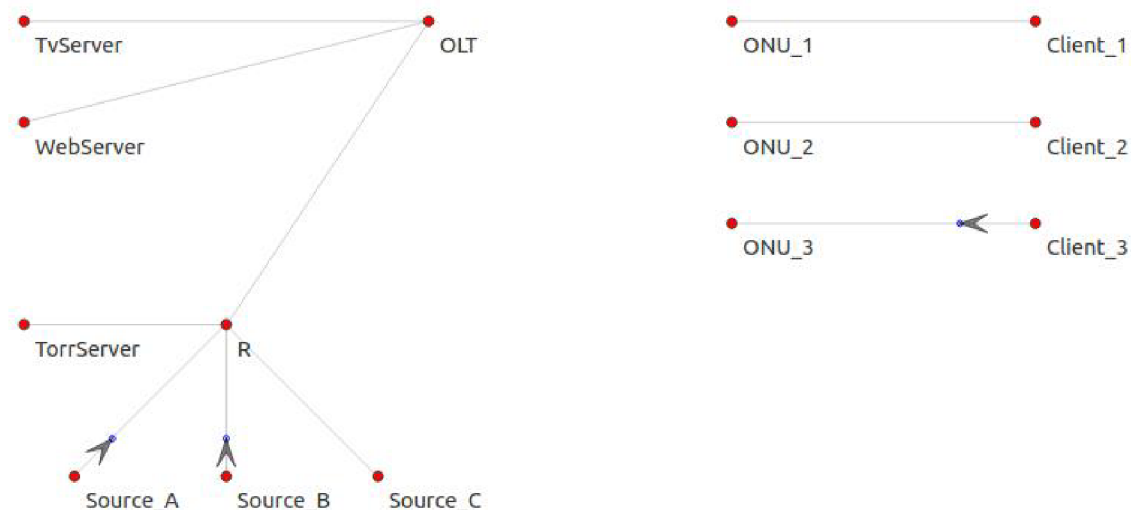
Po zisku všech dat a ukončení komunikace s trackerem dochází ihned k sestavení spojení s dostupnými peery, jak dokumentuje obr. 7.11.

Po sestavení spojení dochází k vysílání žádostí o bloky, které peeri vlastní. Klient v této chvíli získává bloky od dvou peerů.

No.	Time	Source	Destination	Protocol	Length	Info
2079	9.768500	160.0.2.1	55.55.55.2	TCP	1452	49153->6881 [ACK] Seq=14001 Ack=0 win=131072 Len=1400 TSval=9767 TSecr=9601
2080	9.769575	55.55.55.2	160.0.2.1	TCP	52	6881->49153 [ACK] Seq=0 Ack=14001 win=129672 Len=0 TSval=9769 TSecr=9761
2081	9.773625	160.0.2.1	55.55.55.2	TCP	1452	49153->6881 [ACK] Seq=15401 Ack=0 win=131072 Len=1400 TSval=9772 TSecr=9601
2082	9.778750	160.0.2.1	55.55.55.2	TCP	1452	49153->6881 [ACK] Seq=16801 Ack=0 win=131072 Len=1400 TSval=9778 TSecr=9761
2083	9.781950	55.55.55.2	160.0.2.1	TCP	52	6881->49153 [ACK] Seq=0 Ack=16801 win=129672 Len=0 TSval=9781 TSecr=9772
2084	9.900750	160.0.2.1	55.55.55.2	TCP	52	49153->6881 [FIN, ACK] Seq=18201 Ack=0 win=131072 Len=0 TSval=9900 TSecr=9901
2085	9.908825	55.55.55.2	160.0.2.1	TCP	52	6881->49153 [FIN, ACK] Seq=0 Ack=18202 win=131072 Len=0 TSval=9908 TSecr=9901
2086	9.916500	160.0.2.1	55.55.55.2	TCP	52	49153->6881 [ACK] Seq=18202 Ack=1 win=131072 Len=0 TSval=9915 TSecr=9908
2087	10.000250	160.0.2.1	100.0.0.2	TCP	56	[TCP Port numbers reused] 49154->6882 [SYN] Seq=4294966273 win=32768 Len=0
2088	10.000250	160.0.2.1	110.0.0.2	TCP	56	[TCP Port numbers reused] 49155->6883 [SYN] Seq=4294966272 win=32768 Len=0
2089	10.008325	100.0.0.2	160.0.2.1	TCP	56	6882->49154 [SYN, ACK] Seq=0 Ack=4294966273 win=32768 Len=0 NS=4 TSval=1000
2090	10.008325	110.0.0.2	160.0.2.1	TCP	56	6883->49155 [SYN, ACK] Seq=0 Ack=4294966273 win=32768 Len=0 NS=4 TSval=1000
2091	10.016000	160.0.2.1	100.0.0.2	TCP	52	49154->6882 [ACK] Seq=4294966273 Ack=1 win=131072 Len=0 TSval=10014 TSecr=:
2092	10.016000	160.0.2.1	110.0.0.2	TCP	52	49155->6883 [ACK] Seq=4294966273 Ack=1 win=131072 Len=0 TSval=10014 TSecr=:
2093	10.033875	160.0.2.1	100.0.0.2	TCP	1076	49154->6882 [ACK] Seq=4294966273 Ack=1 win=131072 Len=1024 TSval=10032 TSecr=:
2094	10.042075	100.0.0.2	160.0.2.1	TCP	52	6882->49154 [ACK] Seq=1 Ack=1 win=130048 Len=0 TSval=10041 TSecr=10032
2095	10.043375	160.0.2.1	110.0.0.2	TCP	1076	49155->6883 [ACK] Seq=4294966273 Ack=1 win=131072 Len=1024 TSval=10045 TSecr=:
2096	10.053375	110.0.0.2	160.0.2.1	TCP	52	6883->49155 [ACK] Seq=1 Ack=1 win=130048 Len=0 TSval=10053 TSecr=10045
2097	10.053375	160.0.2.1	100.0.0.2	TCP	1076	49154->6882 [ACK] Seq=1 Ack=1 win=131072 Len=1024 TSval=10065 TSecr=10041
2098	10.091750	160.0.2.1	110.0.0.2	TCP	1076	49155->6883 [ACK] Seq=1 Ack=1 win=131072 Len=1024 TSval=10091 TSecr=10053

Obr. 7.11: Wireshark – záznam BitTorrent komunikace.

