

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

ANALÝZA VÝMĚNNÝCH SÍTÍ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

LIBOR BLAŽEK

BRNO 2014



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

ANALÝZA VÝMĚNNÝCH SÍTÍ

ANALYSIS OF SHARING NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

LIBOR BLAŽEK

VEDOUcí PRÁCE
SUPERVISOR

Ing. TOMÁŠ HORVÁTH

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Libor Blažek

ID: 136453

Ročník: 3

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Analýza výměnných sítí

POKYNY PRO VYPRACOVÁNÍ:

V rámci bakalářské práce se bude nutné seznámit s výměnnými protokoly na principu peer to peer. Zejména se detailně seznámit s výměnným protokolem BitTorrent a podrobně jej popsat. Další kapitoly teoretické části bude tvořit popis obrany proti šíření výměnných protokolů v počítačové síti a její bezpečnost.

V praktické části pak bude návrh a implementace BitTorrent protokolu v simulačním prostředí OPNET Modeler. Student provede simulaci pro lokální a rozlehlou počítačovou síť a porovná jejich výsledky. Výsledek bakalářské práce bude zhodnocení vlastní implementace BitTorrent protokolu.

DOPORUČENÁ LITERATURA:

KOEGEL BUFORD, John F, Hong Heather YU a Eng Keong LUA. P2P networking and applications. Burlington: Morgan Kaufmann, c2009, xxi, 415 s. The Morgan Kaufmann series in networking. ISBN 978-0-12-374214-8.

HORVÁTH, T.; ŠIFTA, R.; MÜNSTER, P. Detekce hudebních a komprimovaných souborů v BitTorrent protokolu. Access Server, 2013, roč. 11, č. 1, s. 1-4. ISSN: 1214- 9675.

OPNET Technologies, OPNET Modeler Product Documentation Release 17.5, 2013.

Termín zadání: 10.2.2014

Termín odevzdání: 4.6.2014

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 bakalářské práce je návrh a implementace protokolu v prostředí Opnet Modeler. V uvedené práci budou vytvořeny veškeré prvky, nebyly použité žádné výchozí nastavení. V prvních kapitolách tvorby simulace jsou definovány pakety. Další kapitolu tvoří blokové schéma koncové stanice, ve které se přiřazují funkce blokům. Pro centrální proces je vytvořen procesní model, také pro router. Jednotlivé prvky jsou vloženy do palety s prvky a následně použité v simulaci. Spojení proběhlo novou linkou, která byla definována zvlášť. Jedna z vytvořených simulací bude v menším rozsahu a druhá v globálním měřítku s více prvky.

KLÍČOVÁ SLOVA

BitTorrent, Opnet, FTP, propustnost, využití sítě, Node Model, Process Model, paket

ABSTRACT

The aim of the thesis is the design and implementation of the protocol in OPNET Modeler. In this work will be developed all the elements have been used no default settings. In the first chapters, making simulation are defined packets. Another chapter consists of a block diagram of workstations, which are assigned to function blocks. The central process is created by the process model, also for the router. The individual elements are inserted into a variety of elements and subsequently used in the simulation. Connection is a new service that has been defined separately. One of the simulations will be created on a smaller scale and the other on a global scale with more features.)

KEYWORDS

BitTorrent, Opnet, File Transfer Protocol, throughput, utilization, Node Model, Process Model, paket

BLAŽEK, Libor *Analýza výměnných sítí*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 46 s. Vedoucí práce byl Ing. Tomáš Horváth

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Analýza výměnných sítí“ 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 diplomové práce panu Ing. Tomáši Horváthovi, za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)

OBSAH

1	Úvod do P2P sítí	9
1.1	Historie	9
1.2	Napster	10
1.2.1	Princip	10
1.3	Gnutella	11
1.3.1	Princip	11
1.3.2	PING - PONG	12
2	Mesh síť	13
2.1	Architektura	13
2.2	Princip	13
2.3	Technologie ORiNOCO	14
3	BitTorrent	15
3.1	Charakteristika	15
3.2	Životnost a bezpečnost	16
3.3	Princip	16
3.3.1	Pojmy	16
3.4	Magnet Link	18
4	OPNET Modeler	20
5	První simulace - VoIP, Video	21
5.1	Vytvoření sítě	21
5.2	Application configure	21
6	Protokol BitTorrent	26
6.1	Vytvoření paketu	26
6.2	Uživatel - Seed	27
6.2.1	Node Model	27
6.2.2	Process Model	29
6.3	Uživatel - Peer	30
6.4	Uživatel - Leech	30
6.5	Router	31
6.5.1	Process Model	31
6.6	Tvorba komunikačního vedení	32

7	Programování jednotlivých částí	33
7.1	Seed, Peer, Leech, Router	33
8	Vložení prvků do scénáře	36
9	Výsledky bakalářské práce	38
9.1	Simulace BitTorrent protokolu	38
9.2	Výsledky simulace	38
10	Závěr	41
	Literatura	42
	Seznam symbolů, veličin a zkratk	44
	Seznam příloh	45
A	Obsah CD	46

SEZNAM OBRÁZKŮ

1.1	Schéma - Napster.	10
1.2	Schéma - Gnutella.	11
1.3	PING - PONG.	12
3.1	Schéma - BitTorrent.	15
3.2	Schéma - BitTorrent - pojmy.	18
4.1	Úvodní okno pro Opnet Modeler.	20
5.1	Vytvořená pracovní síť.	22
5.2	Obsah v subnetu - OSTRAVA_VIDEO.	22
5.3	Základní menu v Aplikační konfiguraci.	23
5.4	Nastavení audio přenosu.	23
5.5	Nastavení atributů v „prof_conf“.	24
5.6	Nastavení atributů v „PC_VOIP“, nastavují se zde vytvořené profily.	25
5.7	Znázornění přenosu Video a Audio v síti.	25
6.1	Schéma zapojení pro využití protokolu BitTorrent.	26
6.2	Výběr na tvorbu paketu.	27
6.3	Vytvořený paket pro odpověď BitTorrentu.	27
6.4	Blokové schéma koncové stanice - Seed.	28
6.5	Zapojení procesu - platí pro všechny koncové stanice	29
6.6	Zapojení Node Model pro uživatele Peer (vlevo) a Leech (vpravo).	30
6.7	Zapojení Node Model aktivní prvek - Router.	31
6.8	Zapojení Process Model pro Router.	32
6.9	Nastavení vlastností pro linku spojující prvky.	32
7.1	Definování proměnných v koncové stanici a routeru.	33
7.2	Funkce jednotlivých smyček vytvořených v procesním modelu.	34
7.3	Část kódu pro „FB“, pouze pro „xmt“ vysílač.	35
7.4	Část kódu pro „FB“, jedná se o router.	35
7.5	Zapsání parametrů pro pakety a graf.	35
8.1	Vyvolání nabídky na přidání vytvořených prvků.	36
8.2	Vložení prvků - změna z <i>not included</i> na <i>included</i>	36
9.1	Zobrazen průběh simulace - základní scénář	38
9.2	Zobrazen průběh simulace - základní scénář (nahore), výpadek sítě (dole)	39
9.3	Výsledky Simulace - zobrazení a porovnání grafů s nadefinovaným výpadkem sítě mezi AP a koncovou stanicí	39
9.4	Zobrazen průběh simulace - základní scénář, zobrazení utilization, throughput a delay, mezi prvky a AP	40

1 ÚVOD DO P2P SÍTÍ

Sítě Peer To Peer, neboli taky P2P, jsou takové sítě, které umožňují komunikaci klient-klient (mezi koncovými uzly). Další způsob komunikace je klient-server, to však již nespadá do P2P sítí, jelikož při tomto zapojení jsou si uzly rovnocenné.

Z toho vyplývá, že úlohu serveru a klienta zastupuje každý prvek v síti. Existují také specializované servery sloužící pouze k navázání spojení (proxy server).

P2P sítě jsou dnes hojně využívány především ke sdílení a stahování dat od jiných uživatelů. Hlavní podmínkou pro stahování je nutnost sdílení obsahu, to je již v tomto případě nelegální a sdílení nelegálního obsahu není cílem této práce, nebudou porušována žádná autorská práva a zákony, jenž jsou platná v ČR.

Hlavním cílem výměnných sítí bylo rychlé šíření velkých souborů mezi uživatele, jako například linuxové distribuce, nebo různé programy. Na principu P2P sítí fungují i některé programy, jako je Skype, IRC (Internet Relay Chat, jedná se o internetovou komunikaci v reálném čase), Qnext.

Jednou z hlavních výhod sítě založené na rovnocennosti je to, že s větším množstvím uživatelů přenosová rychlost roste, při zapojení klient-server, kde má konstantní kapacitu pro všechny připojené, to znamená, že při větším počtu stahujících uživatelů rychlost klesá.

1.1 Historie

Prvním nejrozšířenější sítí využívající sdílení využila služba Napster v roce 1999. Jejím hlavním cílem bylo sdílení zvukových souborů, převážně mp3. Tato služba byla závislá na hlavním serveru, který navazoval spojení mezi klienty. Jakmile byla potřeba zrušit tuto službu, stačilo odpojit server. Dalším z hlavních představitelů byla služba Gnutella.

Vytvořila ji skupina programátorů z firmy Nullsoft, která vyvinula a v dnešní době je již prodán belgické firmě Radionomy. Gnutella byla spuštěna 14.března v roce 2000 v období, kdy měl problémy Napster a díky tomu se služba musela následující den zrušit. Od celé věci se firma Nullsoft distancovala a oznámila, že se jednalo o soukromý projekt dvou zaměstnanců firmy.

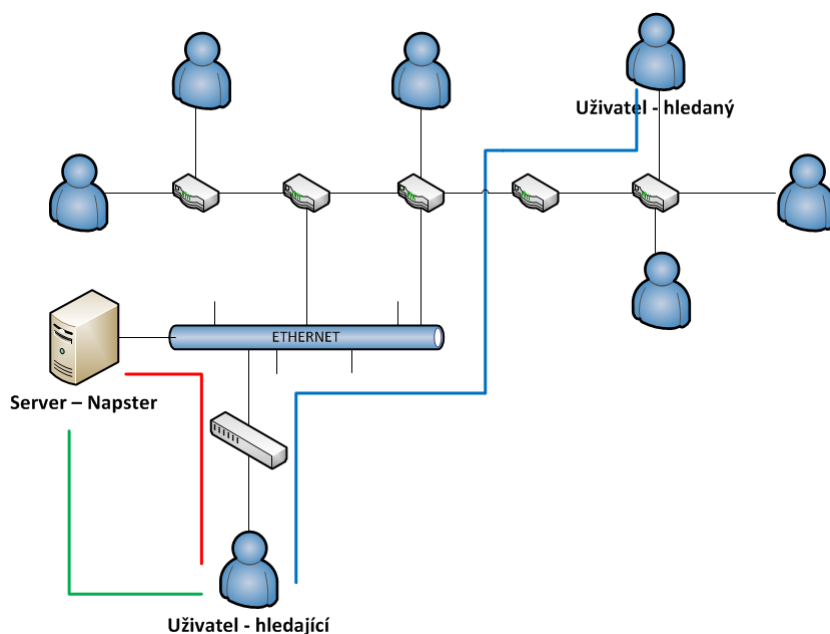
Než byla služba zrušena, byly již vytvořeny reverzní kopie (zpětný překlad kódu) do programovacího jazyka assembler. Poté umístili kódy na Internet a tím vzniklo spousta klientů na bázi Gnutelly.

1.2 Napster

Jedná se o P2P službu zaměřenou na sdílení hudebního obsahu. Službu vytvořil Shaw Fanning. Jelikož služba nabízela nelegální obsah, přesněji tedy uživatelé sdíleli chráněný obsah, převážně mp3, byla služba obviněna hudebním průmyslem z porušování autorských práv.

Tato služba byla velmi oblíbená a vyživovala ji desítky milionů uživatelů, později musel být na soudní příkaz vypnuta. Hlavní příčině na vypnutí serveru měla hudební skupina Metallica. Členové si všimli, že jejich demo písniček kolovala na síti ještě před oficiálním vydáním [2].

