

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO  
KATEDRA INFORMATIKY

# BAKALÁŘSKÁ PRÁCE

Simulace protokolu BGP v simulátorech NS



2015

Milan Hümmel

## **Anotace**

*Síťové simulace jsou přínosným nástrojem nejenom v oblasti projektování počítačových sítí, ale především pomůckou k pochopení a ověření chování komunikačních protokolů. Bakalářská práce obsahuje základní popis teorie simulací, na kterou navazuje popis síťových simulátorů NS-2 a NS-3. Dále součástí této práce je popis směrovacího protokolu BGP, který je poté použit při simulaci počítačových sítí v simulátoru NS-3 a NS-2. Vyhodnocení simulací obsahuje popis chování protokolu BGP při zřizování vazeb mezi směrovači, předávání informací o dostupnosti sítí či aktualizaci směrovacích tabulek, dále ověření výběru cest při směrování pomocí vybraných atributů.*

Děkuji vedoucí bakalářské práce paní doc. Ing. Lence Motyčkové, CSc. za užitečnou metodickou pomoc, připomínky a trpělivost. Dále bych chtěl poděkovat manželce Soni za trpělivost a podporu.

# Obsah

<b>1. Úvod</b>	<b>7</b>
<b>2. Internet protokol</b>	<b>8</b>
2.1. Internetový protokol verze 4	8
2.1.1. Paket IPv4	8
2.2. Směrování	8
2.2.1. Dynamické směrování	9
<b>3. Směrovací protokol BGP(Border Gateway Protocol)</b>	<b>10</b>
3.1. Autonomní systémy	10
3.2. Protokol BGP	10
3.3. Zprávy protokolu BGP	11
3.4. Atributy cesty	12
3.5. Směrovací proces protokolu BGP	14
<b>4. Simulace počítačových sítí</b>	<b>17</b>
4.1. Modelování	17
4.1.1. Analytický přístup	17
4.1.2. Simulační přístup	17
4.2. Simulace	17
4.2.1. Prvky simulace	17
4.2.2. Časově závislá simulace	18
<b>5. Network Simulator 2 (NS-2)</b>	<b>19</b>
5.1. Protokoly a modely podporované NS-2	19
5.2. Základní objekty modelu	19
5.2.1. Uzel (Node)	20
5.2.2. Linka	20
5.2.3. Agenti	21
5.2.4. Aplikace	21
5.3. Error model	21
5.4. BGP++	21
<b>6. Network Simulator 3 (NS-3)</b>	<b>21</b>
6.1. Rozdíly mezi NS-2 a NS-3	22
6.2. Protokoly podporované v NS-3	22
6.3. Diskrétní simulace u NS-3	22
6.4. Základní objekty modelu	23
6.4.1. Uzel	23
6.4.2. Aplikace	23
6.4.3. Kanál	24
6.4.4. Síťové rozhraní	24

6.4.5. Topology Helpers . . . . .	24
6.5. Výstupní soubory a vizualizace . . . . .	25
6.6. Přímé vykonávání kódu . . . . .	25
6.6.1. Quagga . . . . .	26
<b>7. Praktická část</b>	<b>26</b>
7.1. Ověření základního chování protokolu BGP . . . . .	26
7.1.1. Instalace NS-3 . . . . .	27
7.1.2. Vytváření skriptu simulace . . . . .	28
7.1.3. Spuštění simulace . . . . .	33
7.1.4. Vyhodnocení simulace . . . . .	33
7.2. Chování BGP při nastavení atributů WEIGHT, LOCAL_PREF a MED . . . . .	40
7.2.1. Instalace NS-2 . . . . .	40
7.2.2. Vytváření skriptu simulace . . . . .	41
7.2.3. Spuštění simulace . . . . .	44
7.2.4. Vyhodnocení simulace . . . . .	45
<b>Závěr</b>	<b>50</b>
<b>Reference</b>	<b>51</b>

## Seznam obrázků

1.	Dělení dynamických protokolů . . . . .	9
2.	Směrování v autonomních systémech . . . . .	11
3.	Použití atributu AS_PATH . . . . .	13
4.	Ovlivnění výběru cest pomocí atributu LOC_PREF . . . . .	13
5.	Ovlivnění výběru cest pomocí atributu MED . . . . .	14
6.	Ovlivnění výběru cest pomocí atributu WEIGHT . . . . .	14
7.	Unicastový a multicastový uzel . . . . .	20
8.	Simplexní linka v NS . . . . .	20
9.	NS-3 uzel . . . . .	24
10.	Vrstvový model DCE pro základní a pokročilý operační mód v NS-3	26
11.	Návrh simulované sítě . . . . .	29
12.	Navazování spojení mezi peer směrovači AS3 a AS5 . . . . .	33
13.	Zpráva UPDATE vyslaná z AS5 do AS3 . . . . .	34
14.	Zpráva UPDATE vyslaná z AS2 do AS3 . . . . .	34
15.	Směrovací tabulka směrovače AS3. . . . .	35
16.	Zahájení vysílání z AS3 do AS6 . . . . .	35
17.	Přenos zprávy UPDATE mezi směrovači AS5 a AS3 s informacemi o nedostupnosti sítě. . . . .	36
18.	Navazování spojení mezi peer směrovači AS3 a AS5 . . . . .	36
19.	Přesměrování vysílání UDP paketů z AS3 přes peer směrovač AS2	37
20.	Přenos zpráv UPDATE mezi směrovači AS2 a AS4. . . . .	37
21.	Směrovací tabulka směrovače AS2 před a po aktualizaci. . . . .	38
22.	Distribuce zpráv UPDATE ze směrovače AS2 do směrovače AS3. .	39
23.	Směrovací tabulka směrovače AS8. . . . .	40
24.	Návrh simulované sítě . . . . .	42

# 1. Úvod

Jedním z hlavních témat této bakalářské práce, je problematika simulace počítačových sítí. V dnešní době, kdy si bez počítačových sítí nedokážeme představit svět, kdy projektanti počítačových sítí musí navrhovat takové řešení sítí, aby byly schopny zvládnout na tyto sítě rychle rostoucí nároky, jsou simulace a modelování počítačových sítí ekonomicky nejlepším řešením.

Druhým hlavním tématem bakalářské práce je popis a simulace směrovacího protokolu BGP. Internet, který je součástí většiny naší společnosti, je rozdělen na několik částí, kterým se říká autonomní systémy. Protokol BGP je v současné době standardem pro směrování mezi těmito autonomními systémy.

Bakalářská práce je rozdělena na dvě části, na část teoretickou a praktickou. V teoretické části se zabývá popsáním protokolu IP, rozdělením a typy směrovacích protokolů. Součástí teoretické části je popsání základních vlastností protokolu BGP.

Další část části teoretické se zabývá teorií simulací, za kterou následuje popis síťového simulátoru NS-2, který zatím podporuje větší množství protokolů než simulátor NS-3, o kterém pojednává poslední část části teoretické.

Praktická část je rozdělena na dvě části. První část se zabývá instalací simulátoru NS-3 s nástrojem pro přímé vykonávání kódu (DCE) a balíčkem Quagga, dále popisem vytvářeného simulačního skriptu s využitím směrovacího protokolu BGP a následným spuštěním. Poslední kapitola této části se zaměřuje na rozbor výsledku simulace, především na chování protokolu BGP, na navazování spojení, výměnu zpráv, aktualizaci směrovacích tabulek, chování při výpadku linky v průběhu přenosu informací, chování při přidání nové linky a chování při přidání nového směrovače.

Druhá část praktické práce se zaměřuje na ověření ovlivnění výběru cest při směrování pomocí atributů WEIGHT, LOCAL\_PREF a MED protokolu BGP pomocí simulátoru NS-2. Součástí této části je popis instalace simulátoru NS-2, dále popis vytvářeného simulačního skriptu a rozbor výsledku simulace.

## 2. Internet protokol

Internet protokol je v dnešní době nejpoužívanější komunikační protokol. Je základním protokolem pracujícím na síťové vrstvě používaným v počítačových sítích a Internetu [9]. Protokol neposkytuje záruky na přenos dat a rozlišuje pomocí IP adresy pouze jednotlivá síťová rozhraní. Existují dvě varianty protokolu IP:

- IP verze 4
- IP verze 6

Trvalejším řešením nedostatku IP adres verze 4 je nová verze protokolu IPv6, která má větší adresový rozsah ( $2^{128}$  adres). IPv6 je stále využívána jen menšinou zařízení připojených k internetu, ale postupně probíhá přechod na tento protokol.

### 2.1. Internetový protokol verze 4

IP protokol dopravuje data mezi dvěma libovolnými počítači či zařízeními v internetu, tj. i přes mnohé LAN. Data jsou od odesílatele k příjemci dopravována (směrována) přes směrovače. IP protokol je protokol, který umožňuje spojit jednotlivé lokální sítě do celosvětového Internetu. Protokol IPv4 poskytuje omezený adresní prostor - teoreticky  $2^{32}$  adres.

#### 2.1.1. Paket IPv4

Jedná se o základní jednotku přenášených dat. Skládá se ze záhlaví a přenášených dat. Celková délka paketu včetně záhlaví je vždy násobkem 32 bitů. Maximální délka paketu je 65535 oktetů [10]. IP paket je na úrovni síťového rozhraní vždy zabalen do rámce příslušné technologie (Ethernet, PPP, WiFi, atd.), který se mění, tak jak paket prochází přes dílčí síť. [6]

### 2.2. Směrování

Směrování představuje proces hledání cest z jednoho bodu do jiných bodů v rámci propojených sítí. Směrování zajišťují směrovače (routery), ale i koncové stanice (při vysílání) a jejich úkolem je doručit paket adresátovi, pokud možno co nejefektivnější cestou.

Každé síťové rozhraní, komunikující prostřednictvím protokolu IP, má přiřazený jednoznačný identifikátor - IP adresu. Na základě IP adres se ve směrovačích vytváří a udržuje směrovací tabulka. Směrovací tabulka obvykle obsahuje záznamy o cestách do různých sítí [10]. Podle vzniku záznamu ve směrovací tabulce hovoříme o statickém, nebo dynamickém směrování.



*Statické* - směrovací tabulka je plněna ručně administrátorem, který musí přesně určit, co a kudy se bude směřovat. Nevýhodou je nulová dynamika systému a nefunkčnost v případě poruchy v síti, nebo jakékoliv změny. Výhodou nulový provoz v síti v souvislosti s výměnou směrovacích informací. Běžně se používá pouze v malých sítích a konfiguraci koncového směrovače. [6]

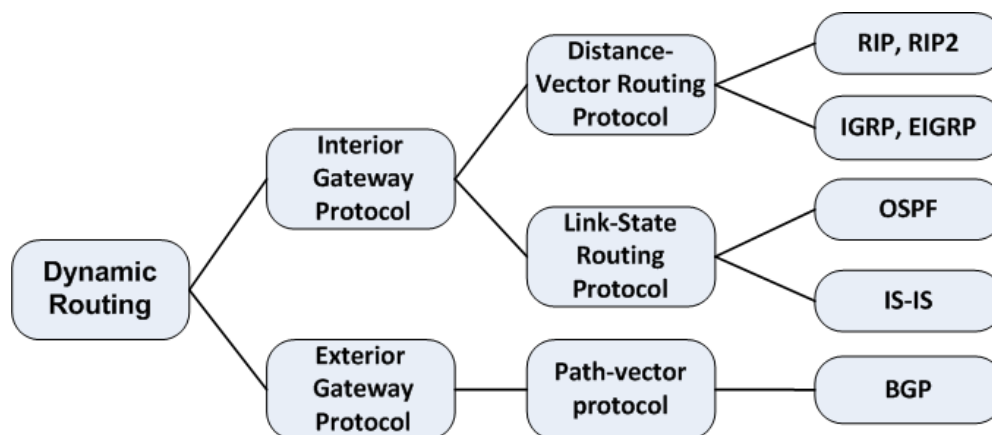
*Dynamické* - směrovací tabulka je naplněna automaticky, směrovače si dynamicky vyměňují směrovací informace na základě směrovacích protokolů.

### 2.2.1. Dynamické směrování

Dynamické směrování má za úkol nalézt pro paket procházející sítí nejlepší cestu do sítě cílové. Každý směrovací protokol má dvě komponenty. *Algoritmus*, což jsou definované kroky, kterými protokol postupuje při plnění obsahu směrovací tabulky, kdy vybere nejlepší z cest, a *zprávy*, údajové struktury, kterými sousední směrovače komunikují, aby si navzájem oznámily informace o dostupných sítích a cestám k nim.

Směrovací protokoly je možné definovat podle různých kritérií. Podle prostředí, ve kterém pracují se dělí do dvou skupin:

- *vnitřní směrovací protokoly* - *Interior Gateway Protocols (IGP)*, které se používají uvnitř autonomních systémů.
- *vnější směrovací protokoly* - *Exterior Gateway Protocols (EGP)*, které se používají na směrování mezi autonomními systémy.