V čase simulace 10,2s dochází k pravidelné aktualizaci dat na tracker, jak ilustruje obr. 7.12. V rámci simulace byla zvolena hromadná aktualizace pro všechny nyní aktivní uživatele.



Obr. 7.12: Aktualizace na tracker.

Po této aktualizaci dochází k sestavení komunikace mezi uzly Source_B a Client_3, kdy uzel Source_B obdržel informaci o dostupnosti klienta:

- Client_3 – IP: 160.0.2.1, port 6885.

Komunikace ve směru Source_B → Client_3 ovšem není příliš aktivní, z důvodu vlastnictví malého počtu rozdílných bloků ze strany uzlu Client_3.

V čase simulace 10,4s dochází k opětovnému připojení uzlu Source_C, který získává z trackeru nejnovější informace o stavu dalších peerů a předává své aktuální informace. Po obdržení informací o uzlech Source_A, Source_B a Client_3 dochází k navázání spojení s těmito uzly a zisku dalších bloků. Po následné hromadné aktualizaci (čas 10,44s) dochází k sestavení spojení s uzlem Source_C (IP: 120.0.0.2,

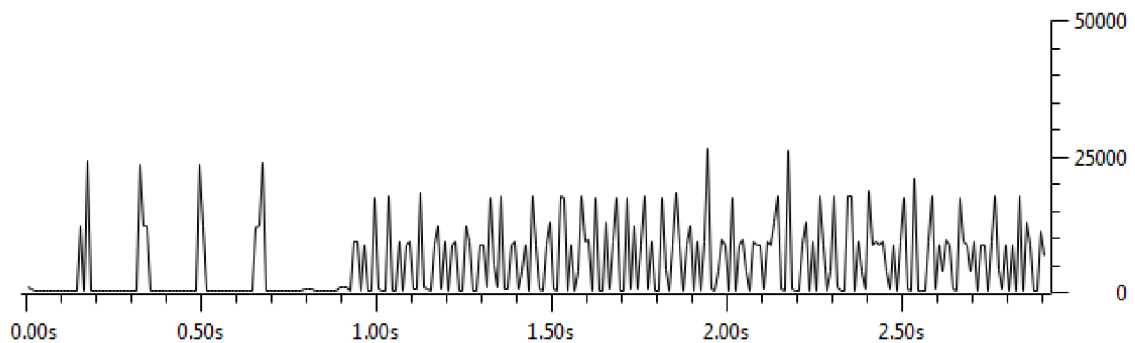
port: 6884) také ze strany uzlů Client_3 a Source_B, kteří obdrželi informaci o aktivitě tohoto uzlu. V tuto chvíli jsou všechny uzly aktivní a tvoří swarm.

Dle prvotní inicializace vlastnil uzel Source_B již při začátku této části simulace velkou část dat, proto se z tohoto leechce po zisku všech bloků v čase 11,5s stává seeder. Tuto chvíli, kdy dochází k ukončení komunikací mezi tímto uzlem ve směru downloadu, je možno vidět na obr. 7.13, jehož zdrojem je soubor SourceB-pcap-9-2, kde časová značka 2s znamená čas 11,5s jak je vysvětleno v kapitole 7.3.6.

No.	Time	Source	Destination	Protocol	Length	Info
177	1.966162	110.0.0.2	100.0.0.2	TCP	1078	49153->6882 [ACK] Seq=34817 Ack=0 win=131072 Len=1024 TSval=11466 TSecr=11467
179	1.967735	120.0.0.2	110.0.0.2	TCP	1078	49156->6883 [ACK] Seq=17409 Ack=0 win=131072 Len=1024 TSval=11467 TSecr=11468
180	1.971826	110.0.0.2	120.0.0.2	TCP	54	6883->49156 [ACK] Seq=0 Ack=18433 win=130048 Len=0 TSval=11471 TSecr=11467
181	1.980032	110.0.0.2	55.55.55.2	TCP	454	49154->6969 [ACK] Seq=5601 Ack=0 win=131072 Len=400 TSval=11479 TSecr=11409
182	2.000000	110.0.0.2	100.0.0.2	TCP	54	49153->6882 [FIN, ACK] Seq=35841 Ack=0 win=131072 Len=0 TSval=11500 TSecr=11500
183	2.000004	110.0.0.2	160.0.2.1	TCP	54	49153->6885 [FIN, ACK] Seq=5121 Ack=0 win=131072 Len=0 TSval=11500 TSecr=11500
184	2.004008	100.0.0.2	110.0.0.2	TCP	54	6882->49153 [FIN, ACK] Seq=0 Ack=35842 win=131072 Len=0 TSval=11504 TSecr=11504
185	2.005597	160.0.2.1	110.0.0.2	TCP	1078	49153->6883 [ACK] Seq=31745 Ack=0 win=131072 Len=1024 TSval=11501 TSecr=11501
186	2.008017	110.0.0.2	100.0.0.2	TCP	54	49153->6882 [ACK] Seq=35842 Ack=1 win=131072 Len=0 TSval=11508 TSecr=11504
187	2.009597	110.0.0.2	160.0.2.1	TCP	54	6883->49155 [ACK] Seq=0 Ack=32769 win=130048 Len=0 TSval=11509 TSecr=11501
188	2.010550	160.0.2.1	110.0.0.2	TCP	54	6885->49155 [FIN, ACK] Seq=0 Ack=5122 win=131072 Len=0 TSval=11506 TSecr=11506
189	2.014558	110.0.0.2	160.0.2.1	TCP	54	49153->6885 [ACK] Seq=5122 Ack=1 win=131072 Len=0 TSval=11514 TSecr=11506
190	2.022348	120.0.0.2	110.0.0.2	TCP	1078	49156->6883 [ACK] Seq=18433 Ack=0 win=131072 Len=1024 TSval=11522 TSecr=11511
191	2.052382	160.0.2.1	110.0.0.2	TCP	1078	49155->6883 [ACK] Seq=32769 Ack=0 win=131072 Len=1024 TSval=11547 TSecr=11511
192	2.060032	110.0.0.2	55.55.55.2	TCP	454	49154->6969 [ACK] Seq=6001 Ack=0 win=131072 Len=400 TSval=11559 TSecr=11409
193	2.064109	55.55.55.2	110.0.0.2	TCP	54	6969->49154 [ACK] Seq=0 Ack=6401 win=130672 Len=0 TSval=11564 TSecr=11559
194	2.077048	120.0.0.2	110.0.0.2	TCP	1078	49156->6883 [ACK] Seq=19457 Ack=0 win=131072 Len=1024 TSval=11576 TSecr=11511

Obr. 7.13: Wireshark – ukončení downloadu ze strany uzlu Source_B.