1.2.1 Princip



Obr. 1.1: Schéma - Napster.

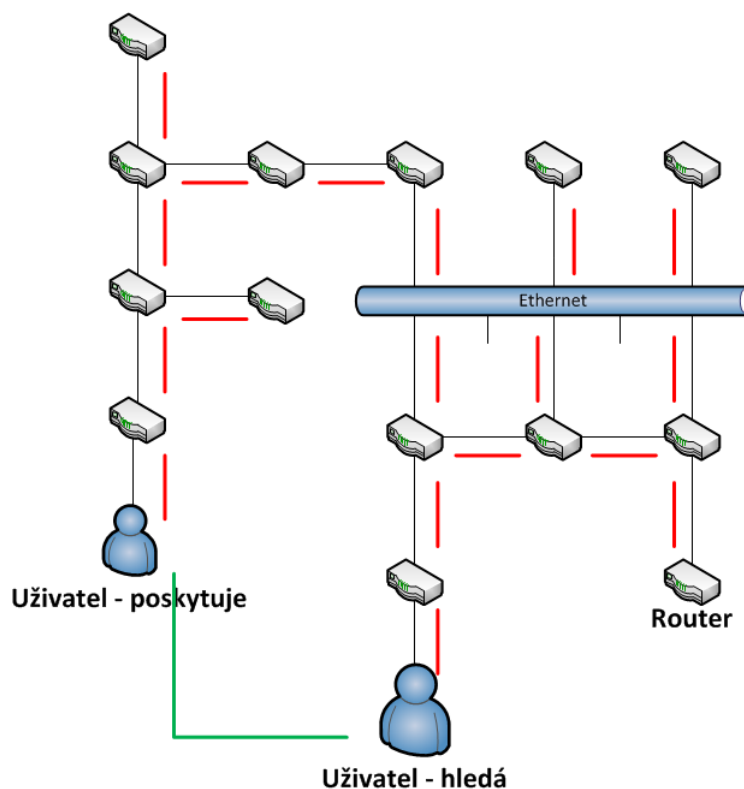
Na obr. 1.1 je znázorněn princip fungování, kde je zobrazen uživatel-hledající a hledaný. Jelikož nemá žádné informace o sdílených datech, posílá dotaz na server (červená barva). V této službě funguje pouze jeden jedinný server, který má veškeré údaje a posílá zpátky informace, kde se hledaný soubor nachází (zelená barva). Jakmile jej uživatel dostane, vytvoří si spojení s tím uživatelem, který poskytuje daný soubor a stáhne si jej (modrá barva).

1.3 Gnutella

Tato síť funguje na odlišném vyhledávání než Napster. Zde se nevyužívá žádný server, jelikož všichni uživatelé jsou si rovni a tvoří jak server, tak i klienta. Tito klienti se v síti nazývají Gnode [2]. Každý má informace pouze o počtu svých sousedních uzlů, se kterými je v kontaktu.

1.3.1 Princip

Při posílání dotazu jsou posílány požadavky do všech okolních uzlů (počítačů), obr. 1.2, které sousedí s uživatelem, který posílá žádost o nějaký soubor. Pokud žádný ze sousedních uživatelů nemá požadovaná data, posílá se dotaz opět k dalším sousedním uzlům. Postup je prováděn tak dlouho, dokud není poslána kladná odpověď.

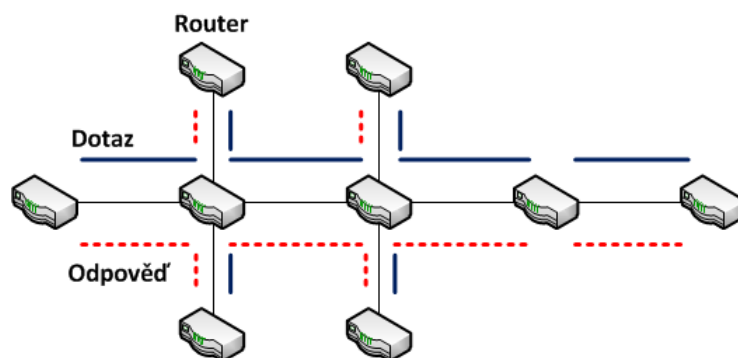


Obr. 1.2: Schéma - Gnutella.

Na obrázku je znázorněné zapojení, kde uživatel posílá dotaz na hledaný soubor (červená barva). Uzel dotaz zpracuje a posílá jej na své sousední uzly. Jakmile dotaz najde hledaný soubor, naváže se spojení mezi hledajícím a hledaným uzlem (zelená barva) a soubor se přenesení. Této metodě se říká PING - PONG.

1.3.2 PING - PONG

Metoda PING - PONG funguje na principu posílání dotazu (PING) a přijímání odpovědi (PONG). Na obrázku 1.3 je vidět schéma fungování metody, kdy je poslán dotaz (modrá barva - plná čára) a odpověď (červená barva - čárkovaná čára). První router posílá PING a následně dostane PONG, je-li odpověď záporná, druhý router posílá PING na další a čeká na PONG dokud není odpověď kladná.



Obr. 1.3: PING - PONG.

Ping se používá k zjištění informací o jiném uzlu v síti (obsahuje TTL a počet uzlů, kterými prošel). Pong obsahuje informace o použitém portu, IP uzlu, počtu sdílených souborů a celkovou velikost dat. Dotaz na hledání souboru se označuje jako „Query“ Pozitivní odpověď je označena jako „QueryHit“ a skládá se ze souborů obsahující data o IP adrese koncového uzlu, číslo portu, přenosovou rychlost, detaily o souboru a PEERID (slouží k identifikaci v uzlu) [8].

2 MESH SÍTĚ

Mesh sítě jsou další alternativou k P2P sítím, fungujícím na bezdrátovém přenosu. V mesh sítích jsou si opět uživatelé i AP (Access point) rovny, tak jak to je i v pevných sítích. Jakékoliv zařízení, připojené bezdrátově, je schopno poskytovat stejnou sadu služeb.

2.1 Architektura

Hlavním cílem je řízení přístupových bodů a oddělit frekvenční pásma. Standard 802.11b/g funguje jako hotspot, pro přístup uživatelů, pásmo 802.11a vytváří datové cesty. Vznikají tedy neomezené bezdrátové sítě, které dokážou pokrýt velké území [7].

2.2 Princip

Přístupové body se navzájem rozpoznají a vytvoří automaticky datové cesty (maximální kapacita s minimálním spožděním). V této síti musí být obsažen aspoň jeden přístupový bod, který bude zastávat bránu k přístupu do Internetu. Další zařízení která navážou spojení, budou vytvářet síť. Zařízení se propojí mezi sebou, avšak pouze v mezích nutných pro fungování sítě.

V principu to funguje takovým způsobem, že jedno zařízení bude připojeno k prvku, který je přímo napojený na síť Internet. Prvek bude do oblasti vysílat bezdrátový signál, na který se bude moct kdokoli připojit, bude jej využívat jako směrovací prvek. Jakmile se připojí, může opět vysílat signál, aby se napojil někdo další. Pokud se uživatel odpojí, celá síť, nebo její část, se přenastaví do funkčního modelu [3].

Mesh sítě mají své výhody:

- **„zastupitelnost“**
při zničení, nebo neschopnosti poskytovat signál z daného prvku, tak jej může nahradit kterýkoliv jiný prvek v dosahu.
- **„úsporu pásma“**
spojení navázané díky mesh sítím, je pouze tehdy, když je potřeba a na dobu kterou je nutné. Tudíž není potřeba zabírat velké pásmo.

- **„nízké náklady“**
jelikož se o veškeré nastavení a řízení provozu stará routovací protokol, není potřeba velkých nákladů.
- **„dosah sítě“**
díky „předávání“ signálu a dat, lze dosáhnout velké vzdálenosti.

Mesh sítě mají nevýhodu ve velkém odběru energie, což může být u mobilních zařízení problém. Pokud by uživatel nechtěl mít funkci zapnutou, může být v economy módu vypnuta.

Veškerá komunikace mezi stanicemi je šifrována pomocí standardu AES. Nadále jsou zařízení vzájemně autentizovaná, z toho důvodu je nemožné se připojit bez autorizace, aniž by o tom uživatel nevěděl. Na jedné fyzické struktuře je možné provozovat až 16 virtuálních bezdrátových sítí [7].

2.3 Technologie ORiNOCO

Jedna z technologií zabývající se vytváření WiFi P2P sítě je dvourádiová technologie ORiNOCO. V principu to funguje způsobem, že jedno „radio“ je připojené k páteřní síti WLAN a druhé se využívá k pokrytí prostoru. Jelikož se používají dvě „rádia“, je síť frekvenčně oddělená od páteřní sítě.

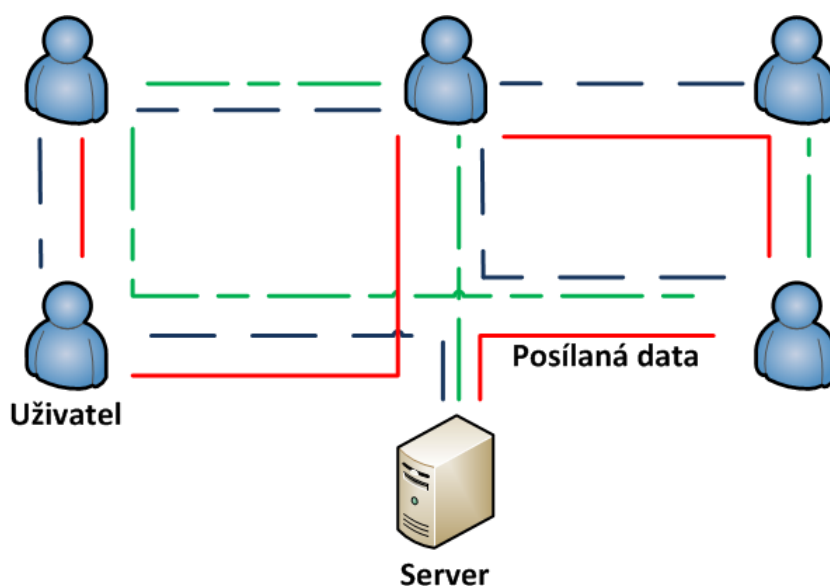
Na přenos dat se používají podle standardu pásma 5 GHz a 2.4 GHz pro přístup do Wifi. ORiNOCO má velmi rychlý datový přenos, podporuje Quality of Service [QoS] - přenos hlasu, videa a ostatních dat. Minimalizuje náklady na vlastní výstavbu. Veškerá komunikace je šifrována, síť zajišťuje také blokování přímé komunikace koncových stanic mezi sebou [5].

3 BITTORRENT

3.1 Charakteristika

Zakladatelem BitTorrentu je Bram Cohen, do provozu byl uveden v roce 2002. Tento protokol je napsán v programovacím jazyce Python a vydán byl jako Open Source. Pokud je využíván BitTorrent, jsou soubory rozděleny na více menších bloků (velikost je nastavitelná) ve výchozím nastavení se jedná o 250 kB pokud se jedná o menší soubory, je-li soubor větší velikosti, rozdělí se až na 4 MB bloky. Každý leech žádá peera o jakýkoliv chybějící blok a zároveň poskytuje již své bloky, které stáhl.

V případě klient-server, se stahují data ze serveru, který musí být výkonný a poskytovat dostatečný upload (posílání dat) klientům. Protokol BitTorrent dovoluje klientovi stahovat jak ze serveru, tak i od ostatních klientů, kteří mají potřebné data. Tím protokol ulehčuje seedům (zdrojům s kompletními daty). Stahování přes BitTorrent je tím rychlejší, čím více je seedů.



Obr. 3.1: Schéma - BitTorrent.

Na obrázku 3.1 je zobrazeno rozesílání dat. Pro lepší přehlednost je použit „Server“, který má celý soubor. Rozdělí jej na 3 části a ty jsou posílány uživatelům. Chybějící data se budou již stahovat od uživatelů, kteří mají danou část.