Obrázek 1. Dělení dynamických protokolů

Podle principu činnosti se vnitřní směrovací protokoly dále dělí na:

- *Distance-Vector Routing Protocols* - hledají nejlepší cestu do vzdálené sítě na základě vzdálenosti. Trasa do sítě, která obsahuje nejnižší počet přeskoků, je považována za optimální. Vektor udává směr do vzdálené sítě. Mezi protokoly s vektorovou vzdáleností patří protokoly RIP, IGRP, EIGRP. Upraveným typem protokolu s vektorovou vzdáleností je *path-vector protocol*, do kterého patří BGP protokol, o kterém pojednává kapitola 3. [11]
- *Link-State Routing Protocols* - také se označují jako protokoly algoritmu nejkratší cesty (Shortest-Path-First Protocol), tzv. Dijkstraův algoritmus. Každý směrovač vytváří tři tabulky. Jedna s těchto tabulek sleduje přímo připojené sousedy, druhá určuje topologii celé datové sítě a třetí slouží jako směrovací tabulka. Směrovače se stavem linky mají o datové síti více informací než směrovače, které pracují se směrovacím protokolem o vektorové vzdálenosti. Mezi protokoly se stavem linky patří protokoly OSPF a IS-IS. [11]

Na obrázku č. 1. můžeme vidět dělení dynamických protokolů podle principu činnosti.

## 3. Směrovací protokol BGP(Border Gateway Protocol)

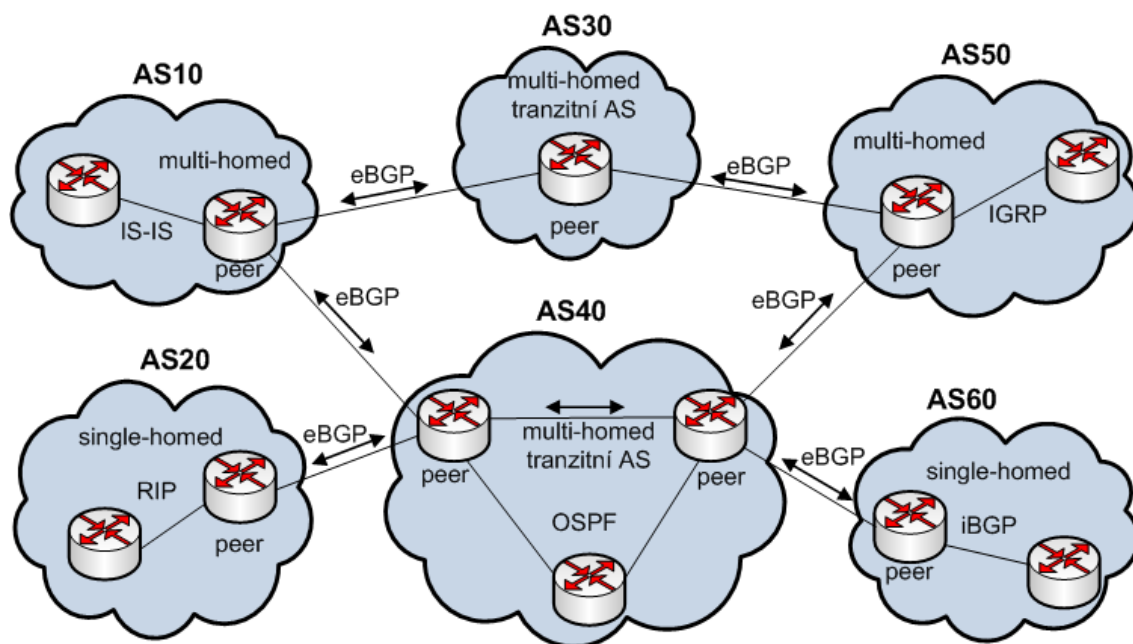
### 3.1. Autonomní systémy

Současný internet je natolik rozsáhlý a proměnlivý, že není reálné udržovat ve směrovačích úplnou informaci o jeho topologii. Proto bylo rozhodnuto směrování v Internetu řešit hierarchickým způsobem a byl rozdělen do tzv. autonomních systémů (AS). Autonomním systémem rozumíme souvislou skupinu sítí a směrovačů, které jsou pod společnou správou a řídí se společnou směrovací politikou. Při směrování v rámci jednotlivých AS se používají vnější směrovací protokoly (EGP) [7].

AS mohou být z topologického hlediska rozděleny do dvou kategorií, *single-homed* a *multi-homed*. Single-homed AS je připojen jedinou linkou k jinému AS, zatím co multi-homed je připojen více linkami. AS může být také *tranzitní* AS, takový systém dovoluje průchod provozu, který v něm nezačíná, ani nekončí [7]. Na obrázku č. 2. můžeme vidět příklad směrování v autonomních systémech.

### 3.2. Protokol BGP

BGP je dynamický směrovací protokol typu EGP (Exterior Gateway Protocol) [6] [7] [8], využívaný ke směrování v Internetu mezi autonomními



Obrázek 2. Směrování v autonomních systémech

systemy (AS). BGP je založen na výměně informací mezi BGP sousedy tzv. *peer* směrovače, spojení mezi těmito směrovači je nakonfigurováno ručně a pro přenos se využívá spolehlivý protokol TCP. Tento směrovací protokol patří mezi tzv. Path-vector protokoly, kde obsah směrovacích informací uchovává cenu cesty a zároveň i všechny předešlé skoky jako seznam AS, jimiž cesta prošla (k detekci smyček). BGP podporuje přenos směrovacích informací o sítích s beztrždní maskou.

AS svým sousedním AS prostřednictvím protokolu BGP sděluje k jakým IP sítím je schopen doručit IP pakety. Okolní AS se, podle své nastavené směrovací politiky, rozhodnou, jestli použijí konkrétní AS pro směrování k dané síti.

BGP může být využito i ke směrování v rámci AS, takovýto BGP se nazývá vnitřní (internal) - iBGP. V případě multi-homingu u AS, je nutné, aby peer směrovače byly navzájem propojeny iBGP. Potom lze uplatnit směrovací politiku na určení výstupu z AS. V tomto případě je nutnou podmínkou, aby peer směrovače byly propojeny každý s každým.

### 3.3. Zprávy protokolu BGP

Existují 4 typy zpráv, které si sousedé mohou mezi sebou vyměňovat. Zpráva protokolu BGP může mít maximální délku až 4096 B.

Typy zpráv:

- OPEN - zpráva vyměňována při zřizování vazby mezi sousedními směrovači.

Přenáší se v ní informace o AS, identifikace směrovače, apod. Musí být potvrzena zprávou KEEPALIVE.

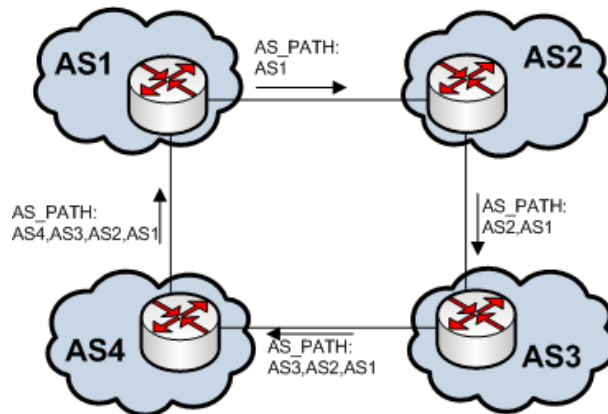
- UPDATE - zpráva nesoucí aktualizaci směrovacích informací. Aktualizace může obsahovat informace o zrušení, nebo o vzniku nových cest.
- KEEPALIVE - zpráva posílána pravidelně pro ověřování funkčnosti spojení mezi sousedy. Spojení se považuje za nefunkční, pokud od souseda nepřišla zpráva KEEPALIVE. Ve výchozí konfiguraci se vysílá každých 60 sekund.
- NOTIFICATION - zpráva používaná pro ukončení vazby mezi sousedy rozpojením TCP spojení. Je zasílána v případě chyby, při absenci zprávy KEEPALIVE, nebo jiné mimořádné situace.

### 3.4. Atributy cesty

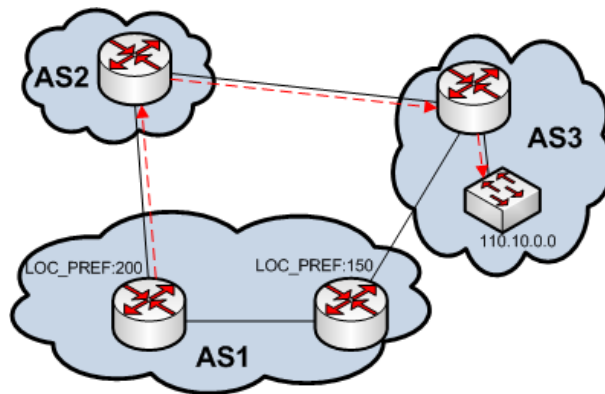
Vyhledávání pomocí algoritmu path-vector umožní nalézt nejkratší cesty do všech AS [7]. Abychom byli schopni explicitně ovlivňovat směrovací politiky, je potřebný mechanismus, kterým bychom vyjádřili preferenci, resp. zakázali některé cesty na základě různých kritérií. K tomuto účelu v BGP protokolu slouží tzv. atributy, které můžeme každému záznamu o cestě k cílové síti přiřadit. Atributy jsou přenášeny ve zprávě UPDATE. Je definováno několik atributů, z nichž některé jsou povinné a některé nepovinné.

Vybrané typy atributů [6]:

- ORIGIN - říká od kterého původce směrovací informace pochází. Přenášenou hodnotou jsou číselné kódy reprezentující buď IGP (Interior Gateway Protocol), EGP protokol, nebo INCOMPLETE (původ cesty není znám). Hodnota by neměla být měněna žádným jiným směrovačem.
- AS\_PATH - cesta k cílové síti, seznam AS kterými zpráva posla, jak můžeme vidět a obrázku č. 3. V případě, že se tento atribut zasílá v rámci autonomního systému, zůstává nezměněn. Pokud se cesta dostane do AS, jehož číslo je již v seznamu uvedeno, cesta se ignoruje, tímto se zabrání vzniku smyček.
- NEXT\_HOP - unicastová IP adresa dalšího skoku, který by měl být použit při směrování k cílové síti. Hodnota se při přenosu zprávy zpravidla mění, tak jak prochází mezi AS.
- LOCAL\_PREF - volitelný atribut, který se používá na nastavení směrovací politiky uvnitř AS. Je tedy využíván pouze v iBGP. Na základě tohoto atributu se rozhoduje který peer směrovač bude preferován, pokud do cílového AS vede více cest. Bude vybrána ta cesta, která má nastavenou větší hodnotu LOCAL\_PREF. Příklad můžeme vidět na obrázku č. 4.

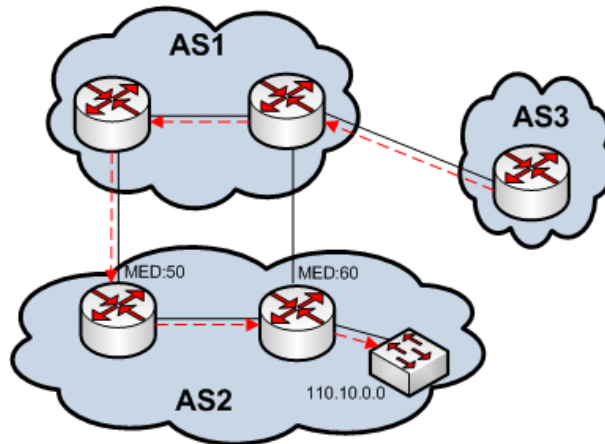


Obrázek 3. Použití atributu AS\_PATH



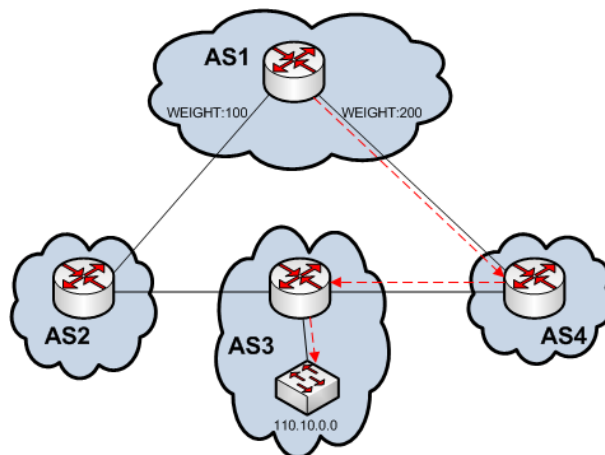
Obrázek 4. Ovlivnění výběru cest pomocí atributu LOC\_PREF

- MULTI\_EXIT\_DISC - volitelný atribut, který se používá k ovlivnění výběru cest směrem do spravované sítě. MED atributy se vyměňují pouze mezi dvěma sousedními AS a nesmí procházet do dalších AS za nimi. Pro směrování do AS je preferována cesta s nastaveným nižším atributem MED. Použití atributu MED můžeme vidět na obrázku č. 5.



Obrázek 5. Ovlivnění výběru cest pomocí atributu MED

- WEIGHT - jedná se o atribut definovaný firmou CISCO, který používá hodnotu váhy k určení nejlepší cesty (vyšší hodnota má vyšší prioritu), definuje se pouze pro lokální směrovač. Příklad použití atributu WEIGHT můžeme vidět na obrázku č. 6.