Tato část simulace je ukončena ve stavu, kdy swarm obsahuje dva seedery a dva leechery. Průběh vytížení linky ze směrovače R směrem ke klientovi Client_3 během této části simulace je zobrazen na obr. 7.14, kde je v první části grafu zobrazeno vytížení při prohledávání torrent serveru a komunikaci na tracker, v druhé části je vidět vytížení během sdílení dat (download i upload). Obdobně jako u předešlých obrázků s vytížením se horizontální časová osa vztahuje pouze k dané části simulace, vertikální osa je kvantována v bit/tick a tick = 0,01 s.



Obr. 7.14: Vytížení linky mezi směrovačem R a OLT.

Srovnání s reálným vytížením je obsaženo rovněž v kapitole 7.3.6.

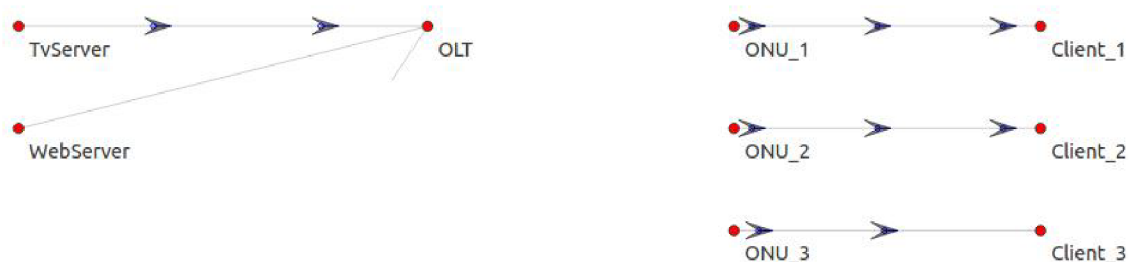
7.3.5 IPTV

Čas simulace 13–13,2 s.

Jednou z hlavních služeb v rámci Triple Play je IPTV, tedy televize přes internetový protokol. V simulaci je představeno její multicastové vysílání spolu s připojením a odpojením uživatele od skupiny pomocí IGMP protokolu.

V čase, kdy je tato část simulace započata dochází k multicastovému vysílání na dva koncové uzly, uživatel Client_3 není v tuto chvíli aktivní.

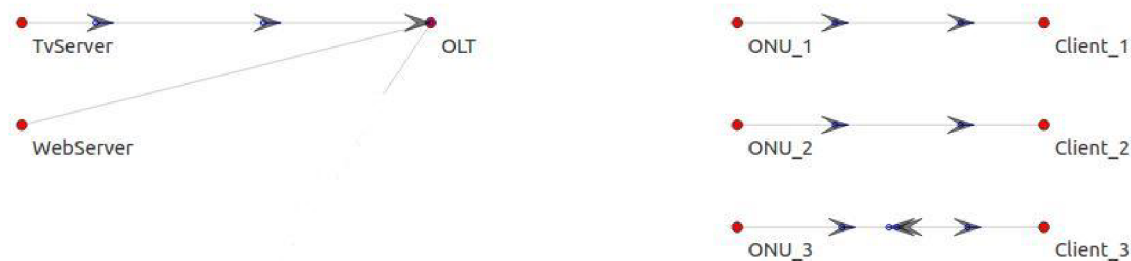
K připojení uživatele k multicastové skupině dochází v čase 13,035 s, kdy Client_3 vysílá zprávu Membership report pomocí IGMP protokolu. Tuto zprávu není možno v topologii simulace vidět, jelikož míří na lokální router (uzel Client_3). Tato zpráva má hodnotu pole TTL (Time to live) pouze 1, proto po zpracování lokálním routerem tento router žádá o příjem multicastového vysílání pomocí PIM (Protocol-Independent Multicast) protokolu. Paket s žádostí o multicastové vysílání je odeslán skrze přístupovou síť. Po zpracování tohoto paketu dochází k vysílání požadovaných dat na lokální router jak je ilustrováno na obr. 7.15.



Obr. 7.15: Počátek vysílání na uzel Client_3.

Pokud dojde ze strany uživatele k výběru příjmu jiného vysílání, musí dojít k odhlášení tohoto uživatele z multicastové skupiny a připojení k nové multicastové skupině. Tento moment ilustruje obr. 7.16, kde je možno pozorovat pakety směřující z uzlu klienta k jednotce ONU. Jejich zachycení pomocí programu Wireshark je viditelné na obr. 7.17. Zde představuje paket číslo 336 zprávu o odhlášení příjmu vysílání pomocí PIM protokolu (u IGMP Leave Group) a paket 337 zprávu s žádostí o příjem nového multicastového vysílání (u IGMP Membership report). Na této situaci můžeme sledovat hlavní nevýhodu IPTV vysílání, kdy po odeslání paketu se zprávou Membership report o přihlášení do nové multicastové skupiny trvá toto přepnutí až jednotky sekund (v rámci simulace vzhledem k délce pouze několik ms).

Ze zachycených paketů na obr. 7.17 je jasné, že klient provedl odhlášení z multicastové skupiny s adresou 225.10.10.50 a přihlášení na skupinu s adresou 225.10.10.51. Toto vysílání klient přijímá do doby simulace 13,073s, kdy odesílá zprávu Leave Group a stává se znovu neaktivním.



Obr. 7.16: Změna multicastové skupiny pomocí NetAnimu.