Každá vytvořená část z celkového souboru obsahuje informace o všech zbylých souborech a také adresu trackeru. Tracker je takzvaný centrální server, který se stará o data a rychlost, kterou budou poskytnuta uživatelům.

3.2 Životnost a bezpečnost

Sít funguje na principu sdílení, tudíž také na tomto stavu záleží životnost sítě. Jakmile by všichni, kdo dostávají soubor, zrušili sdílení souboru, sít by již nadále nefungovala tak, jak by měla. Pouze by nebylo možné stahovat daný soubor, dokud by někdo nezačal sdílet kompletní data. Tomuto stavu se říká že torrent „umře“.

S již zmíněným trackerem souvisí také bezpečnost. Jelikož ovládá a řídí soubory, tak pro omezení P2P sítě stačí aby organizace a velké korporace odstavili právě tracker, který je na daném nelegálním souboru.

Těmto snahám BitTorrent odolával, tyto činnosti směřovaly k vytvoření a používání nového principu přes distribuci Magnet Link. Každý uživatel je v síti reprezentován IP adresou, tudíž se nejedná o bezprostřední anonymní protokol a z toho důvodu je možné sledovat pouze datový tok a vytížení sítě k danému uživateli. Existují způsoby pro maskování IP adres pomocí programů ale nejedná se o zaručenou ochranu uživatele.

BitTorrent protokol vysílá ve výchozím nastavení na portu 6881. Některé programy nedovolují změnu, těchto portů, tudíž je možnost tento port zablokovat a omezit nebo také zastavit přenos dat. Z tohoto důvodu se doporučuje změnit výchozí nastavení, ale nechat aspoň jeden port volný na sdílení souborů.

3.3 Princip

Pokud by uživatel chtěl využívat P2P služeb, musí mít nainstalovanou službu, která podporuje daný protokol. Nejlépe takový program, který je nejvyužívanější, aby bylo sdílení nejefektivnější.

Program uloží údaje do registru, poté je pošle na „domácí server“, to je takový server, který je nutný na funkčnost P2P sítě. Při instalaci programu jsou vybrány soubory, které budou sdíleny. To tvoří „domácí server“, kde jsou uloženy informace o každém uživateli a jeho sdílených souborech. Domácí server pracuje jako prostředník, který navazuje spojení a je nejdůležitější pro tento způsob přenosu.

3.3.1 Pojmy

Při této službě, jsou používány určité pojmy, které jsou vysvětleny. [3]

- **torrent**

Je buď soubor **.torrent**, tedy soubor metadat o downloadu, nebo všechny soubory, které jsou jím popisovány. V tomto souboru jsou uloženy veškeré

informace o souboru, potřebné ke správnému stáhnutí. Jedná se hlavně o velikost, kontrolní součet a také adresa trackeru. Dnes je často nahrazován funkcí magnet link.

- **seed**

Jedná se o označení uživatele, který stále poskytuje celou kopii. Seedování je nejdůležitější na funkci BitTorrentu. Rozhoduje o životnosti a rychlosti downloadu.

- **peer**

Je uživatel, který ještě nemá stažený kompletní soubor a sdílí pouze jeho část.

- **leech**

Jde o uživatele, který si již stáhl kompletní soubor (stává se seedem) a hned poté opouští swarm. Nevhodné je také mít malý poměr upload/download (minimum 1:1), přičemž takové chování je považované za neslušné.

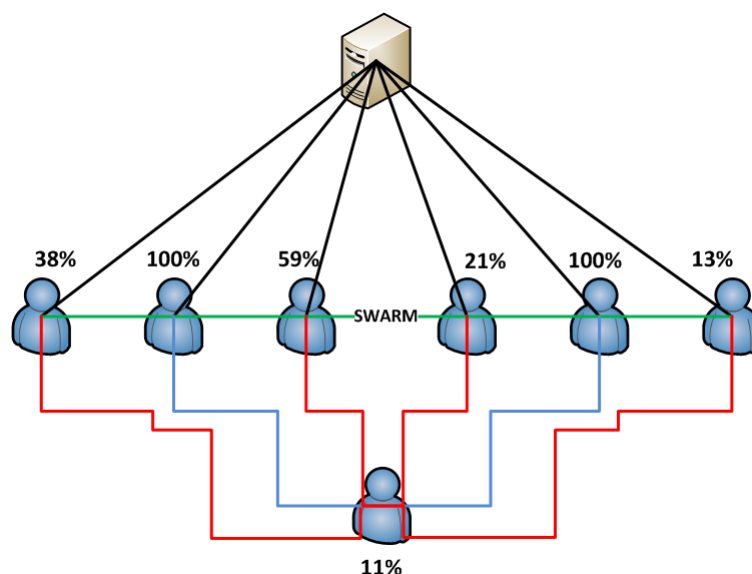
- **swarm**

Je svazek uživatelů, kteří sdílí stejný torrent.

- **tracker**

Je služba, která zprostředkovává a režíruje spojení mezi klienty (přechovává seznamy IP adres peerů), ale data nejsou přes ně přenášena, ani nemá žádnou kopii torrentu. (Při sdílení musí být nastavena adresa rozhraní pro nabízení torrentů, tj nejčastěji „adresa/announce.php“ a následně nahrát soubor .torrent na tracker - tento krok většinou vyžaduje registraci a přihlášení pod jménem)

Obrázek 3.2 znázorňuje schéma, které popisuje jednotlivé části BitTorrentu. Pro lepší přehlednost je umístěn server, který obsahuje veškeré data. Server rozešle data uživatelům, kteří jsou v tzv. swarmu. Jakmile jeden z uživatelů má 100 % dat, stává se z něj seed, který data už jen poskytuje (modrá barva). Ostatní uživatelé, kteří nemají 100 % dat, poskytují a stahují od ostatních uživatelů soubory, které jim chybí(červená barva).



Obr. 3.2: Schéma - BitTorrent - pojmy.

3.4 Magnet Link

Jakmile se do sdílení pomocí Torrentů zapojily organizace na ochranu autorských práv, Evropská unie (EU) a USA, bylo v plánu vytvořit mezinárodní dohodu ACTA (Anti-Counterfeiting Trade Agreement) a SOPA (Stop Online Piracy Act) pouze v USA, měl nejznámější a největší web, The Pirate Bay (dále TPB), s poskytováním těchto služeb velké problémy.

V důsledku omezování svobody k přístupu informacím, nebyla zatím žádná dohoda v globálním měřítku uznána. Jediným možným řešením, jak předejít problémům se sdílením, bylo skončit s Torrenty a přejít na Magnet Linky (dále ML). Soubory, které se ukládaly na web TPB obsahovaly veškeré informace jako IP adresy poskytovatelů, informace o datech. Jakmile se přešlo na ML, veškerá zodpovědnost přešla na poskytovatele (leechery a seedery). Hlavní přednost ML je nepotřebnost trackeru, který navazuje spojení a funguje jako prostředník.

Magnet Link obsahuje pouze základní URI (Uniform Resource Identifier) řetězec „magnet:?xt=urn:sha1“, kde následuje konkrétní hodnota hashe. Jedná se o identifikátor zdroje, který slouží k přesnému určení zdroje dat. Jelikož se jedná o tak malé informace, velikost nepřesahuje 1 kB oproti 50 kB torrentových souborů.

Jedna z nevýhod ML je také to, že informace, jako IP adresy poskytovatelů souborů, musí získat sám. K tomuto „sběru“ informací pomáhají distribuované hashovací tabulky (DHT). Tím pádem dochází ke zpomalení průběhu stahování, než jsou získány potřebné informace, které dříve byly umístěné v souboru torrentu.

Pokud jsou data a potřebné informace stažené z jiných klientů (DHT, Peer Ex-

change), začíná stahování samotného souboru. Tento proces může být urychlen přidáním zakódovaných IP adres do URI souboru. Rychlost stahování však zůstává nadále podobná, ne-li stejná jak u Torrentu[4].

Magnet Link se skládá z několika částí.

- **magnet**

Identifikátor spojení Magnet Linku.

- **?xt=urn:btih**

Definuje informaci hashe BitTorrentu, Edonkey má například tvar:

`xt=urn:ed2k.`

- **5dee65101db281ac9c46344cd6b175cdcad53426**

Jedná se o obsah hashe(vstupní data se převádí do relativně malého čísla).

- **&dn=name**

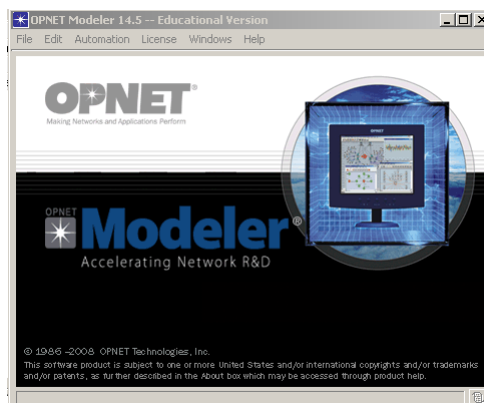
Název souboru

4 OPNET MODELER

Firma OPNET Technologies, Inc. vytvořila program Modeler Accelerating Network R&D. Program je vytvořen k simulování, navrhování a analýze datových sítí. Pracuje v prostředí IT Guru a práce se bude vytvářet v OPNET Modeler verzi 14.5.

Aplikace je také ve studentské verzi, již zmíněné IT Guru, kde po registraci na internetových stránkách je možné stáhnout instalační soubor. Jedná se o prostředí, kde již probíhá samotný návrh, simulace a veškeré práce ohledně sítí. Program je velmi dobře propracován a také přehledný 4.1.

Pokud by však chtěl někdo pracovat více do hloubky, je potřeba se naučit principy a možnosti, které program nabízí. V tomto simulačním programu je nepřehledné množství použitelných prvků, nastavení a mohou se vytvářet tzv. scénáře, od malých domácích sítí až po velké WAN sítě. Na provoz sítí může být použito mnoho protokolů včetně IPv6, různé verze RIP, OSPF protokolů a spoustu dalších.



Obr. 4.1: Úvodní okno pro Opnet Modeler.

5 PRVNÍ SIMULACE - VOIP, VIDEO

Kapitola se bude zabývat přenosem videa, audia a FTP, ve vytvořené síti, která bude simulovaná v programu OPNET. Bude zde ukázán přesný postup při tvorbě scénáře, jakožto nastavení aplikací, profilů a jejich implementace do koncových zařízení.

Pro tento případ je vybrána mapa „Campus“ a na dalším kroku bude nastaven rozměr pracovní plochy. Jako měrná jednotka se může použít kilometr, míle, stopy a další. Ostatní nastavení bude ponecháno, čtverec o délce 10 km a stiskne se na „Next“.

V posledním kroku je shrnutí celého projektu, hlavně použité technologie a mapa. Po potvrzení se dokončí úvodní nastavení projektu a bude ukončena část s vytvářením projektu.

Zobrazí se dvě okna, jedno okno bude obsahovat pracovní plochu a v druhém okně budou záložky s prvky.

V paletě prvků jsou umístěny veškeré prvky. Zobrazí se po rozkliknutí „Shared Object Palettes“ to jsou prvky, které jsme si nedávali do rychlé volby. Záložka „Network's Private Pallete“ obsahuje již zvolené prvky.

5.1 Vytvoření sítě

Do pracovní plochy bude vloženo 6 routerů (ethernet4_slip8_gtwy). Pokud nebudou k nalezení ve vytvořeném paletovém seznamu, můžou být prvky vyhledány podle názvu. Routery se spojí pomocí linky PPP_DS1, jak je zobrazeno na obrázku 5.1.