Obrázek 6. Ovlivnění výběru cest pomocí atributu WEIGHT

### 3.5. Směrovací proces protokolu BGP

U IGP protokolů rozhodují o procesu směrování údaje jako rychlost linky [6], vzdálenost, počet skoků apod. Při směrování mezi AS vystupují i jiné požadavky. Směrovací tabulky obsahují stovky tisíc záznamů nejdůležitějším kritériem nebývá jenom vzdálenost, ale posuzují se nastavitelné parametry například jako cena přenosu a dodatečná pravidla aplikovaná v závislosti na zdroji, cíli seznamu tranzitních autonomních systémů a dalších attributech. Rozhodnutí o politice

směrování je často závislé na administrátorovi.

Směrovací informace si uchovává každý BGP směrovač v tabulkách, nazývajících RIB (Routing Information Base) [6] [8]. Tyto tabulky jsou tří druhů:

- Adj-RIBs-In (Adjacent Routing Information Base - Incoming) obsahuje směrovací informace od svého souseda přijaté v příchozích zprávách UPDATE. Informace obsaženy v této tabulce jsou následně využity v rozhodovacím procesu směrovače.
- Loc-RIB (Local Routing Information Base) obsahuje informace o cestách, které byly vybrány z tabulky Adj-RIBs-In na základě směrovacích politik. Tyto informace jsou lokálně využívány BGP směrovačem.
- Adj-RIBs-Out (Adjacent Routing Information Base - Outgoing) tabulka, kde se nachází průběžně připravované informace, které byly vybrány k rozslání k sousedním BGP směrovačům zprávou typu UPDATE.

Pokud má směrovač k dispozici k cílové síti více cest, musí vybrat jednu do směrovací tabulky. Rozhoduje se podle jednotlivých kritérií v následujícím pořadí:

1. atribut WEIGHT - vyšší hodnota
2. atribut LOCAL\_PREF - vyšší hodnota
3. priorita lokálně vytvořených, nebo přes IGP redistribuovaných sítí
4. atribut AS\_PATH - nejkratší cesta
5. atribut ORIGIN - nejpreferovanější IGP, následuje EGP a poté INCOMPLETE
6. atribut MED - nižší hodnota
7. výběr cesty je preferován přes eBGP před iBGP
8. NEXT\_HOP dostupný kratší cestou přes IGP
9. na základě nižšího ROUTER\_ID, což je identifikátor směrovače. Typicky jako ROUTER\_ID se bere nejvyšší hodnota IP adresy nakonfigurované na rozhraní směrovače

### **Proces aktualizace směrovací tabulky**

Po přijetí UPDATE zprávy se tato zpráva zpracovává následujícím způsobem. Pokud se jedná o zprávu distribuující cesty, tak se v první řadě ověří, zda-li cesta neobsahuje smyčku (atribut AS\_PATH), pokud ano, zpracovávání této cesty se ukončí. Pokud cesta smyčku neobsahuje, je uložena do tabulky Adj-RIB-in pro daného souseda. Pokud již tabulka Adj-RIB-in obsahuje jinou cestu do stejné

destinace (z předchozích zpráv UPDATE), tato cesta se přepíše cestou novou. Pokud přijatá UPDATE zpráva obsahuje oznámení o nedostupnosti destinace (Withdrawn routes), daná cesta se z tabulky Adj-RIB-in odstraní.

### **Výběr nejlepší cesty**

V následujících třech fázích, se provede proces výběru optimálních cest do směrovací tabulky Loc-RIB:

#### 1. Ohodnocení cest podle preferencí

Na počátku této fáze se uzamknou všechny tabulky Adj-RIB-In. Pro každou novou cestu spočítá lokální BGP uzel ohodnocení preferovanosti, která se spočítá na základě směrovacích politik a délky cesty pomocí ohodnocovací funkce. Na závěr této fáze se otevřou tabulky Adj-RIB-in pro zápis.

#### 2. Výběr cest

V této fázi algoritmus pro výběr cest pracuje s výsledky ohodnocení preferovanosti cest, které byly získány v první fázi. Začne se upravovat směrovací tabulka Loc-RIB podle tabulek Loc-RIB-In. Cesty, které byly zrušeny pomocí zprávy UPDATE Withdraw a nacházejí se v tabulce Loc-RIB, budou s ní smazány.

Pro každou destinaci nalezne nejvíce preferovanou cestu podle daných pravidel:

- existuje li k jedné destinaci více cest, bude vybrána cesta s nejvyšším ohodnocením.
- existuje li k jedné destinaci více cest se stejným ohodnocením, zkoumají se u nich atributy cesty. První se vybere cesta s menším výstupním diskriminátorem (MED), pokud existuje stále více cest, vybere ta s menší místní preferencí (LOCEL\_PREF), pokud stále existuje více cest, vybere se cesta s nejmenším identifikátorem směrovače (ROUTER\_ID).
- existuje li k destinaci jediná cesta, vybere se ta.

Vybrané cesty se vloží do směrovací tabulky Loc-RIB. Pokud již cesta do dané destinace existuje, tato cesta je nahrazena cestou novou.

#### 3. Šíření změněných cest

Pokud byl ve druhé fázi obsah tabulky Loc-RIB změněn, mohou být tyto změny distribuovány okolním susedům. Nově přidané destinace a cesty k nim, nebo nové cesty ke stávajícím destinacím se uloží do tabulek Adj-RIB-Out. Obsah těchto tabulek se rozešle okolním susedům ve zprávách UPDATE. Součástí těchto zpráv je i oznámení o nedostupnosti k destinacím, které byly uvedeny v předchozí Loc-RIB. [8]



## 4. Simulace počítačových sítí

### 4.1. Modelování

Modelování je zjednodušená reprezentace skutečného systému. Modelování je důležité v návrhu systému a vývoje, protože dává představu o tom, co by systém provedl, pokud by byl skutečně realizován.

Systémové parametry v modelování mohou být měněny, zkoušeny a analyzovány. Správné použití a zpracování modelování může ušetřit náklady rozvoje reálného systému.

Existují dva přístupy modelování: analytický a simulační přístup. [2]

#### 4.1.1. Analytický přístup

Obecné pojetí analytického modelování spočívá v nalezení způsobu jak popsat systém matematicky pomocí použití matematických nástrojů, jako je například teorie pravděpodobnosti, a poté aplikovat numerické metody k získání přehledu z rozvinutého matematického modelu. [2]

#### 4.1.2. Simulační přístup

Simulace je široce používáno v systému modelování pro aplikace například v strojírenském výzkumu, obchodní analýze, plánování výroby, nebo experimentální biologii. Ve srovnání s analytickým modelováním simulace obvykle vyžaduje v modelu méně abstrakce, v simulačním modelu může být uveden téměř každý možný detail systému a tak co nejlépe popsat skutečný systém. [2]

## 4.2. Simulace

Simulace je proces navrhování modelu reálného systému a provádění experimentů s tímto modelem za účelem pochopení chování systému a nebo vyhodnocování strategie pro provoz systému. [2]

### 4.2.1. Prvky simulace

Simulace se skládají z následujících kroků:

#### *Subjekty*

Subjekty jsou objekty, které se vzájemně ovlivňují a způsobují v simulačním programu některé změny stavu systému.

V rámci počítačové sítě mohou subjekty zahrnovat počítačové uzly, pakety, toky paketů, nebo nefyzické předměty jako jsou simulační hodiny.

#### *Prostředky*

Prostředky jsou součástí komplexních systémů. Obecně platí, že omezený zdroj prostředků musí být sdílen určitou množinou subjektů. To je obvyklé pro případ počítačové sítě, kde například šířka pásma, vysílací čas, počet serverů představují síťové prostředky, které mají být sdíleny mezi subjekty sítě.

#### *Aktivita a události*

Čas od času, se subjekty zapojí do některých aktivit. Toto zapojení vytváří události a spouští změny v systému. Mezi běžné příklady aktivit patří zpoždění, vytváření front.

#### *Plánovač*

Plánovač udržuje seznam událostí a čas jejich provedení. Během simulace spustí simulační hodiny, vytváří události a provádí je.

#### *Globální proměnné*

V simulaci jsou globální proměnné dostupné z jakékoliv funkce nebo subjektu v systému a v podstatě udržují některé společné hodnoty simulace. V rámci počítačové sítě, by mohly takové proměnné představovat například délku fronty paketů, celkové obsazení vysílacího času v bezdrátové síti, nebo celkový počet paketů.

#### *Generátor náhodných čísel*

Je nutné zavést náhodnost do simulačního modelu.

V simulaci počítačové sítě, například proces doručení paketu, proces čekání a proces služby jsou obvykle modelovány jako náhodné procesy.

#### *Statistický sběrač*

Hlavní odpovědností statistického sběrače je sběr dat generovaných během simulace tak, aby bylo z těchto údajů možno vyvodit smysluplné závěry. [2]

### **4.2.2. Časově závislá simulace**

Hlavním typem simulace je simulace závislá na čase, která probíhá chronologicky. Tento typ simulace si udržuje simulační hodiny, které udržují přehled o aktuálním čase simulace. Ve většině případů simulace běží, dokud hodiny nedosáhnou předdefinované hodnoty.

Časově závislá simulace může být dále rozdělena *časově řízenou simulaci* a *událostně řízenou simulaci*.

Časově řízená simulace vyvolává a provádí události v daném časovém intervalu. Na druhé straně, událostně řízený simulátor vyvolává události v libovolném okamžiku. Simulace se přesunuje od jedné události k druhé a opět spustí událost (pokud existuje), dokud simulace neskončí. [2]

## 5. Network Simulator 2 (NS-2)

NS-2 je open-source událostně řízený simulátor, který se používá především v oblasti výzkumu. NS-2 poskytuje značnou podporu pro simulaci TPC, UDP, směrovacích unicastových i multicastových protokolů v drátových i bezdrátových (lokálních a satelitních) sítích. NS byl vyvíjen jako varianta REAL simulátoru vyvíjený na Kalifornské univerzitě v Berkeley od roku 1989. V roce 1995 byl vývoj podporován agenturou DARPA (Agentura pro výzkum pokročilých obranných projektů - Ministerstvo obrany USA). V současné době je NS podporován DARPA a NSF (Národní vědecká nadace - USA). [1]

NS-2 je napsaný ve dvou jazycích, C++ a OTcl. C++ definuje vnitřní mechanismus, výhodou tohoto jazyka je rychlost při běhu programu, a nevýhodou je pomalost při vyhledávání chyb, změnách v kódu a rekompilaci, z tohoto důvodu bylo zavedení druhého programovacího jazyka.

Jazyk OTcl je objektově orientovaný jazyk, který vyšel z jazyka Tcl. Je to jednoduchý skriptovací jazyk, který se používá pro nastavení simulací, konfiguraci objektů a plánování diskretních událostí. Výsledky simulace lze graficky interpretovat v nástrojích jako je NAM (Network Animator) nebo XGraph. [2]

### 5.1. Protokoly a modely podporované NS-2

V NS-2 je implementována celá řada protokolů a zahrnuje komunikaci v drátových, bezdrátových a satelitních sítích. [1]

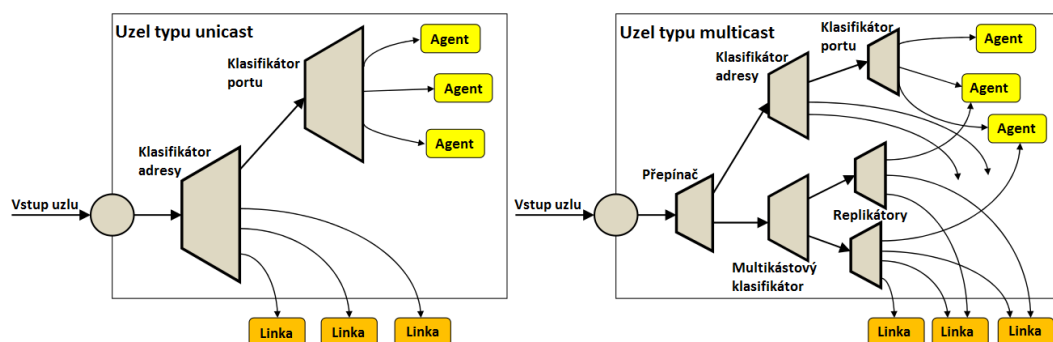
- Aplikační vrstva - telnet, HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), a další.
- Transportní vrstva - TCP, UDP, RTP, SRM
- Síťová vrstva - IPv4, ICMP, STP, ARP
- Práce s frontami - FQ (Fair Queueing), SFQ (Stochastic Fair Queueing), DRR (Deficit Round Robin), FIFO (First In First Out), RED (Random Early Discard), CBQ (Class Based Queueing)
- Linková vrstva - CSMA/CD, MAC
- Směrování - RIP, OSPF, BGP, atd.

### 5.2. Základní objekty modelu

Základní objekty, které NS-2 obsahuje [4] jsou uzly, linky, agenti a aplikace. Topologii sítě definují uzly a linky. Agenti reprezentují koncové body a vytvářejí, nebo zpracovávají pakety.