No.	Time	Source	Destination	Protocol	Length	Info
333	3.958295	160.0.2.2	225.10.10.50	UDP	1054	source port: 49153 destination port: 1000
334	3.959114	160.0.2.2	225.10.10.50	UDP	1054	source port: 49153 destination port: 1000
335	3.959933	160.0.2.2	225.10.10.50	UDP	1054	source port: 49153 destination port: 1000
336	3.960052	160.0.2.1	160.0.2.2	UDP	62	source port: 49154 destination port: 5555
337	3.960103	160.0.2.1	160.0.2.2	UDP	62	source port: 49154 destination port: 5555
338	3.960752	160.0.2.2	225.10.10.50	UDP	1054	source port: 49153 destination port: 1000
339	3.963815	160.0.2.2	225.10.10.51	UDP	1054	source port: 49154 destination port: 1000
340	3.964634	160.0.2.2	225.10.10.51	UDP	1054	source port: 49154 destination port: 1000
341	3.965453	160.0.2.2	225.10.10.51	UDP	1054	source port: 49154 destination port: 1000
342	3.966272	160.0.2.2	225.10.10.51	UDP	1054	source port: 49154 destination port: 1000
343	3.967091	160.0.2.2	225.10.10.51	UDP	1054	source port: 49154 destination port: 1000
344	3.967911	160.0.2.2	225.10.10.51	UDP	1054	source port: 49154 destination port: 1000
345	3.968730	160.0.2.2	225.10.10.51	UDP	1054	source port: 49154 destination port: 1000
346	3.969549	160.0.2.2	225.10.10.51	UDP	1054	source port: 49154 destination port: 1000
347	3.970368	160.0.2.2	225.10.10.51	UDP	1054	source port: 49154 destination port: 1000
348	3.971187	160.0.2.2	225.10.10.51	UDP	1054	source port: 49154 destination port: 1000
349	3.972007	160.0.2.2	225.10.10.51	UDP	1054	source port: 49154 destination port: 1000
350	3.972826	160.0.2.2	225.10.10.51	UDP	1054	source port: 49154 destination port: 1000
351	3.973141	160.0.2.1	160.0.2.2	UDP	62	source port: 49153 destination port: 5555
352	3.973645	160.0.2.2	225.10.10.51	UDP	1054	source port: 49154 destination port: 1000

Obr. 7.17: Wireshark – změna multicastové skupiny, zdroj: onu_3-pcap-3-1.pcap.

7.3.6 Srovnání zobrazených výsledků

V této podkapitole jsou pojednány rozdíly mezi získanými výsledky a reálnými daty a také některé z problémů, jež vyvstaly při tvorbě simulace.

Webové stránky

V rámci simulace komunikace na webové stránky byl použit na straně serveru port 80 pro HTTP protokol. V popisu komunikace je zmíněno, že se může jednat o komunikaci pro kontrolu zaplacení účtu za připojení. V tomto případě by ovšem komunikace probíhala na port 443, který je rezervovaný pro HTTPS (Hypertext Transfer Protocol Secure), kvůli nutnosti šifrování pro zabezpečení spojení mezi webovým prohlížečem a webovým serverem.

Pay per View

Pro stream v této části simulace byl použit jako cílový port číslo 6970. Tento port je z rozsahu nejpoužívanějších portů, které se používají (po domluvě obou stran) pro přenos pomocí RTP protokolu. Pro zobrazení použití RTP bylo v programu provedeno dekodování UDP jako RTP. Toto dekodování je prováděno zvolením paketu s UDP→Analyze→Decode As... a následnou volbou transportního protokolu RTP. Výsledkem této operace je obr. 7.18, kde je zobrazeno v položce Protocol RTP.

No.	Time	Source	Destination	Protocol	Length	Info
203	4.500125	160.0.0.1	172.0.0.1	TCP	52	49154→80 [FIN, ACK] Seq=33793 Ack=0 Win=131072 Len=0 TSval=4500 TSecr=4497
204	4.504200	172.0.0.1	160.0.0.1	TCP	52	80→49154 [FIN, ACK] Seq=0 Ack=33794 Win=131072 Len=0 TSval=4505 TSecr=4505
205	4.511625	160.0.0.1	172.0.0.1	TCP	52	49154→80 [ACK] Seq=33794 Ack=1 Win=131072 Len=0 TSval=4510 TSecr=4505
206	4.999200	172.0.0.1	160.0.0.1	RTP	1052	Unknown RTP version 0
207	4.999950	172.0.0.1	160.0.0.1	RTP	1052	Unknown RTP version 0
208	5.000825	172.0.0.1	160.0.0.1	RTP	1052	Unknown RTP version 0
209	5.001575	172.0.0.1	160.0.0.1	RTP	1052	Unknown RTP version 0
210	5.002450	172.0.0.1	160.0.0.1	RTP	1052	Unknown RTP version 0
211	5.003200	172.0.0.1	160.0.0.1	RTP	1052	Unknown RTP version 0

Obr. 7.18: Dekódování RTP protokolu.

V kódu bylo nutno vytvořit v této části aplikaci s názvem *PpvVodLoginSink*, která zasahuje svou dobou existence i do simulace VoD, z důvodu nemožnosti vytvořit dva sinky pro stejný typ komunikace na uzlu TvServer.