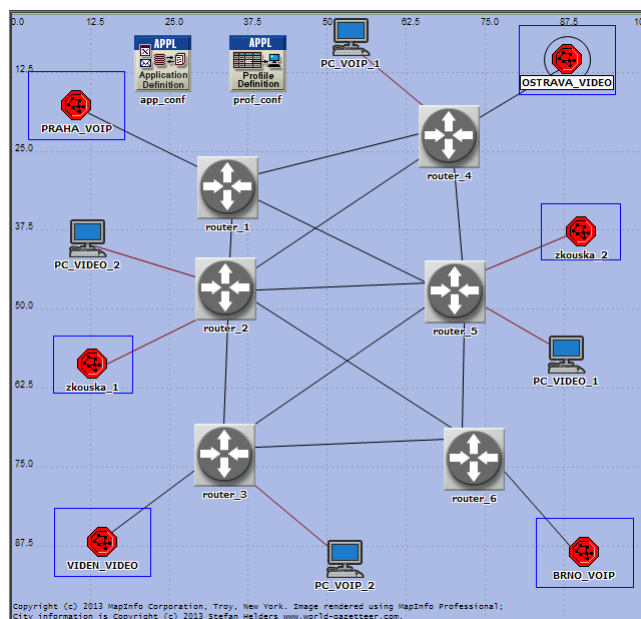
Pro lepší práci s koncovými stanicemi se vytvoří podsítě (subnet), kam umístíme několik stanic. Tyto subnety jsou přímo v okně s prvky. Koncové stanice budou vloženy přetažením z palety prvků, vloží se tedy 4 stanice (ethernet_wkstn).

Pro vstup do subnetu, stačí 2x poklikat myší na ikonu, nebo použít pravé tlačítko a kliknout na „Enter subnet“. Subnet musí obsahovat také router, obr. 5.2, pro propojení s vnější sítí, bude také propojen linkou PPP_DS1.

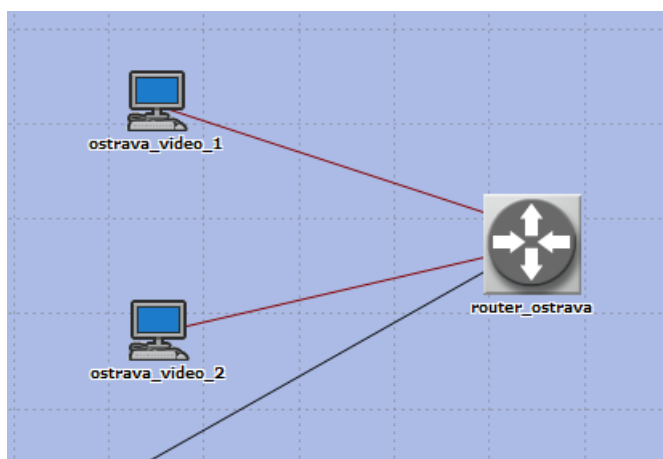
Spojení mezi routery a koncovými stanicemi se provede linkou 100BaseT.

5.2 Application configure

V první řadě bude nastavena konfigurace v aplikačním rozhraní. Vstup do atributů je proveden tak, že bude vyvolána nabídka (pravým tlačítkem na ikonu) a vybere se „Edit Attributes“, poté se zobrazí možnosti s nastavením viz 5.3.



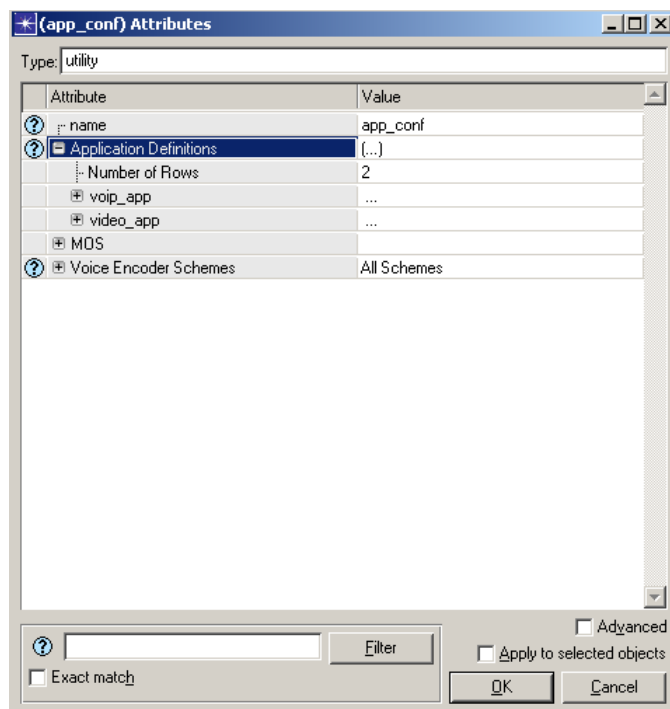
Obr. 5.1: Vytvořená pracovní síť.



Obr. 5.2: Obsah v subnetu - OSTRAVA_VIDEO.

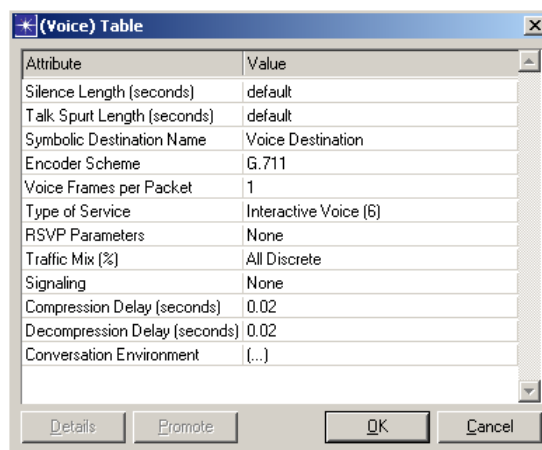
Rozklikne se nabídka App Definitons, jelikož se bude simulovat přenos videa a audia, vybere se hodnota „Number of Rows“ 2, pokud by se simuloval další přenos, stačí zvětšit hodnotu na 3, předchozí nastavení se ponechá. Pokud se bude odebírat, tak se vždy vyruší nastavení s nejvyšším číslem.

Každý řádek se pojmenuje, nejlépe podle přenášených dat. Podle obrázku jsou řádky pojmenovány na `voip_app` a `video_app`. V dalším kroku se nastaví již data, která se budou využívat a jejich kvalitu. V záložce „Description“ nastavíme řádek „Voice“ nebo „Video Conferencing“, podle toho který profil se bude nastavovat.



Obr. 5.3: Základní menu v Aplikační konfiguraci.

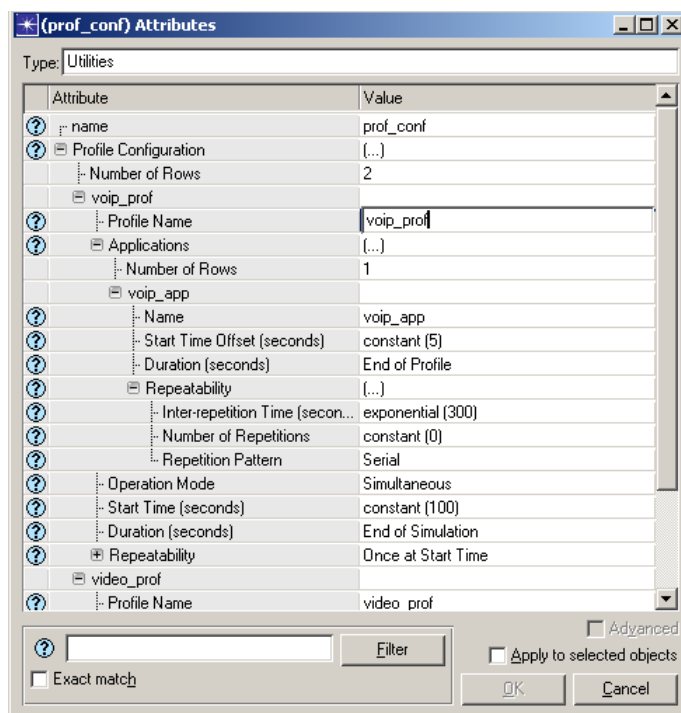
Na obrázku 5.4 je zobrazeno okno s nastavením parametrů. Nastavení, která se budou měnit v případě dané sítě, jsou:



Obr. 5.4: Nastavení audio přenosu.

Stejné nastavení se provádí i s aplikací pro video konference. Bude vybrána však nejnižší kvalita hovoru, pro rychlejší simulaci.

Jakmile budou veškeré údaje v „app_conf“ nakonfigurované, může se nastavit „prof_conf“. Postup bude téměř stejný. Jedná se o přiřazení vytvořených aplikací k profilům a jsou zde nastavení, které se zaměřují na simulace, jako je začátek simulace nebo její konec. Nastavení se provede podle obrázku 5.5, kdy v záložce „Applications“.



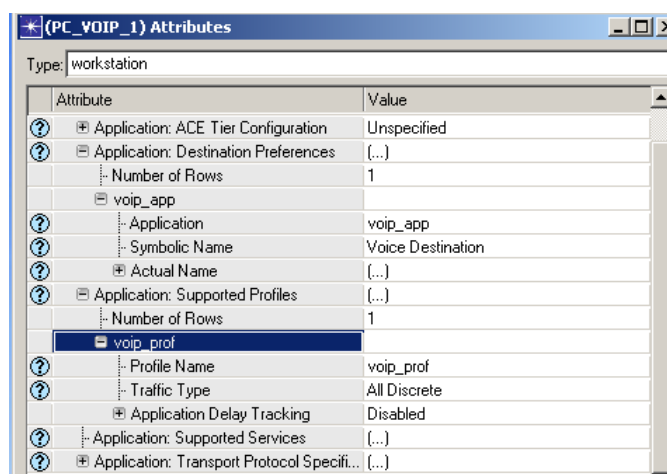
Obr. 5.5: Nastavení atributů v „prof_conf“.

Implementace nastavení do koncových stanic se provede v atributu stanice. Příklad bude ukázán u stanice, která bude komunikovat s jinou stanicí přes VoIP. K tomu slouží nastavení několika parametrů podle obr. 5.6.

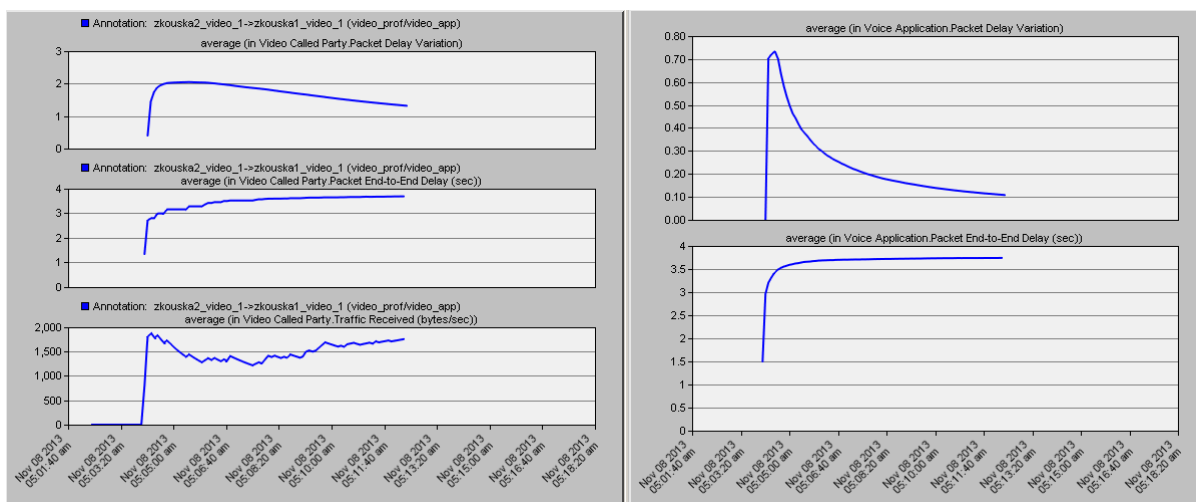
Stejným způsobem se nastaví veškerá koncová zařízení, která jsou v síti a subnetch.

Po nastavení veškerých stanic, budou vybrány protokoly, datové toky, případně další nastavení, která budou simulována. V tomto scénáři **Video Conferencig a Voice**.