### 5.2.1. Uzel (Node)

Společně s linkou tvoří hlavní kostru v topologii simulované sítě. Uzel je složený objekt, který se skládá z dvou hlavních částí. Přístupu uzlu a klasifikátoru (adresový a portový). Klasifikátor adresy určuje, zda-li příchozí paket je adresován danému uzlu či nikoli, klasifikátor portu určuje na základě čísla portu, které aplikaci či agentu paket náleží. [4]

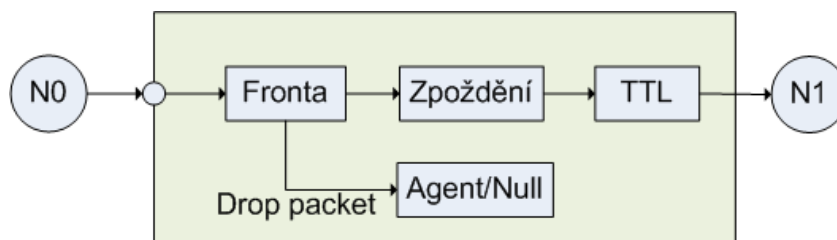


Obrázek 7. Unicastový a multicastový uzel

Struktura uzlu může být definována jako typu unicast, nebo typu multicast, které můžeme vidět na obrázku č.7. Multicastový uzel obsahuje navíc klasifikátor multicastové adresy.

### 5.2.2. Linka

Linky tvoří spojení mezi uzly. Podle druhu spojení můžeme nadefinovat typ linky s příslušnými parametry. NS primárně podporuje simplexní spojení. [4] Strukturu stavby simplexní linky zobrazuje obrázek č. 8.



Obrázek 8. Simplexní linka v NS

Pakety vstupují do fronty (Queue) na jejím výstupu je na ně aplikována simulace zpoždění (Delay) a ztráta paketu (Agent/null). Blok TTL (Time To Live)

zjistí a aktualizuje parametr TTL v hlavičce průchozího paketu.

Pro linku je i možné nastavit velikost vstupní paměti, kde v případě přetečení se s pakety zachází podle zvoleného typu fronty (např. DropTail - zahazování paketů).

### 5.2.3. Agenti

Představují koncové body, kde se vytváří, nebo zpracovává paket síťové vrstvy a používají se při implementaci protokolů vyšších vrstev. Agenti jsou vždy definováni mezi dvěma uzly a ke každému zdrojovému agentu odpovídá určitý druh cílového. [4]

### 5.2.4. Aplikace

Představují datové simulační zdroje a jsou připojeny nad transportní vrstvou, kde se nachází agenti reprezentující transportní protokoly (TCP, UDP, SCTP,...)

## 5.3. Error model

ErrorModel [2] je modul, který ukládá chyby do přenosu paketů. Může být vložen mezi dva NS objekty. Simuluje chybu paketu po přijetí paketu. Pokud je simulováno, že paket má chybu, error model paket zahodí, nebo označí s příznakem chyby. Pokud na paketu není simulována chyba, error model předá paket navazujícímu objektu. Error model může být použit jak na drátových, tak na bezdrátových sítích.

## 5.4. BGP++

BGP++ je implementace do NS-2, která umožňuje použití protokolu BGP při simulaci. BGP++(aktuální verze BGP++ 1.05beta) [13] vychází ze směrovacího software GNU Zebra. BGP++ integruje tzv. Zebra démona přímo do NS-2. Démon je založen na objektově orientovaném prostředí, upraveném pro komunikaci s rozhraním TCP a plánovačem simulátoru. Předností BGP++ je shoda syntaxe s konfigurací CISCO směrovačů.

## 6. Network Simulator 3 (NS-3)

NS-3 je událostně řízený síťový simulátor, zaměřený především na výzkum a vzdělávání, který může být použit k simulaci různých typů sítí. Projekt NS-3 byl zahájen v roce 2006, původně jako náhrada za NS-2 s počátečním důrazem na IEEE 802.11 WiFi modely.

Jádro NS-3 obsahuje sadu knihoven, které nabízí široký rámec nástrojů pro vývoj síťových simulací. Softwarová struktura NS-3 dovoluje simulovat modely v

realistických podmínkách. NS-3 simulační jádro podporuje výzkum jak na obou IP sítích, tak i v sítích bez IP protokolu. Obsahuje mnoho protokolů, které hlavně podporují WiFi, WiMAX, nebo sítě LTE, také obsahuje modely směrovacích protokolů jako je OLSR, nebo AODV. Nemá ale všechny modely, které obsahuje NS-2.

NS-3 je v současné době aktivně vyvíjen, každé 3 měsíce je vydána nová stabilní verze obsahující nové modely, které jsou zdokumentované a ověřené. Momentálně nejnovější verze je NS-3.21. [3]

## 6.1. Rozdíly mezi NS-2 a NS-3

NS-3 není rozšíření NS-2, je to nový simulátor, který není zpětně kompatibilní s NS-2, je vyvíjen také proto, aby NS-2 nahradil. Část NS-2 je psaná v C++ a část v OTcl, NS-3 je napsaný celý v C++. NS-2 simulační skripty jsou psány v skriptovacím jazyce OTcl a simulační skripty u NS-3 mohou být psány v C++, nebo v Pythonu, což má za následek jednoduššího psaní simulací, tím pádem i menší časovou náročnost. NS-3 se také pokouší vyřešit problémy vyskytující se v NS-2. Mezi velké výhody NS-3 patří emulační mód, který umožňuje integraci s reálnými sítěmi. [3]

## 6.2. Protokoly podporované v NS-3

V jádru NS3 je implementována celá řada protokolů, která jsou nezbytná ke správné komunikaci v síti. Mezi tyto protokoly patří:

- Transportní vrstva - UDP, TCP
- Síťová vrstva - IPv4, IPv6, globální statické směrování (unicast, multicast), OLSR, AODV, DSDV
- Linková vrstva - PPP, CSMA, 802.11 MAC
- Fyzická vrstva - 802.11a, základní nastavení kabelových linek (ztráta, zpoždění, kapacita linky)

Díky možnosti přímého vykonávání kódu, viz kapitola 6.6., lze tuto podporu rozšířit o další protokoly, jako například RIPng, OSPFv3, BGP-4. Jelikož je NS-3 stále vyvíjen a zdokonalován, počítá se s větší podporou protokolů, jako je tomu u NS-2.

## 6.3. Diskrétní simulace u NS-3

Podle teorie simulací je diskrétní simulace časově závislá, která se skládá z daných dílců, mezi které například patří:

- entity - zde patří třídy `Node`, `Packed` atd.
- prostředky - třída `Cannal`, nastavení vysílacího času
- aktivity a události - nastavení zpoždění ve třídě `Cannal`, výpadek linky pomocí `Topology Helper` IPv4
- plánovač - možnost nastavení času vysílání paketů, přerušování vysílání
- generátor náhodných čísel - aplikace, která generuje a přijímá pakety
- statický sběrač je možné použít několik typů sběračů, např. `AsciiTraceHelper`, nebo `PcapHelper` viz kapitola 6.5.

## 6.4. Základní objekty modelu

Základní objekty, které NS-3 obsahuje [3] jsou uzly, kanály, síťové rozhraní a aplikace. Topologii sítě definují uzly a linky. Aplikace vytvářejí, nebo zpracovávají pakety.

### 6.4.1. Uzel

V NS-3 je uzel základním abstraktním zařízením. Tato abstrakce je zastoupena v C++ třídou `Node`. Tato třída poskytuje metody pro řízení reprezentace výpočetních zařízení v simulaci.

Uzel si můžeme představit jako počítač, kterému přiřadíme tyto funkce:

- Aplikace
- Sady protokolů
- Síťové rozhraní

Strukturu stavby uzlu můžeme vidět na obrázku č. 9. [3]

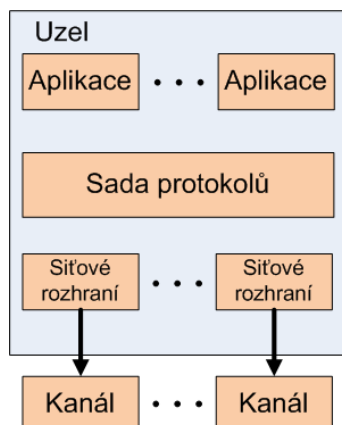
### 6.4.2. Aplikace

Aplikací v se NS-3 nazývá základní abstrakce pro uživatelský program, který generuje nějakou činnost, která má být simulována. Tato abstrakce je v C++ zastoupená třídou `Application`.

Každý uzel může využívat služeb jedné, nebo více aplikací. Mezi aplikace například patří:

- Generátor provozu
- Směrovací agenti
- Vypínání a zapínání simulace

[3]



Obrázek 9. NS-3 uzel

### 6.4.3. Kanál

Kanály spojují síťová rozhraní a umožňují uzlům mezi sebou komunikovat. Kanál může reprezentovat drátové spojení, či spojení přes optické vlákno. Specializovaný kanál může také modelovat složitější prostředí, jako je ethernetový přepínač, tří-dimenzionální prostor plný překážek v případě bezdrátové sítě.

V jazyku C++ je tato abstrakce zastoupena třídou `Channel`. [3]

### 6.4.4. Síťové rozhraní

V případě reálných sítí, pro připojení zařízení je nutné mít v zařízení síťovou kartu (NIC - Network Interface Card). Pro správné fungování této karty je potřeba také softwarové podpory v podobě ovladače. V NS-3 abstrakce síťového rozhraní představuje jak softwarový ovladač, ale také i hardwarovou síťovou kartu. Abstrakce síťového rozhraní je v jazyku C++ zastoupena třídou `NetDevice`. Síťová rozhraní jsou instalována v uzlu, aby tento uzel mohl komunikovat s ostatními uzly v simulaci pomocí kanálů. Stejně jako v reálném počítači, může být uzel připojen k více než jednomu kanálu přes více síťových rozhraní. [3]

### 6.4.5. Topology Helpers

NS-3 obsahuje tzv. Topology Helpers pro zjednodušení konfigurace sítě. V reálné síti má každý síťový prvek MAC adresu, nastavenou IP adresu a nadefinovány parametry přenosu přes fyzické rozhraní. Pro zjednodušení vytváření rozsáhlé sítě v simulátoru NS-3, například při zadávání IP adres síťovým rozhraním, použijeme daný Topology Helper. Použitím Topology Helperu se můžeme vyhnout hůře odstranitelným chybám, jako je například špatné zadání IP adresy, kdy tomuto helperu (konkrétně u IPv4 adres `Ipv4AddressHelper`) určíme adresní rozsah zadáním IP adresy sítě, včetně masky. Topology Helper se postará o přidělení jednotlivých IP adres jednotlivým zařízením.



NS-3 obsahuje velkou sadu funkcí Topology Helper, které se starají o to, aby běžné úkony byli co nejjednodušší a nejrychlejší. [3]

## 6.5. Výstupní soubory a vizualizace

Pro sledování průběhu a sběru výsledku simulace obsahuje NS-3 několik funkcí, jako jsou například funkce AsciiTraceHelper, nebo PcapHelper [3]. Funkce AsciiTraceHelper slouží pro záznam událostí simulace ve formě textu, který se ukládá do souboru typu txt. Funkce PcapHelper slouží pro záznam událostí simulace do souboru typu pcap. Pro analýzu výstupu simulace u tohoto typu souboru lze použít program Wireshark.

**Wireshark** - je nástroj pro sledování sítě a síťový analyzátor, tj. nástroj, který nejenom zachycuje pakety v síti, ale navíc umožňuje jejich analýzu, která zahrnuje jak pitvání datových paketů do největších detailů, tak vytváření nejrůznějších statistik a grafů. [5]

Samotný simulátor NS-3 neobsahuje žádné grafické rozhraní, kterým by bylo možné zobrazit návrh simulovaného návrhu modelu, ale jsou dostupné různé grafické nástavby. Nejpoužívanější grafický program je NetAnim.

**NetAnim** - je offline animátor, zobrazuje animace simulace pomocí souboru XML, do kterého byly nashromážděny informace v průběhu simulace. Tento program umožňuje graficky zobrazit umístění uzlů, samotný průběh simulace v čase i přenos paketů mezi uzly.

## 6.6. Přímé vykonávání kódu

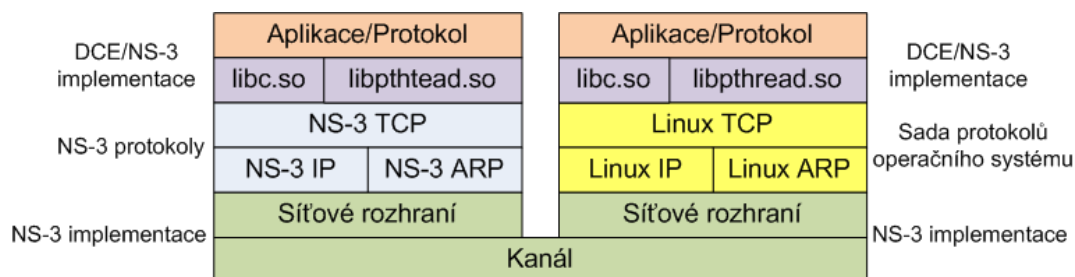
Direct Code Execution (DCE) je nástroj, který umožňuje při použití NS-3 využívat protokolové implementace operačního systému bez nutnosti zasahovat do zdrojového kódu. Podpora DCE ve spolupráci s NS-3 nabízí realističtější simulování sítě s přesnějšími výsledky, než starší simulátory.