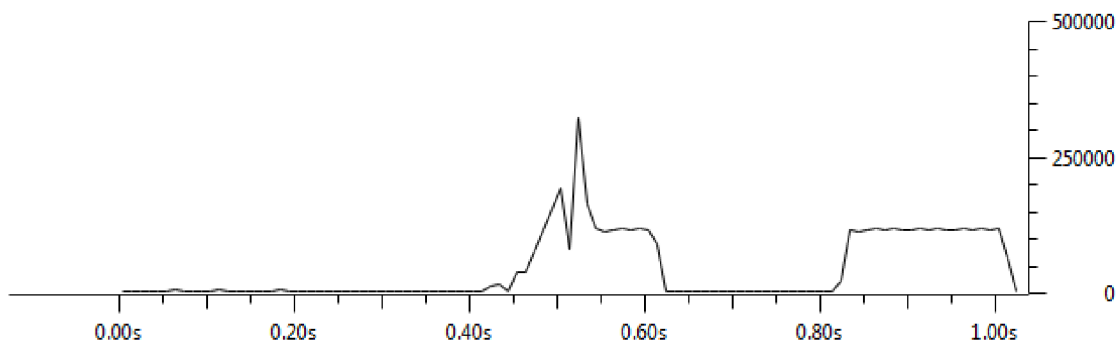
Video on Demand

V kapitole 7.3.3 je zmíněno, že TCP spojení pro streamování je iniciováno serverem z důvodu, aby v doprovodných programech bylo názorně vidět, že po obdržení RTSP paketu od klienta začíná odesílání paketů ze serveru. Tato situace je také ideální pro podotknutí, že v rámci OnOff aplikace jsou data vysílána směrem od iniciátora spojení k cíli, jak je patrné z obr. 7.19 a cíl potvrzuje přijetí paketů zprávou ACK. Tato aplikační vlastnost ovšem není vhodná při komunikaci kdy iniciátorem spojení je klient, který poté odesílá data na server, což je přesným opakem reálné situace. Tento jev je možno vidět například na obr. 7.2 při komunikaci klienta na webový server. Na níže uvedeném obrázku je také vidět dekodování UDP streamu na RTP, který je využíván pro přenos RTSP zpráv.

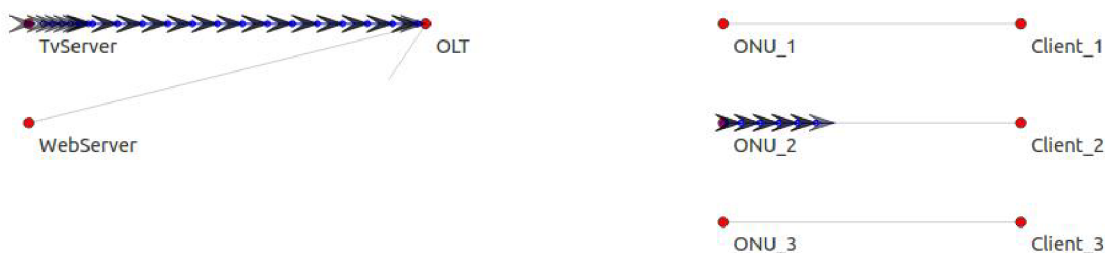
No.	Time	Source	Destination	Protocol	Length	Info
827	6.218325	172.0.0.1	160.0.1.1	TCP	56	[TCP Port number reused] 49153→6000 [SYN] Seq=4294966784 Win=32768 Len=0
828	6.226000	160.0.1.1	172.0.0.1	TCP	56	6000→49153 [SYN, ACK] Seq=4294967295 Ack=4294966785 Win=32768 Len=0 Ws=4
829	6.230075	172.0.0.1	160.0.1.1	TCP	52	49153→6000 [ACK] Seq=4294966785 Ack=0 Win=131072 Len=0 TSval=6231 TSecr=6
830	6.424500	160.0.1.1	172.0.0.1	RTP	328	Unknown RTP version 0
831	6.429700	172.0.0.1	160.0.1.1	X11	564	49153→6000 [ACK] Seq=4294966785 Ack=0 Win=131072 Len=512 TSval=6431 TSecr=
832	6.430075	172.0.0.1	160.0.1.1	X11	564	49153→6000 [ACK] Seq=1 Ack=0 Win=131072 Len=512 TSval=6431 TSecr=6224
833	6.436000	160.0.1.1	172.0.0.1	TCP	52	6000→49153 [ACK] Seq=0 Ack=1 Win=130560 Len=0 TSval=6435 TSecr=6431
834	6.440200	172.0.0.1	160.0.1.1	IPv4	1500	Fragmented IP protocol (proto=TCP 6, off=0, ID=0008) [Reassembled in #835]
835	6.440200	172.0.0.1	160.0.1.1	X11	72	49153→6000 [ACK] Seq=513 Ack=0 Win=131072 Len=1500 TSval=6441 TSecr=6435
836	6.447500	160.0.1.1	172.0.0.1	TCP	52	6000→49153 [ACK] Seq=0 Ack=2013 Win=129572 Len=0 TSval=6446 TSecr=6441
837	6.451700	172.0.0.1	160.0.1.1	IPv4	1500	Fragmented IP protocol (proto=TCP 6, off=0, ID=0009) [Reassembled in #838]
838	6.451700	172.0.0.1	160.0.1.1	X11	72	49153→6000 [ACK] Seq=2013 Ack=0 Win=131072 Len=1500 TSval=6453 TSecr=6446
839	6.451825	172.0.0.1	160.0.1.1	IPv4	1500	Fragmented IP protocol (proto=TCP 6, off=0, ID=000a) [Reassembled in #840]
840	6.451825	172.0.0.1	160.0.1.1	X11	72	49153→6000 [ACK] Seq=313 Ack=0 Win=131072 Len=1500 TSval=6453 TSecr=6446
841	6.451950	172.0.0.1	160.0.1.1	IPv4	1500	Fragmented IP protocol (proto=TCP 6, off=0, ID=000b) [Reassembled in #842]
842	6.451950	172.0.0.1	160.0.1.1	X11	72	49153→6000 [ACK] Seq=5013 Ack=0 Win=131072 Len=1500 TSval=6453 TSecr=6446

Obr. 7.19: VoD – odesílání dat správným směrem.

Při zobrazení throughputu na obr. 7.10 bylo využito zobrazení pomocí logaritmické vertikální osy, pokud je ale použito lineární místo logaritmické osy jako na obr. 7.20, je vidět, že se v době, kdy má být vysílán stream s konstantní bitovou rychlostí, objevuje výrazná špička. Důvodem je vznik situace, kterou ilustruje obr. 7.21 získaný pomocí NetAnimu, kdy dochází z neznámého důvodu k pomalému zvyšování počtu vysílaných paketů s daty ze strany serveru a paketů se zprávou ACK ze strany klienta.



Obr. 7.20: Throughput VoD bez použití logaritmické osy.



Obr. 7.21: Zahlcení při VoD.