Na obrázku 5.7 jsou zobrazeny grafy, které ukazují přenos mezi koncovým uživatelem a routerem. Můžeme vidět, že protokol využívající hlasové služby s postupem času zlepšuje přenos. Zpoždění klesá oproti video konferenci rychleji.



Obr. 5.6: Nastavení atributů v „PC_VOIP“, nastavují se zde vytvořené profily.



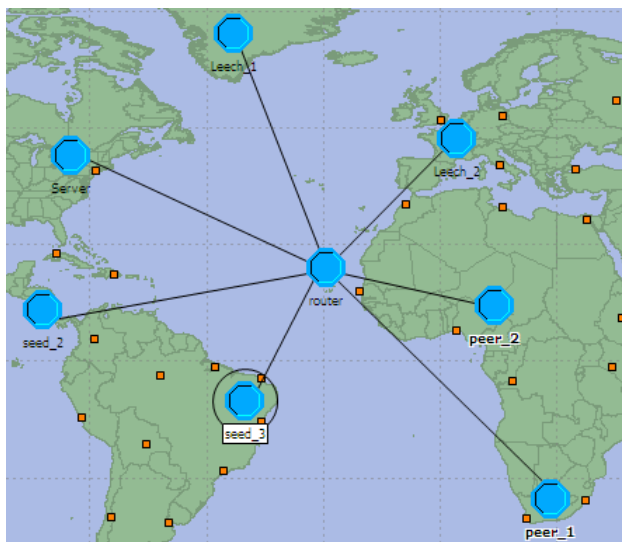
Obr. 5.7: Znáznornění přenosu Video a Audio v síti.

6 PROTOKOL BITTORRENT

Při tvorbě protokolu BitTorrent bude vytvořen nový projekt, kde se umístí „Node“ klienti, kteří budou programováni na určité prvky s danými vlastnostmi. Pro každý možný přenos budou vloženi dva klienti (peer, seed, leech) a také aktivní prvek hub nebo se může použít také router, jak je zobrazeno na obrázku 6.1. Tvorba sítě bude až v poslední části.

Zařízení mezi sebou nebudou ještě propojeny linkou, která bude vytvořena a nastavená na určitý přenos.

Veškeré soubory, které budou vytvořeny, se budou ukládat do složky s projektem. Je nutné při simulaci přenastavit výchozí hodnoty právě na ty, které budou vytvořené.



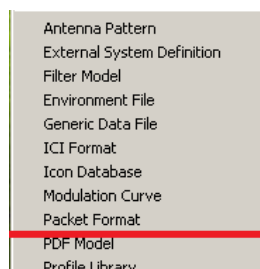
Obr. 6.1: Schéma zapojení pro využití protokolu BitTorrent.

6.1 Vytvoření paketu

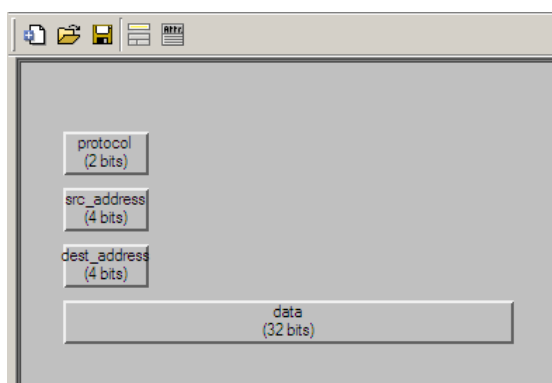
Tvorba paketu je jednoduchá a vytváří se tak, že při spuštění Opnet Moedeleru bude vybráno „New“ a místo nového projektu tvorba paketu 6.2.

Po vytvoření se otevře prázdné okno, kde bude ikona na tvorbu jednotlivých bloků paketu.

Bude navrhnuté blokové schéma, kde bude obsažena adresa výchozí a cílové stanice, data a využitý protokol 6.3.



Obr. 6.2: Výběr na tvorbu paketu.



Obr. 6.3: Vytvořený paket pro odpověď BitTorrentu.

Jelikož každá situace, která nastane v síti, má jiné složení paketu, bude se muset vytvořit paket pro každý případ samostatně. Jedná se o dotazy na BitTorrent a FTP server a jejich odpovědi.

6.2 Uživatel - Seed

Jakmile se začnou vytvářet pracovní stanice, bude nutné vytvořit dva soubory. Jeden ze souborů bude vytvářet „Process“ model a druhý „Node“. Vytvoří se pomocí stejného postupu jako při tvorbě paketu, ale vybere se „Process Model“ nebo „Node Model“. Pro první případ vybereme „Node Model“, který se nastaví na pracovní stanici „Seed“.

6.2.1 Node Model

Node model se může chápat jako schematické zapojení, kde je znázorněné posílání a přijímání paketu. Vytvoří se pomocí vložení procesorů, vysílače a přijímače.

Poté se části spojí spojovacím vedením, v tomto případě „Packet Stream“ (modrá spojovací čára), kde záleží na to, kam směřuje šipka 6.4.

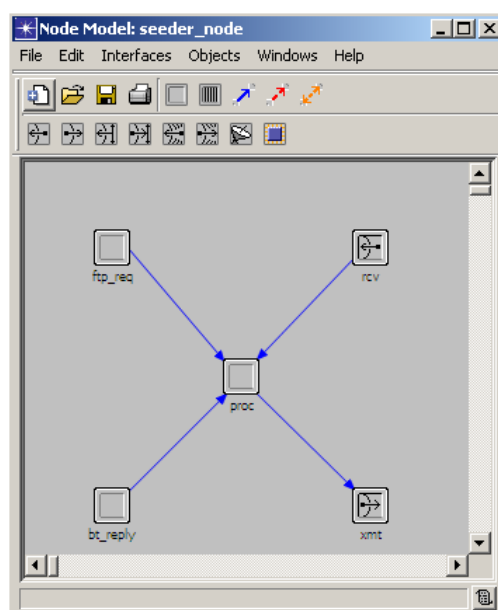
Po vytvoření blokového zapojení se nadefinují vlastnosti a přiřadí se vytvořené modely pro jednotlivé koncové stanice daným blokům. Centrální procesor se nastaví v pozdější části.

Nyní se bude měnit nastavení s bloky vytvářející pakety „ftp_req“ a „bt_reply“. Nastavení procesu bude výběrem „simple_source“. Následně se přidá řádek s formátem paketu a z nabídky bude vybrán příslušný vytvořený paket. Změní se „begsim_intrpt“ na „enable“, docílí se tím to, že při startu simulace bude použito nastavení, které bylo vytvořeno [12].

Z nabídky „Interfaces“ bude vybrána možnost „Node Interfaces“. Povolí se pouze záložka „fixed“ a v druhém okně bude změněno vše na „hidden“, kromě vytvořených paketů.

Tento postup bude pro všechny stanice stejný. Pouze se budou měnit blokové schémata, pakety, které budou vytvářeny a následně jejich vysílání a přijímání.

V atributech vysílače a přijímače se změní nastavení formátů paketů. Veškeré ostatní formáty nebudou podporovány. Na vysílač budou povoleny jen ty pakety, které budou opouštět stanici a na příjem ty, na které stanice může odpovědět [13].



Obr. 6.4: Blokové schéma koncové stanice - Seed.

6.2.2 Process Model

Aby mohl proces fungovat, dvakrát se klikne na centrální procesor v „Node Model“ a následně se objeví prázdné okno. Tím se dostaneme do „Process Model“, kde se vloží dva prvky a zapojí se podle obrázku 6.5.

V horní liště jsou umístěné ikony, které budou použité v dalším postupu, při programování jednotlivých částí.

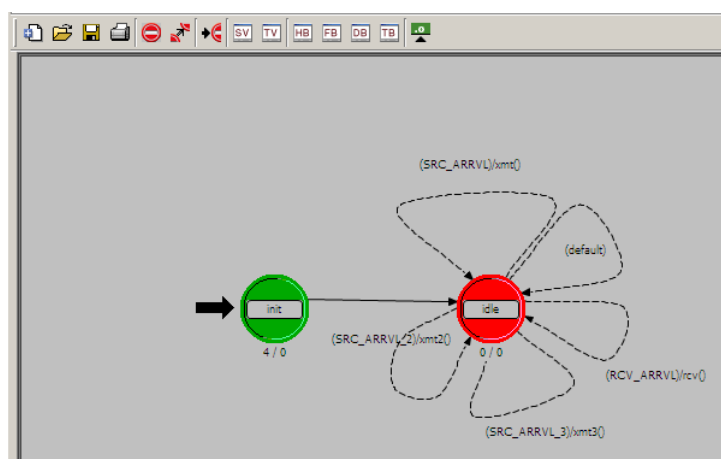
V této části se vytvoří dva stavy, záložka „Create State“, kde jeden stav bude představovat INIT a druhý IDLE. Jedná se o takzvané nevynucené stavy. Pokud bychom chtěli, aby jeden ze stavů byl vynucený, v tomto případě INIT, použije se nastavení z nabídky „Set Initial State“, popřípadě pravým tlačítkem a vybere se „Make Initial State“.

Poté se spojí pomocí „Create Transition“, kde opět záleží na směru. Na stav IDLE vytvoříme smyčky, které budou představovat jednotlivé funkce, které se budou vykonávat podle požadavků.

Stejným způsobem, jako při spojování stavů, vytvoříme pět smyček, které budou vycházet a poté se vracet do IDLE části.

Jedna smyčka bude popsána jako „default“, jedná se o nastavení, do kterého se nebude zasahovat. Další části budou „SRC_ARRRVL“, „SRC_ARRRVL_2“, „SRC_ARRRVL_3“, kde ke každé smyčce bude daná poznámka, se kterým výstupem („xmt“) bude spojena. Jako poslední bude opět „SRC_ARRRVL“ pouze s poznámkou „rcv“, což bude přijímač. **Tímto způsobem se zapojí i nastavení v „Peer“ a „Leech“**, nemusí být vytvořen zvláštní soubor, stačí použít již ten vytvořený [13], [12].

V rozhraní „Global“ budou nadefinována jména požadovaných statistik.

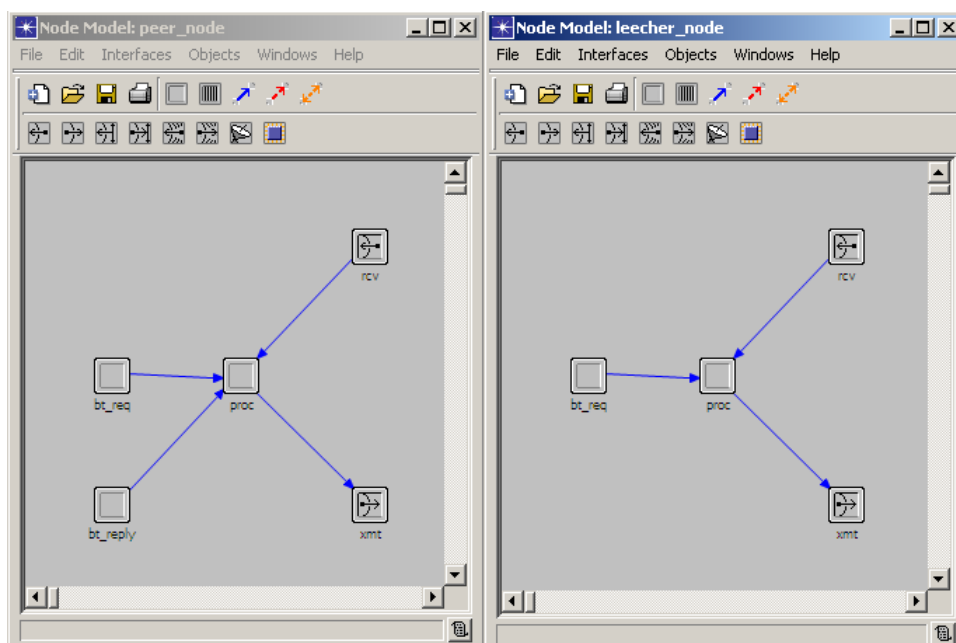


Obr. 6.5: Zapojení procesu - platí pro všechny koncové stanice

Nyní se vrátíme k „Node_Model“ a pravým tlačítkem se klikne na centrální proces, vybere se „Attributes“ a z „process model“ bude přiřazen již vytvořený proces soubor pro danou stanici, v tomto případě se jedná o „seeder_proc“, pro všechny ostatní stanice se přiřadí stejný proces. Změněno bude také nastavení „begsim“ na hodnotu „enable“.