DCE v NS-3 nabízí dva operační módy:

- Základní - používá sadu protokolů TCP/IP implementovanou v NS-3
- Pokročilý - používá sadu síťových protokolů implementovaných v jádru operačního systému (Linux)

Na obrázku č. 10. můžeme vidět vrstevný model DCE pro základní i pokročilý operační mód.

Pro DCE byla umožněna spolupráce s dalšími aplikacemi, které umožňují větší využití simulátoru NS-3. Mezi takové aplikace patří aplikace Quagga. [3]



Obrázek 10. Vrstvový model DCE pro základní a pokročilý operační mód v NS-3

### 6.6.1. Quagga

Quagga je směrovací softwarový balík, poskytující implementace směrovacích protokolů OSPFv2, OSPFv3, RIPv1, RIPv2, RIPng, BGP-4 a síťové protokoly IPv4 a IPv6 pro operační systémy Unix, Linux, Solaris.

Díky programu Quagga je možné v systému využívat démonů, zajišťující funkci směrování podobně jako směrovač. [3]

## 7. Praktická část

Tato část bakalářské práce se zaměřuje na ověření principu chování protokolu BGP. Budou navrženy dvě simulace. První simulace bude provedena na simulátoru NS-3 a zaměřuje se na ověření chování protokolu BGP při navazování spojení, předávání zpráv, činnost protokolu při výpadku linky během přenosu dat, na chování při přidání nové linky mezi autonomní systémy a chování při přidání nového autonomního systému do sítě. Druhá simulace bude provedena na simulátoru NS-2 a zaměřuje se na ověření ovlivnění výběru cest při směrování pomocí atributů WEIGHT, LOCAL\_PREF a MED protokolu BGP.

K dnešnímu datu simulátor NS-3 poskytuje pouze základní podporu protokolu BGP, do které nespadá možnost nastavení atributů WEIGHT, LOCAL\_PREF a MED. Z tohoto důvodu bude druhá úloha provedena pomocí simulátoru NS-2.

### 7.1. Ověření základního chování protokolu BGP

Tato úloha ověří chování protokolu BGP při navazování spojení, předávání zpráv, činnost protokolu při výpadku linky během přenosu dat, na chování při přidání nové linky mezi autonomní systémy a chování při přidání nového autonomního systému do sítě. Simulace bude provedena pomocí simulátoru NS-3. Součástí této části práce bude popis instalace simulátoru NS-3 a dodatečných balíčků DCE a Quagga, dále popis vytváření modulu simulace.

### 7.1.1. Instalace NS-3

NS3 je primárně vyvíjen pro platformy GNU/Linux. Pro použití NS3 v systému Windows je možné využít prostředí Cyrwin, které ale už není v dnešní době podporováno. Další alternativou je nainstalování virtualizačního nástroje, jako je například VirtualBox a nainstalování virtuálního operačního systému Linux.

Pro potřeby bakalářské práce je použita linuxová distribuce Ubuntu 13.10.

Před samotnou instalací simulátoru NS3 je potřeba nainstalovat požadované nástroje a balíčky:

- Mercurial - nástroj pro správu organizace a změn zdrojového kódu a dokumentace
- Waf - nástroj pro kompilaci a instalaci aplikací. Nahrazuje nástroj make, který není příliš vhodný pro použití velkých a vysoce konfigurovatelných systémů. Nástroj Waf je založený na jazyku Python.
- Vývojové prostředí - z důvodu použití jazyka C++ při psaní skriptů, je nutné nainstalovat kompilátor pro C++

Tyto základní balíčky získáme zadáním příkazu v terminálu:

```
sudo apt-get install gcc g++ python python-dev mercurial
```

Z důvodu podpory protokolu BGP programem Quagga, bude dále popsána instalace simulátoru NS3 včetně DCE-Quagga.

DCE s podporou Quagga vyžaduje několik balíčků: autoconf, automake, flex, git-core, wget, g++, libc-dbg, bison, indent, pkgconfig, libssl-dev, libsysfs-dev, gawk.

Při instalaci NS3 DCE Quagga, můžeme využít instalačního nástroje Bake, vyvinutého pro projekt NS3. Nejprve si stáhneme a nainstalujeme nástroj Bake pomocí příkazu hg programu Mercurial do složky bake.

```
hg clone http://code.nsnam.org/bake bake
```

Definujeme proměnné, pro možnost použití Bake mimo adresář, do kterého byl nainstalovaný.

```
export BAKE_HOME='pwd' /bake
export PATH=$PATH:$BAKE_HOME
export PYTHONPATH=$PYTHONPATH:$BAKE_HOME
```

V dalším kroku si nainstalujeme simulátor ns-3-dce-quagga do složky, kterou si vytvoříme. Po zadání `-e dce -linux-|verze|` v parametrech `bake.py configure`, bude nainstalována verze, která bude používat protokolové implementace linuxového systému. Pokud bychom chtěli používat protokoly ns3 simulátoru, jako parametry zadáme `-e dce -ns3-|verze|`.

```
mkdir dce
cd dce
bake.py configure -e dce- linux-|verze| -e dce-quagga-|verze|
bake.py download
bake.py build
```

Po úspěšné instalaci je vhodné provést kontrolu programu, zda byl správně sestaven. Kontrolu provedeme příkazem:

```
cd source/ns-3-dce ./test.py -s dce-quagga
```

[12]

### 7.1.2. Vytváření skriptu simulace

Na obrázku č. 11. můžeme vidět návrh sítě, kterou použijeme při simulaci. Skládá se z osmi uzlů, které budou simulovat peer směrovače jednotlivých autonomních systémů. Při simulaci výpadku linky během přenosu dat směrovači AS3 přidělíme roli klienta, který bude vysílat data v podobě UDP paketů směrovači AS6, kterému přidělíme roli serveru. Velikost přenášených dat v jednom paketu bude nastavena na 1024 bytů. Použitým směrovacím protokolem bude BGP. Délka trvání simulace bude 3000s, klient začne vysílat UDP pakety po 100 sekundách běhu simulace, délka vysílání bude 20s. Linky spojující uzly budou typu point-to-point, každé spojení uzlů se bude nacházet v samostatné síti. Všechny linky budou mít nastavenou stejnou kapacitu a to 5Mb/s a zpoždění na linkách bude nastaveno na 2ms.

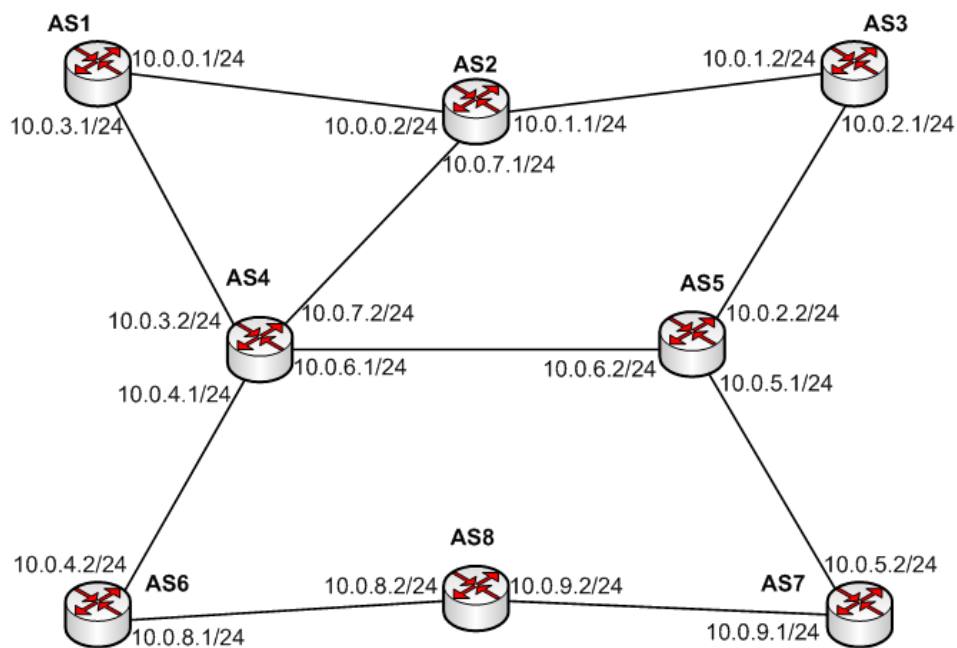
Simulaci budeme provádět, aby jsme zjistili a ověřili pomocí nastavených událostí, jaké je chování protokolu BGP při navazování spojení, odesílání aktualizací pro směrovací tabulky pomocí zpráv UPDATE (vytváření a rušení směrovacích informací). Jelikož nebudeme v simulaci nastavovat parametry jako, jsou váha (WEIGHT), nebo místní preference (LOCAL\_PREF), měla by při zahájení přenosu dat, při možnosti více cest, být zvolena cesta s nejnižším počtem procházených AS (AS\_PATH). Dále se zaměříme na chování směrovačů při výpadku linky v průběhu přenosu dat, na chování při přidání nové linky a při přidání nového směrovače.

Při simulaci přenosu dat bude role klienta vysílajícího data přidělena uzlu AS3 a role serveru přijímajícího tyto data přidělena uzlu AS6. Událost výpadku

linky bude nastavena na čas 105s. Výpadek linky bude simulován vypnutím portu uzlu AS5 s linkou k AS4 (10.0.6.2).

Linkou mezi směrovači AS1 a AS4 (10.0.7.0) bude simulováno vytvoření nového spojení. Událost přidání linky bude nastavena na čas 130s.

Směrovačem AS8 bude simulováno připojení nového autonomního systému do sítě. Událost přidání směrovače AS8 bude nastavena na čas 140s.



Obrázek 11. Návrh simulované sítě

Nyní přejdeme k samotnému návrhu simulace. Skript budeme psát v jazyku v jazyce C++. Pro vytvoření, editaci skriptu můžeme použít jeden s velkého množství textových editorů, nebo vývojových platforem, jako jsou například Eclipse, nebo Geany.

Na začátku programu připojíme externí moduly z knihovny NS-3. Není to nejefektivnější řešení, jelikož moduly mohou obsahovat i hlavičkové soubory, které v programu nepoužijeme, ale psaní scénáře tímto způsobem je mnohem jednodušší.

```
#include "ns3/netanim-module.h"
#include "ns3/network-module.h"
#include "ns3/core-module.h"
#include "ns3/internet-module.h"
#include "ns3/dce-module.h"
#include "ns3/quagga-helper.h"
#include "ns3/point-to-point-helper.h"
#include "ns3/applications-module.h"
```

Moduly zajišťují funkčnost a správné chování simulovaných sítí.

Dále definujeme obor názvů `namespace`. Potom následuje hlavní funkce programu, do které nadefinujeme celý scénář simulace.

```
using namespace ns3;
int main (int argc, char *argv[])
```

K definování uzlů použijeme helper `NodeContainer`, který nám slouží k jednoduchému vytvoření, zorganizování a přístupu k uzlům. Metodou `Create` vytvoříme požadované množství uzlů:

```
NodeContainer nodes;
nodes.Create (8);
```

Dále nadefinujeme linku a její parametry. K tomuto použijeme topology helper `PointToPointHelper`. Jelikož parametry linek mezi uzly budou stejné, bude stačit pouze jedna definice:

```
PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue("5Mbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue("2ms"));
```

Dalším krokem bude přiřazení linek mezi uzly. Pro přiřazení využijeme pomocí helperu `NetDeviceContainer` vytvořením kontejneru, který můžeme chápat jako síťové rozhraní na daných uzlech. Pomocí metody `Install` přiřadíme linky mezi dvojice uzlů:

```
NetDeviceContainer devices;
devices = pointToPoint.Install (nodes.Get (0), nodes.Get (1));
devices = pointToPoint.Install (nodes.Get (1), nodes.Get (2));
```

Jelikož budeme chtít při simulaci využívat sadu protokolů implementovaných přímo v linuxovém jádru, musí být tato sada nastavena. Pro toto nastavení využijeme pomocí helperu `DceManagerHelper`. Metodou `SetNetworkStack` určíme sdílenou knihovnu, pomocí které bude simulátor využívat protokolů jádra.

```
DceManagerHelper processManager;
processManager.SetTaskManagerAttribute ("FiberManagerType",
                                         EnumValue (0));
processManager.SetNetworkStack ("ns3::LinuxSocketFdFactory", "Library",
                                 StringValue ("liblinux.so"));
processManager.Install (nodes);
```

Nyní nastavíme IP adresy na jednotlivých uzlech, k čemuž využijeme funkce `AddAddress` a `RunIP`. Konfiguraci IP adres a aktivování rozhraní zajišťuje linuxový program `ip`, který má podobu:

```
ip -f inet addr add address dev name
```

K volání tohoto příkazu slouží funkce `RunIP` a `AddAddress`.

```
AddAddress (nodes.Get (0), Seconds (0.1), "sim0", "10.0.0.1/24");
RunIp (nodes.Get (0), Seconds (0.11), "link set lo up");
RunIp (nodes.Get (0), Seconds (0.11), "link set sim0 up");
```