V čase simulace 6,54s dochází k samovolné nápravě této situace.

BitTorrent

Pro zobrazení výsledků v této části simulace byly generovány další výstupy pro program Wireshark. U těchto souborů neodpovídá zaznamenaný čas času simulace z důvodu zaznamenávání od doby, kdy je zaznamenán první paket.

Konec kapitoly pojednávající o simulaci BitTorrent protokolu obsahuje obr. 7.14 s grafem vytížení, který je zároveň grafem rychlosti sdílení. Tento graf neodpovídá skutečnosti, jelikož jeho zaznamenaná rychlost je brána z jednotlivých paketů procházejících rozhraním směrovače R. Reálná rychlost sdílení dat udávaná bittorrent

klientem je uvedena na obr. 7.22, kde nedochází k záznamu špiček ale průběh je vykreslen s ohledem na bitovou rychlost za vteřinu. Rychlost sdílení je v tomto případě omezena definováním parametru *Maximální upload (kB/s)* na hodnotu 1. V grafu je z důvodu vysokého poměru download/upload křivka uploadu špatně viditelná.



Obr. 7.22: Obecný graf přenosové rychlosti udávaný bittorrent klientem.

Bittorrent klient také zobrazuje další informace, jako aktuální rychlost uploadu, počet připojených peerů, seznam trackerů nebo seznam peerů, se kterými je aktuálně spojen, což je uvedeno na obr. 7.23. Z obrázku je vidět, že všichni klienti jsou seedery, jelikož vlastní 100 % dat a klient od těchto klientů pouze stahuje, žádná data neodesílá. Pokud by byla vytvořena podobná tabulka pro simulovaný scénář, byly by v této tabulce uvedeny uzly Source_A, Source_B a Source_C (pokud by byly v daný moment všechny aktivní).

IP	Klient	Směr	Procent	Rychlost Down	Rychlost Up	Požadavky	Odesláno	Staženo
20:	µTorrent 3.4.2	D IHXP	100.0	263.7 kB/s	0.2 kB/s	67 0		33.6 MB
92:	µTorrent 2.2.1	D HP	100.0	122.9 kB/s	0.1 kB/s	28 0		23.5 MB
36:	µTorrent 3.4.2	D IHP	100.0	171.4 kB/s	0.1 kB/s	50 0		10.9 MB
37:	µTorrent 3.4.2	D HX	100.0	89.3 kB/s	0.1 kB/s	27 0		6.40 MB
86:	µTorrent 3.4.2	D IHXP	100.0	139.3 kB/s	0.2 kB/s	50 0		5.92 MB
21:	BitTorrent 7.9.2	D HXP	100.0	148.7 kB/s	0.1 kB/s	35 0		5.42 MB
77:	BitTorrent 7.9.2	D HXP	100.0	172.6 kB/s	0.1 kB/s	65 0		3.90 MB
46:	µTorrent 2.2.1	D IXEP	100.0	191.5 kB/s	0.1 kB/s	61 0		3.57 MB
31:	uTorrent 3.3	D HXP	100.0	116.3 kB/s	0.0 kB/s	34 0		3.07 MB
18:	µTorrent Mac 1.8.7	D IXP	100.0	128.8 kB/s	0.1 kB/s	30 0		2.34 MB
94:	µTorrent 3.3.2	D XP	100.0	20.0 kB/s		7 0		1.60 MB
21:	µTorrent 3.4.2	D HXP	100.0	8.7 kB/s		4 0		592 kB
79:	µTorrent 3.4.2	D HXP	100.0	8.8 kB/s		4 0		432 kB
92:	BitTorrent 7.9.2	D XP	100.0	2.0 kB/s		3 0		320 kB

Obr. 7.23: Seznam připojených klientů udávaný bittorrent klientem.

IPTV

V této části simulace jde o zobrazení principu multicastového vysílání IPTV skrze PON, v případě prostředí NS-3 skrze modul XG-PON. Tohoto bohužel nebylo plně dosaženo. Modul XG-PON obsahuje dvě základní jednotky, a to OLT a ONU. Pro jednotku OLT je v modulu XG-PON definováno rozhraní `XgponOltNetDevice` a pro jednotku ONU `XgponOnuNetDevice`, obě typu `NetDevice`.

Obecný příkaz pro přidání zdroje multicastového vysílání je uveden níže. V tomto příkazu `node` označuje uzel, který je zdrojem vysílání a rozhraní definuje odchozí rozhraní tohoto vysílání.

```
multicast.SetDefaultMulticastRoute(node, rozhraní);
```

Níže uvedený obecný příkaz provádí přidání cesty pro multicastové vysílání, kde je definován uzel, přes který vysílání prochází, adresa zdroje vysílání, multicastová adresa, jedno vstupní rozhraní uzlu a výstupní rozhraní (i několik) na tomto uzlu. Vstupní rozhraní uzlu je požadováno typu `NetDevice`, ovšem výstupní jako `NetDeviceContainer`.

```
multicast.AddMulticastRoute(node, IP zdroje, multicastová skupina,  
vstupní rozhraní uzlu, výstupní rozhraní uzlu);
```

Jako vstupní rozhraní lze zadat `NetDeviceContainer` (s parametrem `.Get(i)` pro výběr rozhraní z tohoto containeru), ovšem po zadání výstupního rozhraní typu `NetDevice` nehlásí prostředí žádné chyby, avšak tento příkaz nefunguje. Z tohoto důvodu není možné uskutečnit vysílání skrze XG-PON modul. Proto byla zvolena varianta, kdy byly vytvořeny v každém z uzlů ONU nové zdroje vysílání se stejnou multicastovou adresou a vysílání je směřováno do klientské části sítě.