6.3 Uživatel - Peer

Tento uživatel je zapojený do sítě a poskytuje pouze data, které má již stažené, proto bude jeho zapojení odlišné. Sdílení bude stejné jako u uživatele „Seed“, bude tedy obsahovat procesor, request a reply bloky a jelikož potřebuje ještě stahovat od ostatních uživatelů, přidá se blok „bt_req“. Tento blok bude posílat požadavky na stahování dat 6.6.



Obr. 6.6: Zapojení Node Model pro uživatele Peer (vlevo) a Leech (vpravo).

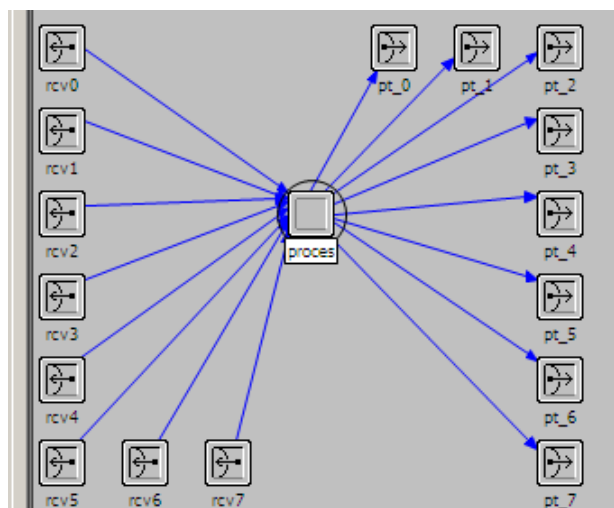
6.4 Uživatel - Leech

Jedná se o uživatele, který po dostahování ukončuje spojení. Z důvodu ukončení nebude obsahovat blok na sdílení, ale pouze na požadavky 6.6.

6.5 Router

Zapojením aktivního prvku bude provedeno přidáním libovolného počtu přijímačů a vysílacích bloků a jednoho centrálního „řídícího“ bloku, počet však nesmí být menší, než počet uživatelů v síti. Pro připojení k centrálnímu procesu musí směřovat šipky z přijímačů a odchozí k vysílacům 6.7. Nastavení procesu bude provedeno podobně jak v koncových stanicích. Pouze v „Node Interfaces“ bude povolena položka „fixed“ [12].

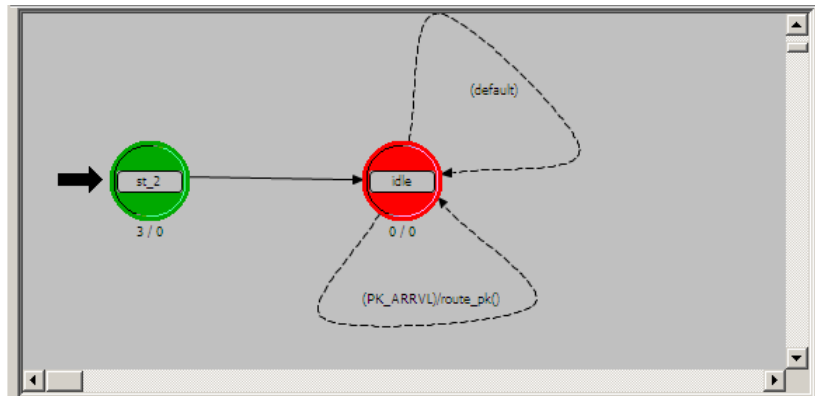
V dalším kroku se označí veškeré přijímače a vysílače, otevře se nabídka atributů (nesmí se zapomenout označit „Apply to selected objects“) a všem prvkům bude přiděleno povolení na přenos pouze těch paketů, které byly vytvořeny pro konkrétní simulaci, ostatní budou zakázány.



Obr. 6.7: Zapojení Node Model aktivní prvek - Router.

6.5.1 Process Model

V tomto modelu budou vloženy dva stavy. IDLE stav bude obsahovat pouze dvě smyčky, jedna „default“ a druhá „PK_ARRVL“ 6.8.



Obr. 6.8: Zapojení Process Model pro Router.

6.6 Tvorba komunikačního vedení

Linka bude nastavena na duplexní přenos, ostatní přenosy budou zakázány. Z nabídky „File“ a následným otevřením „Declared External Files“ bude povolena „link_delay“. V attributech nastavíme vše podle obrázku, budou povoleny veškeré pakety [12].

Add Delete

Attribute Name	Status	Initial Value
data rate	set	9,600
delay	set	0.0
ecc model	set	ecc_zero_err
error model	set	error_zero_err
financial cost	set	0.00
line style	set	solid
packet formats	set	bt_reply_packet, bt_req_packet, ftp_reply_pac
propdel model	set	dpt_propdel
role	set	
symbol	set	none
tag	set	
thickness	set	1
txdel model	set	dpt_txdel
user id	set	0

Link Type	Supported	Palette Icon
ptsimp	no	
ptdup	yes	dup_pt_lk
bus	no	
bus tap	no	

Declared External Files

Select included entries:

- ☐ isis_tlv_support
- ☐ jammer_auto_addr
- ☐ lacp_notif_log_support
- ☐ lane_auto_addr
- ☐ lane_gw
- ☐ lane_notif_log_support_v2
- ☐ lane_v2
- ☒ link_delay
- ☐ llm_support
- ☐ llms_notif_log_support
- ☐ manet_support
- ☐ manet_tora_imep
- ☐ mframe_api
- ☐ mframe_supp
- ☐ mipv6_signaling_sup
- ☐ mipv6_sup

Obr. 6.9: Nastavení vlastností pro linku spojující prvky.

7 PROGRAMOVÁNÍ JEDNOTLIVÝCH ČÁSTÍ

Veškeré nastavení provedené v „Node Model“ se projeví do ostatních stejných prvcích.

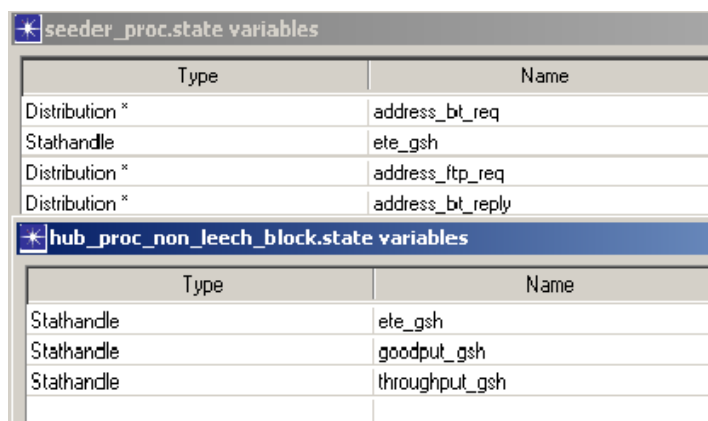
Programování jednotlivých částí, jak koncových stanic, tak i aktivního prvku, bude opět po jednotlivých částech. Každá koncová stanice (dvojice) má odlišnou funkci, tudíž i část s programem. Většina kódu a nastavení bude velmi podobná, proto bude důkladně vytvořen postup pro stanici „Seed“ a od toho se budou odvíjet ostatní prvky. Bude potřeba otevřít již vytvořené soubory s procesním schématem, kde budou psány jednotlivé funkce.

7.1 Seed, Peer, Leech, Router

Pomocí „Node Model“ se vyvolá nabídka „Process Model“ pro danou stanici, veškeré změny a kódy se projeví i v ostatních prvcích, které využívají daný „Node Model“. Jedná se o ikony „SV“, „HB“, „FB“.

- **State Variable**

Budou přidány datové třídy typu Opnet. Jedná se o „Stathandle“ a do jména se připsí „ete_gsh“, „goodput_gsh“, „throughput_gsh“. Jedna se o proměnné na měření přenosů. Toto nastavení bude platit pro router. V případě koncových stanic budou přidány typy „Distribution“ a jména budou podle dat, které se posílají nebo přijímají 7.1. Server bude obsahovat oproti stanicím odpověď na „FTP“ data.



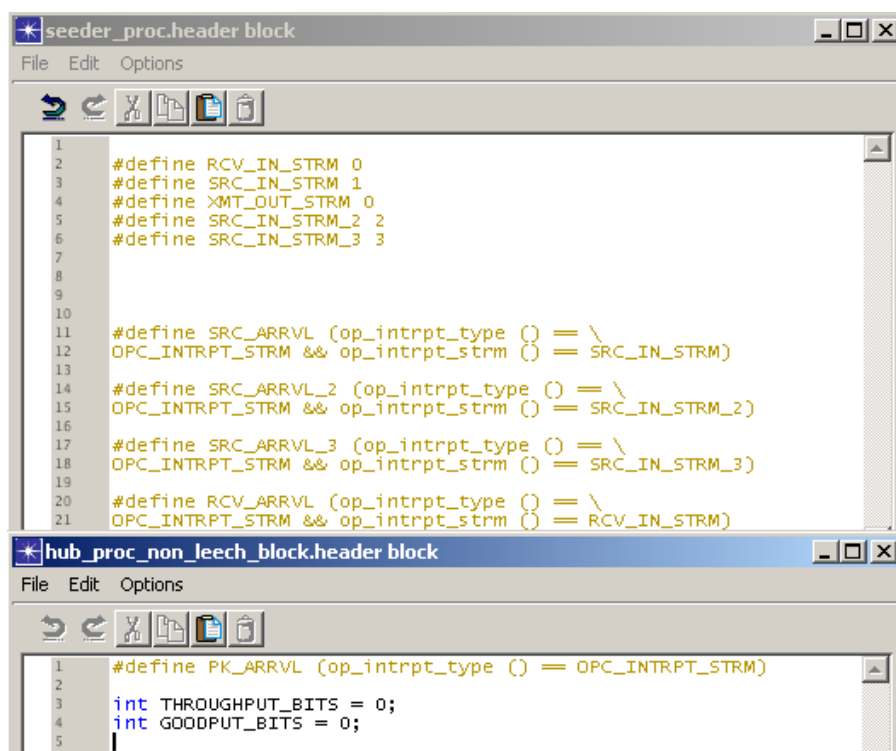
seeder_proc.state variables	
Type	Name
Distribution *	address_bt_req
Stathandle	ete_gsh
Distribution *	address_ftp_req
Distribution *	address_bt_reply

hub_proc_non_leech_block.state variables	
Type	Name
Stathandle	ete_gsh
Stathandle	goodput_gsh
Stathandle	throughput_gsh

Obr. 7.1: Definování proměnných v koncové stanici a routeru.

- **Header Block**

Budou nadefinovány jednotlivé proměnné, jako je vysílač a přijímač, na kterých „STRM“ streamech budou vysílány. Následně se nadefinují jednotlivé smyčky, které jsou vytvořeny v procesním modelu. Následně se jim přiřadí i výchozí stream 7.2. Pro router budou nadefinovány ještě dvě proměnné typu „INT“ s výchozí hodnotou 0 [14].



Obr. 7.2: Funkce jednotlivých smyček vytvořených v procesním modelu.

- **Function Block**

Jedná se o zdrojový kód. Z důvodu jeho rozsahu bude přidán spolu se závěrečnou prací na přenosném nosiči.