Dalším krokem je nastavení směrovacího protokolu BGP pomocí helperu `QuaggaHelper`. Metodou `EnableHelper` přiřadíme daný směrovací protokol uzlům, na kterých tento protokol požadujeme. Pomocí metody `BgpAddNeighbor` nastavíme na jednotlivých uzlech směrovací informace o sousedních směrovačích.

```
QuaggaHelper quagga;
quagga.EnableBgp (nodes);
quagga.BgpAddNeighbor (nodes.Get (0), "10.0.0.2",
                       quagga.GetAsn (nodes.Get (1)));
quagga.BgpAddNeighbor (nodes.Get (0), "10.0.3.2",
                       quagga.GetAsn (nodes.Get (3)));
quagga.Install (nodes);
```

Dále vytvoříme aplikace pro generování paketů, kterou přiřadíme klientovi a aplikace pro přijímání paketů kterou přiřadíme serveru. Pro vytvoření těchto aplikací slouží topology helper `DceApplicationHelper`, který umožňuje spustit binární kód linuxového programu. Pro generování paketů a určení příjemce bude využíván program `udp-perf`. Pomocí metody `AddArgument` nastavíme délku trvání aplikací a čísla portů. U klienta, který bude generovat UDP datagramy, nastavíme množství přenášených dat v jednom datagramu a IP adresu cílového uzlu. Pomocí metody `Start` nastavíme čas spuštění aplikací.

```
DceApplicationHelper process;
ApplicationContainer apps;

std::ostringstream oss;
process.SetBinary ("udp-perf");
process.AddArgument ("--duration=100");
process.AddArgument (oss.str ().c_str ());
oss.clear ();
process.AddArgument ("--port=9");
process.AddArgument (oss.str ().c_str ());
apps = process.Install (nodes.Get(5));
apps.Start (Seconds (100.0));

process.SetBinary ("udp-perf");
```

```

process.ResetArguments ();
process.AddArgument ("--pktsize=500");
process.AddArgument (oss.str ().c_str ());
oss.clear ();
process.AddArgument ("--port=9");
process.AddArgument (oss.str ().c_str ());
oss.clear ();
process.AddArgument ("--client");
process.AddArgument (oss.str ().c_str ());
oss.clear ();
oss.str ("");
oss << "--host=" << Ipv4AddressToString ("10.0.4.2");
process.AddArgument (oss.str ().c_str ());
oss.clear ();
oss.str ("");
process.AddArgument ("--duration=20");
apps = process.Install (nodes.Get(2));
apps.Start (Seconds (100.0));

```

Pro simulaci výpadku linky využijeme možnosti funkce RunIP, kdy parametrem Down deaktivujeme rozhraní na uzlu. Čas výpadku nastavíme na uzlu č.4 (AS5), přes který bude datový tok směřován.

```
RunIp (nodes.Get (4), Seconds (110.0), "link set sim1 down");
```

Pro simulaci vytvoření nové linky a připojení nového AS využijeme také funkci RunIP, kdy parametrem UP aktivujeme rozhraní na uzlech.

```
RunIp (nodes.Get (1), Seconds (130.0), "link set sim2 up");
```

Posledním krokem v definování skriptu bude nastavení uložení výsledků simulace do výstupních souborů. Pro tuto činnost využijeme pomoci topology helperů AsciiTraceHelperForDevice, PcapHelperForDevice a třídu AnimationInterface modulu netanim-module.

```

pointToPoint.EnableAsciiAll ("vysledky-txt/quagga-bgpd");
pointToPoint.EnablePcapAll ("vysledky-pcap/quagga-bgpd");
AnimationInterface anim ("vysledky-xml/quagga-bgpd.xml");

```

První metoda ukládá výsledky simulace jako textový soubor, druhá metoda jako soubor typu pcap. Soubor typu pcap je možné otevřít v programu Wireshark. Poslední metoda ukládá výsledky simulace do souboru xml, který je spustitelný v programu NetAnim.



### 7.1.3. Spuštění simulace

Pro sestavení, kompilaci a spuštění se používá program Waf. Před spuštěním je nejdříve nutné skript uložit do složky example v adresáři myscripts/ns-3-quagga a poté upravit konfigurační soubor wscript uložený ve složce myscripts/ns-3-quagga. Do konfiguračního souboru je potřeba nadefinovat modely použité v simulaci, název souboru, cílový adresář a jméno zkompilevaného souboru. Simulace se spustí příkazem:

```
./waf --run quagga-bgpd
```

[3] [12]

### 7.1.4. Vyhodnocení simulace

#### Chování protokolu při výpadku linky během přenosu dat.

Pro vyhodnocení simulace použijeme výsledky simulace zaznamenané na peer směrovačích AS3 a AS5 (quagga-bgpd-2-0.pcap, quagga-bgpd-2-1.pcap, quagga-bgpd-4-0.pcap, routingtableevery2). U AS3 budeme vyhodnocovat provoz na obou linkách (AS5, AS2), z důvodu možnosti vidět přenos zpráv UPDATE a následnou úpravu směrovací tabulky na základě informací přenášených těmito zprávami. Konkrétně nás zajímají údaje o směrovacích informacích do sítě 10.0.4.0, kde se nachází server, do kterého je směrován přenos dat. U směrovače AS5 nás bude zajímat chování směrovače při výpadku linky.

Komunikace mezi směrovači začíná vysláním zprávy OPEN, která slouží pro zřízení vazby mezi směrovači. Jelikož protokol BGP používá ke komunikaci protokol TCP dochází mezi směrovači k navázání spojení a poté je vyslána zpráva OPEN. Aby byla relace mezi směrovači navázána, odpovídá druhý směrovač

No.	Time	Source	Destination	Protocol	Length	Info
6	8.000092	fe80::206ff02::2		ICMPv6	58	Router Solicitation from 00:00:00:00:00:06
7	9.489999	10.0.2.1	10.0.2.2	TCP	62	53630 > bgp [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK PERM=1 TSval=2399
8	9.494198	10.0.2.2	10.0.2.1	TCP	62	bgp > 53630 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK PERM=1
9	9.494198	10.0.2.1	10.0.2.2	TCP	54	53630 > bgp [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSval=2401 TSecr=2400
10	9.494284	10.0.2.1	10.0.2.2	BGP	107	OPEN Message
11	9.498371	10.0.2.2	10.0.2.1	TCP	54	bgp > 53630 [FIN, ACK] Seq=1 Ack=1 Win=5792 Len=0 TSval=2401 TSecr=2401
12	9.498371	10.0.2.1	10.0.2.2	TCP	54	53630 > bgp [FIN, ACK] Seq=54 Ack=2 Win=5840 Len=0 TSval=2402 TSecr=2400
13	9.498523	10.0.2.2	10.0.2.1	TCP	42	bgp > 53630 [RST] Seq=1 Win=0 Len=0
14	9.502524	10.0.2.2	10.0.2.1	TCP	42	bgp > 53630 [RST] Seq=2 Win=0 Len=0
15	13.092100	10.0.2.2	10.0.2.1	TCP	62	36908 > bgp [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK PERM=1 TSval=3300
16	13.092100	10.0.2.1	10.0.2.2	TCP	62	bgp > 36908 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK PERM=1
17	13.096286	10.0.2.2	10.0.2.1	TCP	54	36908 > bgp [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSval=3301 TSecr=3300
18	13.096457	10.0.2.2	10.0.2.1	BGP	107	OPEN Message
19	13.096457	10.0.2.1	10.0.2.2	TCP	54	bgp > 36908 [ACK] Seq=1 Ack=54 Win=5792 Len=0 TSval=3301 TSecr=3301
20	13.096544	10.0.2.1	10.0.2.2	BGP	126	OPEN Message, KEEPALIVE Message
21	13.100832	10.0.2.2	10.0.2.1	TCP	54	36908 > bgp [ACK] Seq=54 Ack=73 Win=5840 Len=0 TSval=3302 TSecr=3301
22	13.100979	10.0.2.2	10.0.2.1	BGP	92	KEEPALIVE Message, KEEPALIVE Message

Obrázek 12. Navazování spojení mezi peer směrovači AS3 a AS5

vysláním zprávy KEEPALIVE. Spojení mezi směrovači je navázáno. Tuto komunikaci můžeme vidět na obrázku č. 12., kdy dochází k navázání spojení mezi směrovači AS3 a AS5.

Dále dochází k distribuci zpráv UPDATE. Tato zpráva obsahuje směrovací informace o dostupnosti sítí. Distribuci zpráv UPDATE můžeme vidět na obrázcích č. 13. a č. 14. ze směrovačů AS2 a AS5 do směrovače AS3. V našem

No.	Time	Source	Destination	Protocol	Length	Info
28	19.100921	10.0.2.2	10.0.2.1	BGP	110	UPDATE Message
29	19.100979	10.0.2.1	10.0.2.2	BGP	170	UPDATE Message, UPDATE Message
30	19.105337	10.0.2.2	10.0.2.1	TCP	54	36908 > bgp [ACK] Seq=211 Ack=319 Win=5840 Len=0 TSval=4803 TSecr=4802
31	19.150921	10.0.2.1	10.0.2.2	BGP	81	UPDATE Message

```

▶Point-to-Point Protocol
▶Internet Protocol Version 4, Src: 10.0.2.2 (10.0.2.2), Dst: 10.0.2.1 (10.0.2.1)
▶Transmission Control Protocol, Src Port: 36908 (36908), Dst Port: bgp (179), Seq: 155, Ack: 203, Len: 56
▼Border Gateway Protocol - UPDATE Message
  ▼Path attributes
    ▶ORIGIN: INCOMPLETE (4 bytes)
    ▼AS_PATH: 5 4 (14 bytes)
      ▶Flags: 0x50 (Well-known, Transitive, Complete, Extended Length)
        Type code: AS_PATH (2)
        Length: 10 bytes
      ▼AS path: 5 4
        ▼AS path segment: 5 4
          Path segment type: AS_SEQUENCE (2)
          Path segment length: 2 ASs
          Path segment value: 5 4
        ▶NEXT_HOP: 10.0.2.2 (7 bytes)
    ▼Network layer reachability information: 8 bytes
      ▶10.0.3.0/24
      ▶10.0.4.0/24
  
```

Obrázek 13. Zpráva UPDATE vyslaná z AS5 do AS3

No.	Time	Source	Destination	Protocol	Length	Info
27	13.503158	10.0.1.2	10.0.1.1	TCP	54	41783 > bgp [ACK] Seq=151 Ack=151 Win=5840 Len=0 TSval=3403 TSecr=3402
28	18.498743	10.0.1.2	10.0.1.1	BGP	110	UPDATE Message
29	18.503243	10.0.1.1	10.0.1.2	BGP	166	UPDATE Message, UPDATE Message
30	18.503243	10.0.1.2	10.0.1.1	TCP	54	41783 > bgp [ACK] Seq=207 Ack=263 Win=5840 Len=0 TSval=4653 TSecr=4652

```

▶Internet Protocol Version 4, Src: 10.0.1.1 (10.0.1.1), Dst: 10.0.1.2 (10.0.1.2)
▶Transmission Control Protocol, Src Port: bgp (179), Dst Port: 41783 (41783), Seq: 151, Ack: 207, Len: 112
▶Border Gateway Protocol - UPDATE Message
▼Border Gateway Protocol - UPDATE Message
  ▼Path attributes
    ▶ORIGIN: INCOMPLETE (4 bytes)
    ▼AS_PATH: 2 1 4 (18 bytes)
      ▶Flags: 0x50 (Well-known, Transitive, Complete, Extended Length)
        Type code: AS_PATH (2)
        Length: 14 bytes
      ▼AS path: 2 1 4
        ▼AS path segment: 2 1 4
          Path segment type: AS_SEQUENCE (2)
          Path segment length: 3 ASs
          Path segment value: 2 1 4
        ▶NEXT_HOP: 10.0.1.1 (7 bytes)
    ▼Network layer reachability information: 8 bytes
      ▶10.0.6.0/24
      ▶10.0.4.0/24
  
```

Obrázek 14. Zpráva UPDATE vyslaná z AS2 do AS3

případě nebyly nastaveny atributy WEIGHT a LOCAL\_PREF, které mají při výběru cesty vyšší prioritu než atribut AS\_PATH, tím pádem bude rozhodováno při směrování do cílové sítě na základě vzdálenosti, kterou obsahuje atribut AS\_PATH. Konkrétně nás zajímá údaj o vzdálenosti do sítě 10.0.4.0, na základě kterého bude rozhodnuto o směrování přenosu dat z klienta do serveru. Dále

zpráva UPDATE obsahuje atribut NEXT\_HOP s IP adresou dalšího skoku, který by měl být použit při směrování k cílové síti. Dále zde můžeme vidět zaznamenané AS, kterými zpráva prošla (Path segment value). Po vyhodnocení přijatých zpráv UPDATE dochází k aktualizaci směrovací tabulky. Na obrázku č. 15. vidíme aktualizovanou tabulku směrovače AS3. Směrování do sítě 10.0.4.0, pro přenos dat, je nastaveno přes směrovač AS5 a to z důvodu menší vzdálenosti.