Pro možnost záznamu vyslaných paketů byl níže uvedeným kódem generován výstup `onu_3-pcap-3-1.pcap`.

```
NetDeviceContainer onu_3track;  
onu_3track.Add(p2pCDevices[2].Get(1));  
  
pointToPoint.EnablePcap("onu_3-pcap", onu_3track);
```

Při pohledu na zobrazené výsledky se nachází odchylky na obr. 7.17, kdy je zdrojem soubor `onu_3-pcap-3-1.pcap`. Jedná se o soubor generovaný z rozhraní uzlu `ONU_3` v klientské části sítě, který začíná zaznamenávat s prvním procházejícím paketem, neshoduje se zde tedy časování simulace a tohoto souboru. Druhou odlišností na tomto obrázku je použití UDP protokolu pro pakety číslo 336, 337 a 351, kdy by tyto pakety měly být pro reálné zobrazení překódovány na protokol PIM,

který se ke komunikaci mezi routery využívá. Program Wireshark není ovšem schopen dekódovat takto pouze vybrané pakety (ani při aplikaci filtru na cílový port), ale dekóduje celou UDP komunikaci, což je nežádoucí.

8 ZÁVĚR

Bakalářská práce se věnuje tématu Triple Play a simulaci dané služby v sítích XG-PON v prostředí NS-3. V rámci teoretické části práce byly představeny pasivní optické sítě jak z hlediska funkčnosti, tak z hlediska historického. V práci bylo také nastíněno, jakými způsoby mohou být do sítě připojeni koncoví uživatelé.

Následně byl představen soubor služeb, které obsahuje Triple Play. Pro každou z těchto služeb byl vysvětlen princip, její výhody a nevýhody plynoucí ze začlenění do prostředí IP sítí. Jedním z hlavních důvodů přechodu uživatelů k službě Triple Play je nabízení IPTV. U IPTV byly popsány pro uživatele velmi lákavé služby Video on Demand a Pay per View. V závěru byly popsány očekávání u služby Triple Play do budoucnosti.

Vs stručnosti byly popsány metody IntServ a DiffServ pro zajištění doručení nabízených služeb v požadované kvalitě a také BitTorrent protokol.

V rámci praktické části byly v prostředí NS-3 simulovány jednotlivé služby Triple Play. V rámci podkapitoly 7.3 je každá část simulace věnující se jedné službě rozepsána detailněji.

U služeb PPV a komunikace na webové stránky nebyly objeveny žádné problémy, pouze byla simulace v podkapitole 7.3.6 teoreticky doplněna o další používané protokoly a grafické výstupy byly upraveny uvedeným způsobem pro zobrazení reálně využívaných protokolů.

V části simulující službu VoD bylo nutno pro správné zobrazení v grafické výstupu použít navázání spojení z uzlu serveru, které reálně navazuje klient. Tato situace je dále popsána v podkapitole 7.3.6. Ve zdrojovém kódu nebylo možno vytvořit na stejném uzlu dva sinky pro jeden transportní protokol, používající pouze jiné porty, proto byl pro tuto část použit stejný sink jako pro simulaci PPV.

Pro simulaci BitTorrent protokolu nebylo možno využít některého z modulů, jelikož nebyl tento protokol doposud zahrnut do NS-3 simulátoru. Simulace byla vytvořena pomocí již existujících aplikací. Pro zobrazení celkového throughputu byl v topologii vytvořen jeden uzel – směrovač R, přes který je směrována veškerá komunikace. V prostředí NS-3 ovšem není možno sledovat zmíněnou statistiku na uzlu, pouze jednotlivě na jeho rozhraních, proto je zobrazeno vytížení linky směrem ze směrovače R ke klientovi Client_3, kde se nachází optická síť. V podkapitole 7.3.6 je pro srovnání uvedeno, jak je zobrazována přenosová rychlost klienty pro sdílení dat pomocí BitTorrent protokolu.

Poslední simulovanou službou v této práci je IPTV, kdy vysílání služby mělo být realizováno pomocí multicastového vysílání skrze optickou síť. Vysílání se nepodařilo realizovat skrze modul XG-PON, proto bylo realizováno ve dvou částech, a to prvně na serverové části sítě a poté na části klientské. V grafickém výstupu

pro program NetAnim není ovšem zvolená realizace vidět z důvodu nezobrazování průchodu paketů v optické síti.

Pro celkový pohled na tuto práci je nutno při čtení práce pracovat i s výstupy, které jsou přiloženy jako příloha k tomuto dokumentu.

LITERATURA

- [1] Cisco Systems. *Cisco.com* [online]. 2010, [cit. 14. 11. 2014]. Differentiated Services. Dostupné z URL: <<http://www.cisco.com/c/en/us/products/ios-nx-os-software/differentiated-services/index.html>>.
- [2] FILKA, M. *Přenosová média*. 1. vyd. Brno: FEKT VUT v Brně, 2012, 202s.
- [3] GORSHE, S.;MANDIN, J. *Introduction to IEEE 802.3av 10Gbit/s Ethernet Passive Optical Networks (10G EPON)*. [online]. 2009, [cit. 2. 11. 2014]. Dostupné z URL: <<http://www.chinacommunications.cn/fileup/PDF/2009-6-4-021.pdf>>.
- [4] HENS, Francisco J a José Manuel CABALLERO. Triple play: building the converged network for IP, VoIP and IPTV. Hoboken, NJ: Wiley, c2008, xiii, 401 p. ISBN 04-707-5367-6.
- [5] HON, P. Jak funguje řízení datových toků s QoS. *Connect.cz* [online]. 2012, [cit. 10. 11. 2014]. Dostupné z URL: <<http://connect.zive.cz/clanky/jak-funguje-rizeni-datovych-toku-s-qos/sc-320-a-161738>>.
- [6] HORVÁTH, T.; ŠIFTA, R.; MÜNSTER, P. Detekce hudebních a komprimovaných souborů v BitTorrent protokolu. *Access Server* [online]. 2013, [cit. 28. 2. 2015]. Dostupné z URL: <<http://access.feld.cvut.cz/view.php?cisloclanku=2013040001>>.
- [7] IEEE. *802.3ah - IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements* [online]. 2004, [cit. 7. 11. 2014]. Dostupné z URL: <http://www.ieee802.org/21/doctree/2006_Meeting_Docs/2006-11_meeting_docs/802.3ah-2004.pdf>.
- [8] ITU-T. *G.983.1 : Broadband optical access systems based on Passive Optical Networks (PON)* [online]. 2005, [cit. 2. 11. 2014]. Dostupné z URL: <<http://www.itu.int/rec/T-REC-G.983.1-200501-I/en>>.
- [9] ITU-T. *G.984.1 : Gigabit-capable passive optical networks (GPON): General characteristics* [online]. 2008, [cit. 1. 11. 2014]. Dostupné z URL: <<http://www.itu.int/rec/T-REC-G.984.1/en>>.
- [10] ITU-T. *G.987.1 : 10-Gigabit-capable passive optical networks (XG-PON): General requirements* [online]. 2010, [cit. 1. 11. 2014]. Dostupné z URL: <<http://www.itu.int/rec/T-REC-G.987.1-201001-I>>.