Obrázek 7.3 tvoří část kódu pro vysílač „xmt“. Pro další vysílače je zdrojový kód obdobný a pro „rcv“ bude nadefinován zápis pro graf a následné zničení paketu. Na dalším obrázku 7.4 je zobrazena část funkce pro router. Zde se vytváří proměnné pro tvorbu grafů a jejich následná úprava na požadovanou velikost a zápis[?].

```

static void xmt(void)
{
    int protocol;
    Packet * pkptr;
    FIN(xmt());
    pkptr = op_pk_get (SRC_IN_STRM);

    op_pk_nfd_set_int32(pkptr, "src_address", op_id_self());
    op_pk_nfd_get_int32(pkptr, "protocol", &protocol);
    if (protocol==0){
        printf("sending protocol 0");
        op_pk_nfd_set_int32(pkptr, "dest_address", (int)op_dist_outcome(address_ftp_req));
    }
    else if (protocol==1){
        printf("sending protocol 1");
        op_pk_nfd_set_int32(pkptr, "dest_address", (int)op_dist_outcome(address_bt_req));
    }
    else if (protocol==2){
        printf("sending protocol 2");
        op_pk_nfd_set_int32(pkptr, "dest_address", (int)op_dist_outcome(address_bt_reply));
    }
    op_pk_send (pkptr, XMT_OUT_STRM);
    FOUT;
}

```

Obr. 7.3: Část kódu pro „FB“, pouze pro „xmt“ vysílač.

```

static void route_pk(void)
{
    int dest_address;
    int node_id;
    int protocol;
    double GOODPUT;
    double THROUGHPUT;

    Packet* pktptr;
    FIN(route_pk());
    pktptr = op_pk_get(op_intrpt_strm());
    op_pk_nfd_get_int32 (pktptr, "dest_address", &dest_address);
    op_pk_nfd_get_int32 (pktptr, "protocol", &protocol);
    op_pk_nfd_get_int32 (pktptr, "src_address", &node_id);
    if (protocol ==2) //pokud je bt reply paket
    {
        if (dest_address ==4) //leech node
        {
            THROUGHPUT_BITS = THROUGHPUT_BITS + 1024*8;
            //spravna velikost paketu
            THROUGHPUT = THROUGHPUT_BITS/op_sim_time ();
            op_stat_write (throughput_gsh, THROUGHPUT);
        }
    }
}

```

Obr. 7.4: Část kodu pro „FB“, jedná se o router.

- **Enter Executives**

Do tohoto rozhraní se dostaneme pravým kliknutím na daný proces a vybere se „Edit Enter Execs“. Již vytvořené proměnné v kroku „SV“ budou nadefinovány podle obrázku 7.5.

```

address_bt_req = op_dist_load ("uniform_int", 0, 5);
address_ftp_req = op_dist_load("uniform_int", 6, 6);
address_bt_reply = op_dist_load("uniform_int", 2, 5);
ete_gsh = op_stat_reg ("ETE Delay", OPC_STAT_INDEX_NONE, OPC_STAT_GLOBAL);

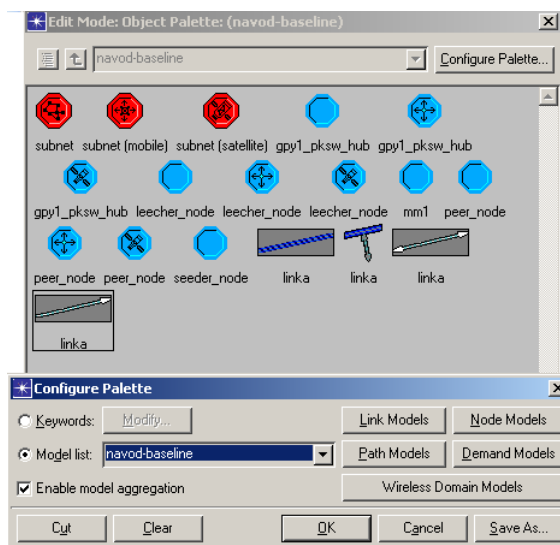
```

Obr. 7.5: Zapsání parametrů pro pakety a graf.

8 VLOŽENÍ PRVKŮ DO SCÉNÁŘE

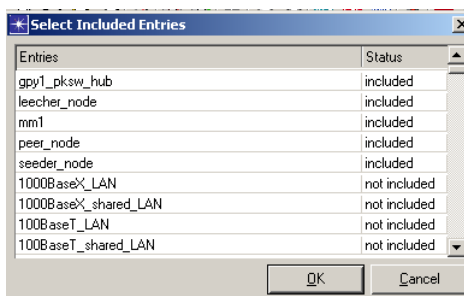
Aby bylo možné používat vytvořené stanice a ostatní prvky, musí být nahrány do menu s prvky.

Otevře se hlavní menu s prvky a v levé horní části se klikne na ikonu „Open palette in Icon View“. Bude otevřeno nové okno, poté se pokračuje na konfiguraci. Pokud jsou již obsažené některé prvky v menu, budou vymazány stisknutím tlačítka „Clear“, 8.1.



Obr. 8.1: Vyvolání nabídky na přidání vytvořených prvků.

Z nabídky „Link Models“ a „Node Models“ jsou vybrány veškeré prvky, co byly vytvořeny tak, že se změnil jejich stav z *not included* na *included* 8.2. Poté může být soubor uložen.



Obr. 8.2: Vložení prvků - změna z *not included* na *included*

Nyní jsou nahrány prvky pro tvorbu sítě. Vloží se libovolný počet prvků, avšak nesmí být napojeno víc prvků na router, než dovoluje nastavení a vytvoří se síť. V dalším kroku se přistoupí k samotné simulaci.

9 VÝSLEDKY BAKALÁŘSKÉ PRÁCE

9.1 Simulace BitTorrent protokolu

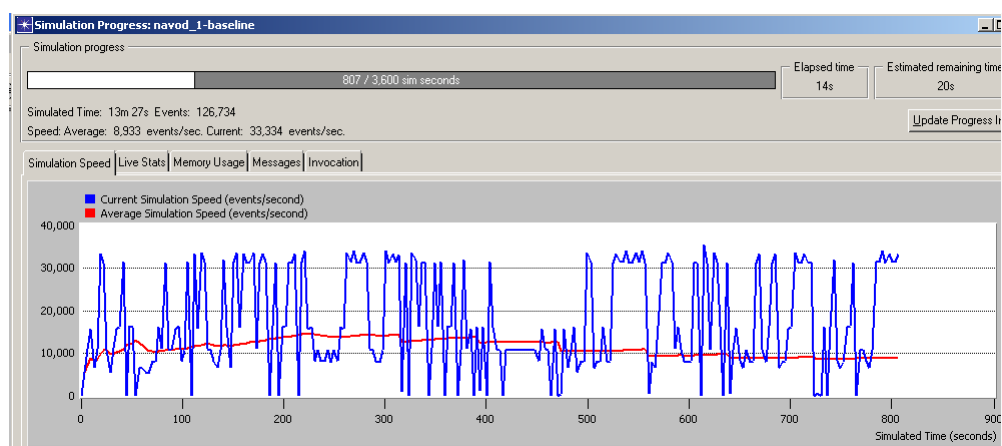
Veškeré nastavení, které bylo provedeno v předchozích kapitolách, bude nastaveno a vytvoří se síť. Zapojení bude obsahovat jeden router, server a libovolný počet stanic, které se budou odlišovat funkčností. Budou sdílet nebo stahovat data.

Před samotnou simulací budou nastaveny statistiky pro grafy.

V simulaci nebude nastavena žádná zásadní změna, můžou se upravit hodnoty času a počty údajů, které se budou vyhodnocovat.

9.2 Výsledky simulace

V průběhu simulace je možno sledovat vytížení sítě v první záložce 9.1. Po skončení simulace se budou porovnávat výsledky vytížení sítě u jednotlivých tras mezi koncovými stanicemi a routerem.

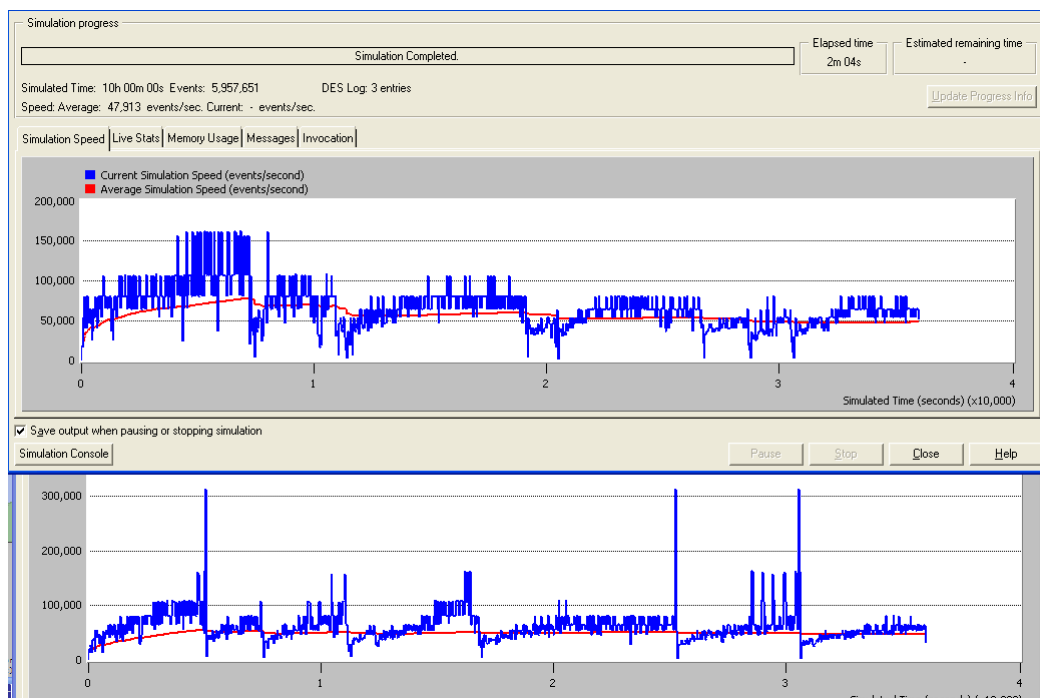


Obr. 9.1: Zobrazen průběh simulace - základní scénář

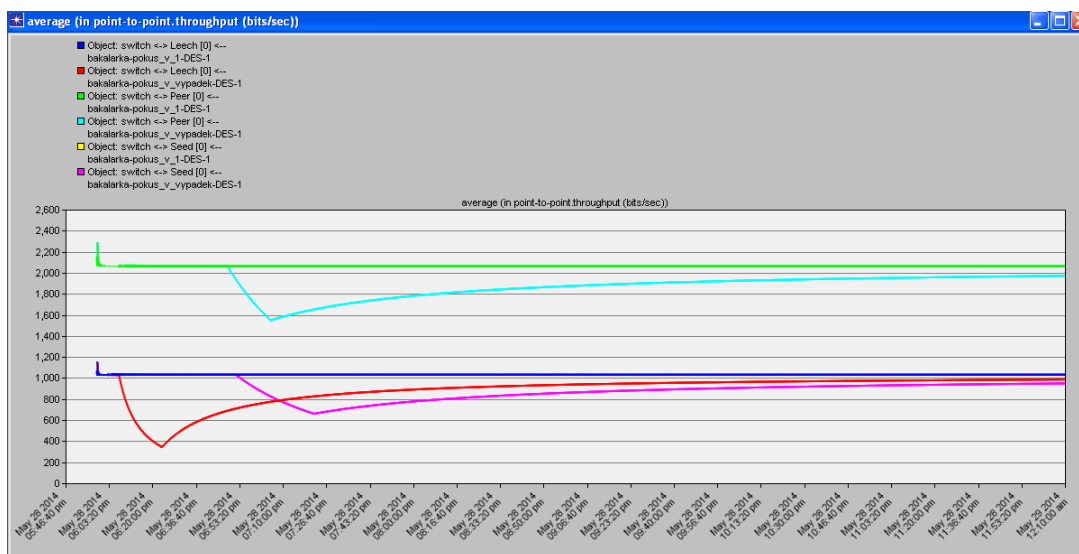
Pro porovnání byl vytvořen duplicitní scénář, kde vznikl výpadek sítě. Tento výpadek byl následně opraven pro každý jednotlivý druh stanice zvlášť, z důvodu srovnání grafů 9.2.