Time: 20s						
Destination	Gateway	Genmask	Flags	Metric	Ref	Use Iface
127.0.0.0	0.0.0.0	255.0.0.0	U	0	-	0
10.0.1.0	0.0.0.0	255.255.255.0	U	0	-	1
10.0.2.0	0.0.0.0	255.255.255.0	U	0	-	2
10.0.6.0	10.0.2.2	255.255.255.0	UGS	1	-	2
10.0.5.0	10.0.2.2	255.255.255.0	UGS	1	-	2
10.0.3.0	10.0.1.1	255.255.255.0	UGS	0	-	1
10.0.4.0	10.0.2.2	255.255.255.0	UGS	0	-	2
10.0.0.0	10.0.1.1	255.255.255.0	UGS	1	-	1

Obrázek 15. Směrovací tabulka směrovače AS3.

Po navázání spojení dochází mezi směrovači k opakovaným výměnám zpráv KEEPALIVE, což je zprávou, která je vyměňována mezi sousedy pro ověření funkčnosti linky. Výměnu zpráv KEEPALIVE můžeme vidět na obrázku č. 16. V našem případě směrovače mají nastavený časovač na vysílání těchto zpráv po každých šedesáti sekundách, což je nastavení výchozí.

Podle nastavených událostí pro simulaci je v čase 100s (99,8986s) zahájeno vysílání UDP paketů ze směrovače AS2 do směrovače AS6. Na základě údajů ve směrovací tabulce směrovače AS2 dochází k odesílání paketů přes směrovač AS5. Zahájení vysílání můžeme vidět na obrázku č. 16.

No.	Time	Source	Destination	Protocol	Length	Info
32	19.155137	10.0.2.2	10.0.2.1	TCP	54	36908 > bgp [ACK] Seq=211 Ack=346 Win=5840 Len=0 TSval=4815 TSecr=4815
33	73.096457	10.0.2.1	10.0.2.2	BGP	73	KEEPALIVE Message
34	73.100660	10.0.2.2	10.0.2.1	TCP	54	36908 > bgp [ACK] Seq=211 Ack=365 Win=5840 Len=0 TSval=18302 TSecr=18301
35	73.100862	10.0.2.2	10.0.2.1	BGP	73	KEEPALIVE Message
36	73.138000	10.0.2.1	10.0.2.2	TCP	54	bgp > 36908 [ACK] Seq=365 Ack=230 Win=5792 Len=0 TSval=18312 TSecr=18302
37	99.898695	10.0.2.1	10.0.4.2	UDP	1054	Source port: 40104 Destination port: discard
38	99.907391	10.0.2.1	10.0.4.2	UDP	1054	Source port: 40104 Destination port: discard
39	99.916086	10.0.2.1	10.0.4.2	UDP	1054	Source port: 40104 Destination port: discard
40	99.924782	10.0.2.1	10.0.4.2	UDP	1054	Source port: 40104 Destination port: discard

▶ Frame 37: 1054 bytes on wire (8432 bits), 1054 bytes captured (8432 bits)  
 ▶ Point-to-Point Protocol  
 ▶ Internet Protocol Version 4, Src: 10.0.2.1 (10.0.2.1), Dst: 10.0.4.2 (10.0.4.2)  
 ▶ User Datagram Protocol, Src Port: 40104 (40104), Dst Port: discard (9)  
 ▶ Data (1024 bytes)

Obrázek 16. Zahájení vysílání z AS3 do AS6

V čase 105s (104,95) dochází k plánovanému výpadku linky mezi směrovači AS4 a AS5. U těchto směrovačů dojde k úpravě směrovacích tabulek a následně

je odeslána UPDATE zpráva do sousedních směrovačů s informacemi zrušení možnosti směrování do sítí, které měli směrovače nastaveny pro směrování přes nefunkční linku. Přenos zprávy UPDATE, nesoucí informace o zrušení možnosti směrování do daných sítí, konkrétně ze směrovače AS5 do směrovače AS3 můžeme vidět na obrázku č. 17.

Po přijetí této zprávy dochází k aktualizaci směrovací tabulky. Změnu ve směrovací tabulce směrovače AS3 můžeme vidět na obrázcích č. 15., na kterém je zachycena konfigurace původní a č. 18. s tabulkou upravenou po zpracování zprávy UPDATE obsahující informace o zrušení cest.

No.	Time	Source	Destination	Protocol	Length	Info
622	104.950869	10.0.2.1	10.0.4.2	UDP	1054	Source port: 40104 Destination port: discard
623	104.952142	10.0.2.2	10.0.2.1	BGP	89	UPDATE Message
624	104.952555	10.0.2.1	10.0.2.2	TCP	54	bgp > 36908 [ACK] Seq=365 Ack=265 Win=5792 Len=0 TSval=26265 TSecr=26265
625	104.959565	10.0.2.1	10.0.4.2	UDP	1054	Source port: 40104 Destination port: discard

▶ Frame 623: 89 bytes on wire (712 bits), 89 bytes captured (712 bits)  
 ▶ Point-to-Point Protocol  
 ▶ Internet Protocol Version 4, Src: 10.0.2.2 (10.0.2.2), Dst: 10.0.2.1 (10.0.2.1)  
 ▶ Transmission Control Protocol, Src Port: 36908 (36908), Dst Port: bgp (179), Seq: 230, Ack: 365, Len: 35  
 ▼ Border Gateway Protocol - UPDATE Message  
 Marker: ffffffffffffffffffffffffffffffffff  
 Length: 35  
 Type: UPDATE Message (2)  
 Unfeasible routes length: 12 bytes  
 ▼ Withdrawn routes:  
 ▶ 10.0.3.0/24  
 ▶ 10.0.4.0/24  
 ▶ 10.0.6.0/24  
 Total path attribute length: 0 bytes

Obrázek 17. Přenos zprávy UPDATE mezi směrovači AS5 a AS3 s informacemi o nedostupnosti sítí.

Time: 105s

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
127.0.0.0	0.0.0.0	255.0.0.0	U	0	-	-	0
10.0.1.0	0.0.0.0	255.255.255.0	U	0	-	-	1
10.0.2.0	0.0.0.0	255.255.255.0	U	0	-	-	2
10.0.6.0	10.0.1.1	255.255.255.0	UGS	1	-	-	1
10.0.5.0	10.0.2.2	255.255.255.0	UGS	1	-	-	2
10.0.3.0	10.0.1.1	255.255.255.0	UGS	0	-	-	1
10.0.4.0	10.0.1.1	255.255.255.0	UGS	0	-	-	1
10.0.0.0	10.0.1.1	255.255.255.0	UGS	1	-	-	1

Obrázek 18. Navazování spojení mezi peer směrovači AS3 a AS5

Na základě změny ve směrovací tabulce směrovače AS3 dochází k přesměrování datového toku odesílaného do AS6 přes směrovač AS2. Před samotným přesměrováním datového toku směrovač AS3 odesílá do směrovače

AS2 zprávu UPDATE informacemi o zrušení cest. Vyslání UPDATE zprávy ze směrovače AS3 do směrovače AS2 a následné přesměrování datového toku můžeme vidět na obrázku č. 19.

No.	Time	Source	Destination	Protocol	Length	Info
36	72.502946	10.0.1.1	10.0.1.2	TCP	54	bgp > 41783 [ACK] Seq=282 Ack=282 Win=5792 Len=0 TSval=18152 TSecr=18152
37	105.002142	10.0.1.2	10.0.1.1	BGP	85	UPDATE Message
38	105.006364	10.0.1.1	10.0.1.2	TCP	54	bgp > 41783 [ACK] Seq=282 Ack=313 Win=5792 Len=0 TSval=26278 TSecr=26278
39	105.020434	10.0.1.2	10.0.4.2	UDP	1054	Source port: 40104 Destination port: discard
40	105.029130	10.0.1.2	10.0.4.2	UDP	1054	Source port: 40104 Destination port: discard
41	105.037825	10.0.1.2	10.0.4.2	UDP	1054	Source port: 40104 Destination port: discard

▶ Frame 37: 85 bytes on wire (680 bits), 85 bytes captured (680 bits)  
 ▶ Point-to-Point Protocol  
 ▶ Internet Protocol Version 4, Src: 10.0.1.2 (10.0.1.2), Dst: 10.0.1.1 (10.0.1.1)  
 ▶ Transmission Control Protocol, Src Port: 41783 (41783), Dst Port: bgp (179), Seq: 282, Ack: 282, Len: 31  
 ▼ Border Gateway Protocol - UPDATE Message  
 Marker: ffffffffffffffffffffffffffffffffff  
 Length: 31  
 Type: UPDATE Message (2)  
 Unfeasible routes length: 8 bytes  
 ▼ Withdrawn routes:  
 ▶ 10.0.4.0/24  
 ▶ 10.0.6.0/24  
 Total path attribute length: 0 bytes

Obrázek 19. Přesměrování vysílání UDP paketů z AS3 přes peer směrovač AS2

### Chování protokolu při přidání nové linky.

Pro vyhodnocení simulace použijeme výsledky simulace zaznamenané na peer směrovači AS2(quagga-bgpd-1-1.pcap, quagga-bgpd-1-2.pcap, routingtable-every1. Simulátor má nastavenou událost přidání nové linky mezi směrovače AS2 a AS4 v čase 103s. Linka nemá jiné vlastnosti, než mají ostatní linky v síti.

Po přidání nové linky, směrovače AS2 a AS4 zahájí distribuci zpráv UPDATE se směrovacími informacemi o dostupnosti nové sítě 10.0.7.0. Dále si tyto směrovače mezi sebou vzájemně vymění směrovací informace o dostupnosti všech sítí viz obrázek č.20.

No.	Time	Source	Destination	Protocol	Length	Info
5	132.190000	10.0.7.1	10.0.7.2	TCP	62	50445 > bgp [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM=1 TSval=33075 TSecr=0
6	132.194198	10.0.7.2	10.0.7.1	TCP	62	bgp > 50445 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=33075 TSecr=0
7	132.194198	10.0.7.1	10.0.7.2	TCP	54	50445 > bgp [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSval=33076 TSecr=33075
8	132.194285	10.0.7.1	10.0.7.2	BGP	107	OPEN Message
9	132.198542	10.0.7.2	10.0.7.1	TCP	54	bgp > 50445 [ACK] Seq=1 Ack=54 Win=5792 Len=0 TSval=33076 TSecr=33076
10	132.198744	10.0.7.2	10.0.7.1	BGP	126	OPEN Message, KEEPALIVE Message
11	132.198744	10.0.7.1	10.0.7.2	TCP	54	50445 > bgp [ACK] Seq=54 Ack=73 Win=5840 Len=0 TSval=33077 TSecr=33076
12	132.198830	10.0.7.1	10.0.7.2	BGP	92	KEEPALIVE Message, KEEPALIVE Message
13	132.203094	10.0.7.2	10.0.7.1	BGP	73	KEEPALIVE Message
14	132.242000	10.0.7.1	10.0.7.2	TCP	54	50445 > bgp [ACK] Seq=92 Ack=92 Win=5840 Len=0 TSval=33088 TSecr=33077
15	133.198744	10.0.7.1	10.0.7.2	BGP	337	UPDATE Message, UPDATE Message, UPDATE Message, UPDATE Message, UPDATE Message
16	133.203542	10.0.7.2	10.0.7.1	BGP	353	UPDATE Message, UPDATE Message, UPDATE Message, UPDATE Message, UPDATE Message

Obrázek 20. Přenos zpráv UPDATE mezi směrovači AS2 a AS4.

Tyto zprávy směrovače vyhodnotí a upraví směrovací tabulky. Příklad změny ve směrovací tabulce můžeme vidět na obrázku č.21. Jedná se o výpis směrovací tabulky směrovače AS2.

Time: 130s							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
127.0.0.0	0.0.0.0	255.0.0.0	U	0	-	-	0
10.0.0.0	0.0.0.0	255.255.255.0	U	0	-	-	1
10.0.1.0	0.0.0.0	255.255.255.0	U	0	-	-	2
10.0.7.0	0.0.0.0	255.255.255.0	U	0	-	-	3
10.0.2.0	10.0.1.2	255.255.255.0	UGS	1	-	-	2
10.0.3.0	10.0.0.1	255.255.255.0	UGS	1	-	-	1
10.0.6.0	10.0.0.1	255.255.255.0	UGS	1	-	-	1
10.0.4.0	10.0.0.1	255.255.255.0	UGS	1	-	-	1
10.0.5.0	10.0.1.2	255.255.255.0	UGS	0	-	-	2

Time: 135s							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
127.0.0.0	0.0.0.0	255.0.0.0	U	0	-	-	0
10.0.0.0	0.0.0.0	255.255.255.0	U	0	-	-	1
10.0.1.0	0.0.0.0	255.255.255.0	U	0	-	-	2
10.0.7.0	0.0.0.0	255.255.255.0	U	0	-	-	3
10.0.2.0	10.0.1.2	255.255.255.0	UGS	1	-	-	2
10.0.3.0	10.0.0.1	255.255.255.0	UGS	1	-	-	1
<u>10.0.6.0</u>	<u>10.0.7.2</u>	<u>255.255.255.0</u>	<u>UGS</u>	<u>1</u>	<u>-</u>	<u>-</u>	<u>3</u>
<u>10.0.4.0</u>	<u>10.0.7.2</u>	<u>255.255.255.0</u>	<u>UGS</u>	<u>1</u>	<u>-</u>	<u>-</u>	<u>3</u>
10.0.5.0	10.0.1.2	255.255.255.0	UGS	0	-	-	2

Obrázek 21. Směrovací tabulka směrovače AS2 před a po aktualizaci.