- [11] KEISER, Gerd. FTTX concepts and applications. Hoboken, N.J.: IEEE, c2006, xvii, 293 p. ISBN 978-047-1704-201.
- [12] KOLÁŘ, R. BitTorrent - Technologie. *ROOT* [online]. 2004, [cit. 28. 2. 2015]. Dostupné z URL: <<http://www.root.cz/clanky/bittorrent-technologie/>>.
- [13] KREJČÍ, J.; ZEMAN, T. Úvod do IPTV. *Access Server* [online]. 2008, [cit. 14. 11. 2014]. Dostupné z URL: <<http://access.feld.cvut.cz/view.php?cisloclanku=2008100002>>.
- [14] LAFATA, P.; VODRÁŽKA, J. Optické přístupové sítě a přípojky FTTx. Vyd. 1. Praha: České vysoké učení technické v Praze, 2014, 293 s. ISBN 978-80-01-05463-5.
- [15] O'DRISCOLL, Gerard. Next generation IPTV services and technologies. Hoboken, N.J.: Wiley-Interscience, c2008, xvii, 490 p. ISBN 978-0-470-16372-6.
- [16] PASTYŘÍK, M. Digitalizace zvuku. *osu.cz* [online]. 2012, [cit. 16. 11. 2014]. Dostupné z URL: <<http://hucak.osu.cz/zvukapocitac/4.html>>.
- [17] *Rlc.cz* [online]. 2010, [cit. 14. 11. 2014]. FTTx ŘEŠENÍ. Dostupné z URL: <<http://fttx.rlc.cz/?p=1>>.
- [18] SIMPSON, Wes. Video over IP: IPTV, internet video, H.264, P2P, and streaming: a complete guide to understanding the technology. 2nd ed. Amsterdam: Focal Press, 2008, xxi, 478 s. ISBN 978-0-240-81084-3.
- [19] The NS-3 Development team. Documentation. *Ns-3* [online]. [cit. 2014-12-05]. Dostupné z URL: <<http://www.nsnam.org/ns-3-21/documentation/>>.
- [20] TODOROV, P. Allied Telesis - Přes FTTx a EPON po kruhovou topologii. *NETGURU* [online]. 2010, [cit. 14. 11. 2014]. Dostupné z URL: <<http://www.netguru.cz/09102-penos-vidoa-v-ip-sitich/allied-telesis-pes-fttx-a-epon-po-kruhovou-topologii.html>>.
- [21] ZAKOWSKI, B. Passive vs. Active Optical Networking. [online]. 2004, s. 28-29 [cit. 14. 11. 2014]. Dostupné z URL: <<http://www.broadbandproperties.com/2004issues/may04issues/fibercity.pdf>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AF	Assured Forwarding
AON	Active Optical Network
APON	ATM PON
ATM	Asynchronous Transfer Mode
BER	Bit Error Rate
BPON	Broadband PON
CoS	Class of Service
CS	Class Selector
DHT	Distributed Hash Table
DiffServ	Differentiated Services
DSCP	Differentiated Services Code Point
DVD	Digital Video Disc
EF	Expedited Forwarding
EPON	Ethernet Passive Optical Network
FBT	Fused Bionic Taper
FDM	Frequency-Division Multiplexing
FTTA	Fiber to the Antenna
FTTB	Fiber to the Building
FTTC	Fiber to the Curb
FTTH	Fiber to the Home
FTTN	Fiber to the Node
GEM	GPON Encapsulation Mode
GPON	Gigabit-capable Passive Optical Networks
HDTV	High-definition Television

HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
IBM	International Business Machines Corporation
IEEE	Institute of Electrical and Electronics Engineers
IGMP	Internet Group Management Protocol
IntServ	Integrated Services
IP	Internet Protocol
IPTV	Internet Protocol Television
ISP	Internet Service Provider
ITU	International Telecommunication Union
iTV	Interactive Television
LLID	Logical Link Identifier
MPEG	Moving Picture Experts Group
OLT	Optical Line Terminal
ONT	Optical Network Terminal
ONU	Optical Network Unit
P2MP	Point to Multipoint
P2P	Point to Point
PC	Personal Computer
PHB	Per Hop Behaviour
PIM	Protocol- Independent Multicast
PLC	Planar Lightwave Circuit
PLC	Packet Loss Concealment
PON	Passive Optical Network

PPV	Pay per View
QoS	Quality of Service
RSVP	Resource Reservation Protocol
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
SDP	Session Description Protocol
SDTV	Standard Definition Television
SLA	Service Level Agreement
SNR	Signal to Noise Ratio
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
ToS	Type of Service
TTL	Time to Live
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VA	Voice Admit
VoD	Video on Demand
VoIP	Voice over Internet Protocol
WDM	Wavelength Division Multiplexing
WLAN	Wireless Local Area Networks
XGTC	XG-PON Transmission Convergence

SEZNAM PŘÍLOH

A Obsah přiloženého CD

62

A OBSAH PŘILOŽENÉHO CD

Na CD je přiložena elektronická verze bakalářské práce ve formátu PDF. Přiloženy jsou také soubory

- pro Wireshark:
Bachelor-pcap-0-0.pcap,
Bachelor-pcap-1-0.pcap,
Bachelor-pcap-2-0.pcap,
Bachelor-pcap-3-0.pcap,
BT-pcap-9-0.pcap,
IPTV-pcap-4-0.pcap,
ONU_3-pcap-3-1.pcap,
SourceA-pcap-9-1.pcap,
SourceB-pcap-9-2.pcap.
- pro NetAnim:
Bachelor.xml.