Po skončení simulace budou zobrazeny statistiky pro hodnoty „throughput“, mezi aktivním prvkem(AP) a koncovou stanicí, na které byl v jednom ze scénářů nadefinovaný výpadek a také originální scénář. Vše bude zobrazeno v jednotném grafu 9.3.

Z následujícího grafu 9.4 lze vyčíst využití a propustnost sítě (utilization a throughput). Největší hodnoty má uživatel Leech, který pouze stahuje od všech ostatních



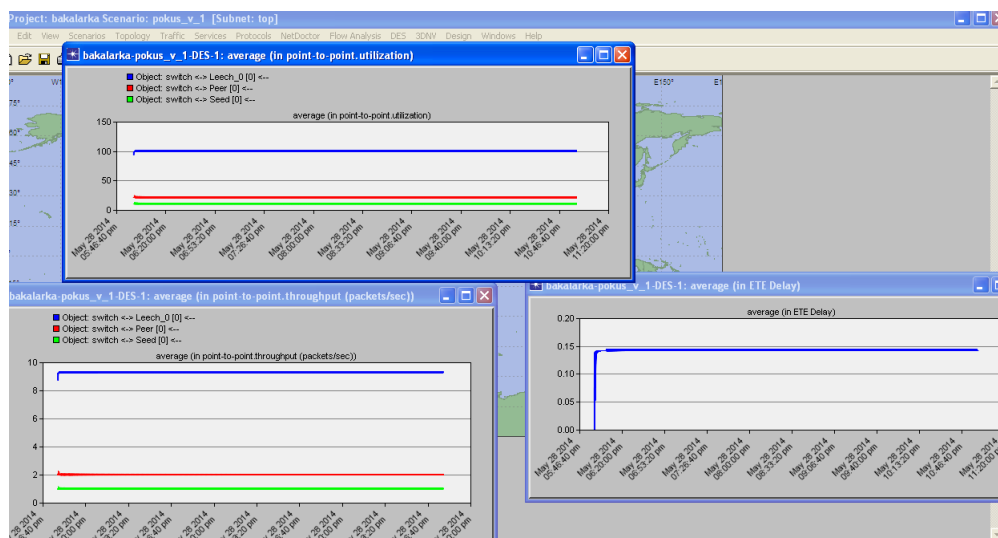
Obr. 9.2: Zobrazen průběh simulace - základní scénář (nahore), výpadek sítě (dole)



Obr. 9.3: Výsledky Simulace - zobrazení a porovnání grafů s nadefinovaným výpadkem sítě mezi AP a koncovou stanicí

uživatelů. Nejmenší využití má uživatel Seed, jehož jedinou funkcí je sdílet data, kdežto Peer sdílí a také stahuje. Tyto funkce se odvíjí také do zpoždění, které stoupne na danou hodnotu a drží se na jedné úrovni. Graf byl odsimulován pouze se základ-

ním scénářem.



Obr. 9.4: Zobrazen průběh simulace - základní scénář, zobrazení utilization, throughput a delay, mezi prvky a AP

10 ZÁVĚR

Cílem bakalářské práce bylo vytvořit a následně odsimulovat virtuální síť s protokolem BitTorrent. Základní poznatky a práce s programem Opnet Modeler bylo řešeno v semestrálním projektu. Daná práce byla zaměřena na vytváření prvků koncových stanic, routeru, popřípadě i FTP serveru.

Pro správné fungování byly vytvořeny pakety, které se následně posílaly přes router k daným stanicím. Aby bylo možné komunikovat mezi prvky, nadefinovalo se vedení, přes které probíhal přenos.

V simulaci jsou tři druhy koncových stanic, které se liší hlavně posílaným paketem. Jejich blokové schéma bude odlišné podle funkce, kterou by měla stanice vykonávat (sdílení, stahování nebo jejich kombinace). Hlavní část je tvořena v Process Modu. Procesní mod řeší veškerou funkci stanice. Jedná se o nejdůležitější část, kde je vytvořena komunikace v procesoru, který je umístěn v „Node Model“. Nesmí se zapomenout také nadefinovat funkce pro zaznamenávání dat a tvorby grafů.

Po nastavení a propojení sítě byla následně odsimulována funkčnost protokolu. Vyberou se statistiky, které se mají zaznamenávat do výsledků. Pro další příklad je možné vytvořit scénář s výpadkem sítě a následným obnovením pro porovnání vytíženosti linky.

Největším problémem nastává v zabalení projektu do souboru *.opcfa, který slouží k přenosu celého projektu. Je možné, že po rozbalení jsou některé funkce nekompletní, nastavení, které bylo provedeno, bude v původních hodnotách. Může se stát, že po úpravě do požadovaných stavů, nebude možné odsimulovat scénář.

Další problém, který nastal, bylo znovunačtení projektu. Po úspěšné simulaci a následném ukončení programu byly potřeba vykonat úpravy. Z toho důvodu byl hlavní scénář duplikován, pro zachování zálohy. Před samotnou změnou byla provedena simulace. Program nebyl schopen odsimulovat duplikovaný a posléze ani originální scénář.

Problémy, které nastaly při simulacích, mohou být tvořeny již starší verzí programu, nebo také provádění simulací ve virtuálních strojích.

LITERATURA

- [1] MIKLE, Ondrej. [Http://Root.cz](http://root.cz). *Seriál Techniky skryté v p2p sítích* [online]. 2006, 24.7.2006 [cit. 2013-10-20]. Dostupné z URL: [<http://www.root.cz/serialy/techniky-skryte-v-p2p-sietach/>](http://www.root.cz/serialy/techniky-skryte-v-p2p-sietach/).
- [2] KOEGEL BUFORD, John F, Hong Heather YU a Eng Keong LUA. *P2P networking and applications*. Burlington: Morgan Kaufmann, c2009, xxi, 415 s. The Morgan Kaufmann series in networking. ISBN 978-0-12-374214-8.
- [3] Torrent how to: Pojmy. [Http://torrent.howto.cz/](http://torrent.howto.cz/). *Torrent.howto: Pojmy* [online]. 2013 [cit. 2013-10-23]. Dostupné z URL: [<http://torrent.howto.cz/pojmy/>](http://torrent.howto.cz/pojmy/).
- [4] BRINKMANN, Martin. [Http://www.ghacks.net/](http://www.ghacks.net/). *What Is A Magnet Link And How Does It Differ From Torrents?* [online]. 2010 [cit. 2013-11-01]. Dostupné z URL: [<http://www.ghacks.net/2010/06/05/what-is-a-magnet-link-and-how-does-it-differ-from-torrents/>](http://www.ghacks.net/2010/06/05/what-is-a-magnet-link-and-how-does-it-differ-from-torrents/).
- [5] VACULÍN, Jan. [Http://www.intelekt.cz/](http://www.intelekt.cz/). *ORiNOCO: Wifi mesh sít* [online]. 2008 [cit. 2013-11-16]. Dostupné z URL: [<http://www.intelek.cz/art_doc-9213631AF330CB22C12573ED00467382.html/>](http://www.intelek.cz/art_doc-9213631AF330CB22C12573ED00467382.html/).
- [6] ZANDL, Patrick. [Http://www.marigold.cz/](http://www.marigold.cz/). *Mesh sítě: P2P architektura v bezdrátových sítích*. [online]. 2003 [cit. 2013-11-16]. Dostupné z URL: [<http://www.marigold.cz/item/mesh-site-p2p-architektura-v-bezdratovych-sitich/>](http://www.marigold.cz/item/mesh-site-p2p-architektura-v-bezdratovych-sitich/).
- [7] Wifi Praha. PUBLIX. [Http://wifi.praha-mesto.cz/](http://wifi.praha-mesto.cz/). *Bezdrátová Praha* [online]. 2003 [cit. 2013-11-16]. Dostupné z URL: <http://wifi.praha-mesto.cz/zdroj.aspx?typ=2&Id=4&sh=75096>.
- [8] Gnutella. FURTADO, Jonas a Bravo VINICIOS. [Http://www.gta.ufrj.br/grad/06_1/p2p/index.html/](http://www.gta.ufrj.br/grad/06_1/p2p/index.html/). *Redes P2P* [online]. 2009 [cit. 2013-11-16]. Dostupné z URL: http://www.gta.ufrj.br/grad/06_/p2p/gnutella.html.
- [9] KLAŠKA, Luboš. Základní kvalitativní parametry sítě. *Svět sítí* [online]. 15.12.2006 [cit. 2014-01-01]. Dostupné z URL: <http://www.svetsiti.cz/rubrika.asp?rid=17&tid=325>.
- [10] HYNČICA, Ondřej. Sítě typu mesh. *Svět sítí* [online]. 2005, č. 12. [cit. 2014-01-01]. Dostupné z URL: http://www.odbornecasopisy.cz/index.php?id_document=30826.

- [11] KAUFMANN, Morgan. *Network Simulation using OPNET* [online].
London, 2006 [cit. 2014-06-02]. Dostupné z URL: <http://www.staff.city.ac.uk/~veselin/opnet/EPM775_lab.pdf>. Laboratory Assignment. City University School of Engineering and Mathematical Sciences.
- [12] SVENSSON, Tommy a Alex POPESCU *Development of laboratory exercises based on OPNET Modeler* [online].
Blekinge, 2003 [cit. 2014-06-02]. Dostupné z URL: <http://staff.ustc.edu.cn/~bhua/experiments/Lab_Exercices_Modeler.pdf>.
- [13] KAMAL, Ahmed. *Opnet* [online].
Iowa, 2013 [cit. 2014-06-02]. Dostupné z URL: <<http://www.ece.iastate.edu/~kamal/Opnet/lab7.pdf>>.
- [14] *Support riverbed* [online].
2013 [cit. 2014-06-02]. Dostupné z URL: <[https://support.riverbed.com/bin/support/static/doc/opnet/17.5.A/online/modeler_17.5_PL5/Programmers\\$\\$\\$20Reference/wwhelp/wwhimpl/common/html/wwhelp.htm#href=aaam_splash.html&single=true](https://support.riverbed.com/bin/support/static/doc/opnet/17.5.A/online/modeler_17.5_PL5/Programmers$$$20Reference/wwhelp/wwhimpl/common/html/wwhelp.htm#href=aaam_splash.html&single=true)>.
- [15] happylg123 *Communication-Mobile* [online].
2009 [cit. 2014-06-02]. Dostupné z URL: <http://read.pudn.com/downloads154/sourcecode/others/683047/pk_switch/my_pksw_nd_proc.pr.c__.htm>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

BT BitTorrent – síťový protokol na přenos dat.

VoIP Voice over Internet Protokol – přenos digitalizovaného hlasu.

FTP File Transfer Protocol – protokol pro přenos souboru.

P2P Peer To Peer – označení typu počítačové sítě.

TTL Time To Live – je to čas, který omezuje životnost paketu, nebo počet skoků přes aktivní prvek.

IP Internet Protokol – protokol síťové vrstvy.

PEERID Identifikátor v BitTorrent síti.

AP Access Point – přístupový bod.

QoS Quality of Service – zajišťuje stabilitu počítačové sítě, rezervaci a řízení datových toků.

FB Function Block – funkční blok.

HB Header Block – hlavičkový blok.

EE Enter Exec – vstupní část pro procesy v Process Modu.

SV State Variable – blok stavových proměnných.

SEZNAM PŘÍLOH

A Obsah CD

46

A OBSAH CD

Elektronická verze bakalářské práce – Blazek_Libor.pdf

Projekt bakalářské práce v OPNET Modeleru – bakalarskaprace.opcfa – balíček obsahuje veškeré soubory pro verzi OM 14.5.

Součástí je také zabalený projekt bakalarskaprace.rar. Je zde umístěna také bakalářská práce v souborech pro L^AT_EX a obrázky.