Po zpracování směrovacích informací a zapsání do tabulek, směrovače tyto změny dále distribuují dalším sousedním směrovačům, příklad takové zprávy můžeme vidět na obrázku č.22. Tato komunikace probíhá v celé síti, dokud nedojde k vyhodnocení těchto informací ve všech směrovačích a zapsání do směrovacích tabulek.

No.	Time	Source	Destination	Protocol	Length	Info
1783	134.198744	10.0.1.1	10.0.1.2	BGP	165	UPDATE Message, UPDATE Message
1784	134.203094	10.0.1.2	10.0.1.1	TCP	54	bgp > 37939 [ACK] Seq=332 Ack=

```

▶ Transmission Control Protocol, Src Port: 37939 (37939), Dst Port: bgp (179), Seq: 388, Ack: 332, Len:
▼ Border Gateway Protocol - UPDATE Message
  Marker: ffffffffffffffffffffffffffffffff
  Length: 55
  Type: UPDATE Message (2)
  Unfeasible routes length: 0 bytes
  Total path attribute length: 28 bytes
  ▼ Path attributes
    ▶ ORIGIN: INCOMPLETE (4 bytes)
    ▶ AS_PATH: 2 (10 bytes)
    ▶ NEXT_HOP: 10.0.1.1 (7 bytes)
    ▶ MULTI_EXIT_DISC: 1 (7 bytes)
  ▼ Network layer reachability information: 4 bytes
    ▶ 10.0.7.0/24
▼ Border Gateway Protocol - UPDATE Message
  Marker: ffffffffffffffffffffffffffffffff
  Length: 56
  Type: UPDATE Message (2)
  Unfeasible routes length: 0 bytes
  Total path attribute length: 25 bytes
  ▼ Path attributes
    ▶ ORIGIN: INCOMPLETE (4 bytes)
    ▶ AS_PATH: 2 4 (14 bytes)
    ▶ NEXT_HOP: 10.0.1.1 (7 bytes)
  ▼ Network layer reachability information: 8 bytes
    ▶ 10.0.6.0/24
    ▶ 10.0.4.0/24

```

Obrázek 22. Distribuce zpráv UPDATE ze směrovače AS2 do směrovače AS3.

### Chování protokolu při přidání nového autonomního systému do sítě.

Simulátor má nastavenou událost přidání nového autonomního systému AS8 v čase 140s. Pro vyhodnocení simulace použijeme výsledky simulace zaznamenané na peer směrovači AS8(quagga-bgpd-7-0.pcap, quagga-bgpd-7-1.pcap, routingtableevery7).

Po připojení nového autonomního systému AS8 k autonomním systémům AS6 a AS7 zahájí směrovače AS6 a AS7 distribuci zpráv UPDATE k sousedním směrovačům s informacemi o dostupnosti nových sítí. Směrovač AS6 o dostupnosti sítě 10.0.8.0/24 a směrovač AS7 o dostupnosti sítě 10.0.9.0/24. Sousední směrovače tuto zprávu zpracují, dochází k přepsání směrovacích tabulek a poté tuto informaci o dostupnosti nové sítě distribuují dalším sousedům. Tato komunikace probíhá v celé síti, dokud nedojde k vyhodnocení těchto informací ve všech směrovacích a zapsání do směrovacích tabulek.

Směrovač AS8 přijímá v čase 251,4s UPDATE zprávu ze směrovače AS6 o dostupnosti sítí, směrovač AS8 tuto zprávu zpracuje, zapíše do směrovací tabulky a dále tyto směrovací informace přeposílá do směrovače AS7. V čase 254,7s přijímá směrovač AS8 zprávu UPDATE se směrovacími informacemi vyslanou směrovačem AS7. Směrovač AS8 tuto zprávu zpracuje a na základě atributu AS\_PATH upraví směrovací tabulku a poté změny ve směrovací tabulce distribuují dále směrovači AS6. Upravenou směrovací tabulku směrovače AS8 můžeme



vidět na obrázku č.23. V případě změn ve směrovacích tabulkách směrovačů, směrovače tyto změny distribuují dalším sousedům, které tyto UPDATE zprávy zpracovávají a případné změny ve směrovacích tabulkách distribuují dále.

```

Time: 255s
Destination Gateway Genmask Flags Metric Ref Use Iface
127.0.0.0 0.0.0.0 255.0.0.0 U 0 - - 0
10.0.8.0 0.0.0.0 255.255.255.0 U 0 - - 1
10.0.9.0 0.0.0.0 255.255.255.0 U 0 - - 2
10.0.4.0 10.0.8.1 255.255.255.0 UGS 1 - - 1
10.0.5.0 10.0.9.1 255.255.255.0 UGS 1 - - 2
10.0.1.0 10.0.8.1 255.255.255.0 UGS 0 - - 1
10.0.2.0 10.0.9.1 255.255.255.0 UGS 0 - - 2
10.0.6.0 10.0.8.1 255.255.255.0 UGS 0 - - 1
10.0.0.0 10.0.8.1 255.255.255.0 UGS 0 - - 1
10.0.3.0 10.0.8.1 255.255.255.0 UGS 0 - - 1
10.0.7.0 10.0.8.1 255.255.255.0 UGS 0 - - 1

```

Obrázek 23. Směrovací tabulka směrovače AS8.

## 7.2. Chování BGP při nastavení atributů WEIGHT, LOCAL\_PREF a MED

Tato úloha ověřuje ovlivnění výběru cest při směrování pomocí atributů WEIGHT, LOCAL\_PREF a MED.

### 7.2.1. Instalace NS-2

NS lze nainstalovat na různých platformách, jako jsou Unix(Linux), Windows a Mac systémy. Jelikož je NS vyvíjen v prostředí Unix, je tento systém pro běh NS nejvhodnější. Pro používání simulátoru ve Windows je nutné použít Cygwin(Unix emulátor). Zdrojové kódy NS-2 jsou distribuované ve dvou formách, v all-in-one sadě a po jednotlivých komponentách. All-in-one sada obsahuje všechny potřebné součásti spolu s některými volitelnými komponentami. Součástí tohoto balíčku je `install` skript, který konfiguruje NS-2 prostředí a vytvoří spustitelný soubor. Nevýhodou All-in-one sady je velikost, protože obsahuje některé komponenty, které nepotřebujeme pro sestavení ns-2 a NAM (Network Animator). Sadu All-in-one, nebo jednotlivé komponenty je možno získat ze stránky: <http://www.isi.edu/nsnam/ns/ns-build.html>



Při simulaci bude použita implementace BGP++, která byla vytvořena pro starší verze simulátoru NS-2. Bude použit NS-2 verze 2.26. Dále z důvodu nutnosti staršího kompilátoru gcc(maximálně verze 2.3.2), bude simulátor nainstalován ve starší verzi OS GNU/Linux, konkrétně Ubuntu 9.04.

Před samotnou instalací simulátoru NS-2 je potřeba nainstalovat požadované balíčky a nástroje. Konkrétně se jedná o kompilátor jazyka C++, v našem případě verzi gcc 2.3.2, dále knihovny grafického prostředí X11(libx11-devel, xorg-x11-proto-devel).

Instalaci simulátoru provedeme následnou sekvencí příkazů:

```
./configure  
make  
./install
```

Pro kontrolu správné instalace spustíme příkaz:

```
./validate
```

### 7.2.2. Vytváření skriptu simulace

Na obrázku č. 24. můžeme vidět návrh sítě, kterou použijeme při simulaci. Skládá se z devíti uzlů, které budou simulovat peer směrovače. Směrovače jsou dále rozděleny do čtyř autonomních systémů. Linky spojující uzly budou mít nastavenou šířku pásma 1Mbit a zpoždění 1ms.

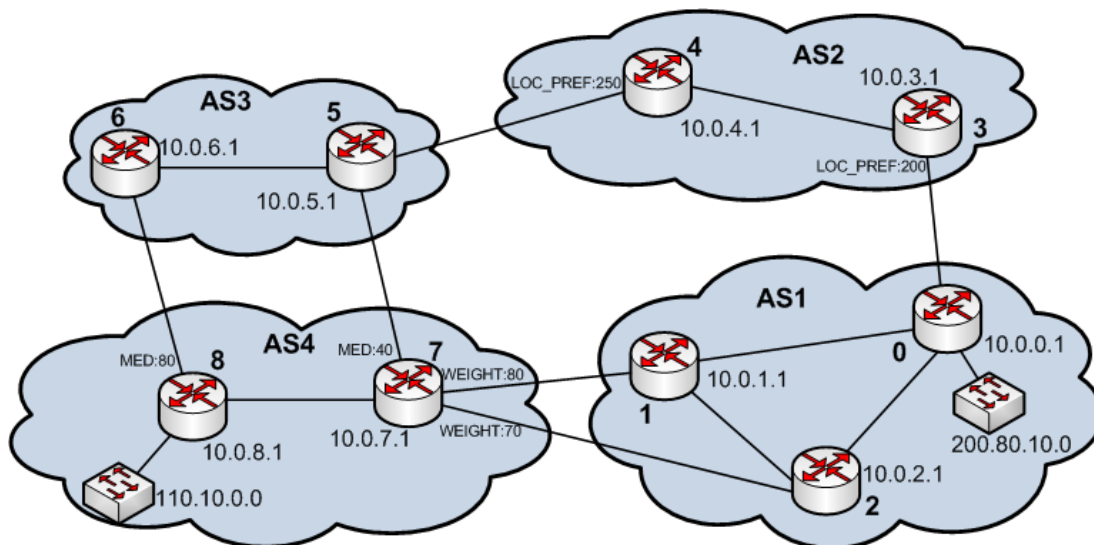
Síť byla navržena takovým způsobem, abychom mohli ověřit funkčnost atributů WEIGHT, MED a LOCAL\_PREF. Atribut WEIGHT bude testován v autonomním systému č.4 na směrovači č.7, který má se sousedním směrovačem č.2 nastavenou hodnotu WEIGHT 70 a s druhým sousedním směrovačem č.1 hodnotu 80. Atribut MED bude testován v autonomním systému č.3, kde se bude rozhodovat o výběru cesty k síti vytvořené v autonomním systému č.4 pomocí atributů zaslanych z AS4 směrovače č.7, kde bude hodnota MED nastavena na 40 a ze směrovače č.8, kde bude hodnota MED nastavena na 80. Atribut LOCAL\_PREF bude testován v autonomním systému č.2, kde na lince do směrovače č.5 bude nastavena hodnota LOCAL\_PREF 250 a na lince do směrovače č.0 hodnota 200.

Pro psaní simulačních scénářů můžeme použít jakéhokoliv textového editoru, jako je třeba joe, nebo emacs.

V první řadě musíme vytvořit objekt simulátoru, to se provádí pomocí příkazu:

```
set ns [new Simulator]
```

Dále otevřeme trasovací soubor pro zápis, používaný pro zobrazení simulace v NAM.



Obrázek 24. Návrh simulované sítě

```
set nnam [open out.nam w]
$ns namtrace-all $nnam
```

Dále přidáme finish proceduru, která uzavře trasovací soubor a spustí NAM.

```
proc finish { } {
    global ntrace nnam ns
    if [info exists nnam] {
        close $nnam
    }
    exec nam out.nam
    exit 0
}
```

Dále si definujeme uzly.

```
set n0 [$ns node]
set n1 [$ns node]
```

Definujeme linku mezi uzly.

```
$ns duplex-link $n0 $n1 1Mb 1ms DropTail
```

Vytvoříme registr, který představuje databázi, ve které se mapují IP adresy k jednotlivým BGP instancím. [13]

```
set reg [new BgpRegistry]
```

Vytvoříme BGP instanci, ke které namapujeme IP adresu z registru. Dále oznámíme BGP instanci čas ukončení. Pomocí příkazu `config-fil` přiřadíme BGP instanci konfigurační soubor, a poté připojíme BGP instanci k uzlu.

```
set BGPO [new Application/Route/Bgp]
$BGPO register $reg
$BGPO finish-time 1000
$BGPO config-file ./bgp0.conf
$BGPO attach-node $n0
```

Konfigurace směrovačů se provádí pomocí konfiguračních souborů, kde se definuje číslo autonomního systému ve kterém se směrovač nachází, IP adresa, sousedící BGP směrovače, prefixy směrovacích cest, povolení logování do souboru.

Konfigurační soubor `bgp0.conf` :

```
! Prirazeni uzlu k autonomnimu systemu(AS1)
router bgp 1
! Nastaveni IP adresy smerovace
bgp router-id 10.0.0.1
! Nastaveni smerovacich informaci k susednim BGP smerovacum
neighbor 10.0.1.1 remote-as 1
neighbor 10.0.2.1 remote-as 1
neighbor 10.0.3.1 remote-as 2
! Povoleni ruznych urovni ladeni s vystupem do souboru log.
debug bgp
debug bgp fsm
debug bgp keepalives
debug bgp filters
debug bgp events
debug bgp updates
! Vytvoreni souboru log.
log file bgp0.log
```

Na jednotlivých směrovačích dále nastavíme dané atributy podle návrhu simulované sítě:

- směrovač č.7 autonomního systému AS4. Nastavení atributu `WEIGHT` na směr k jednotlivým směrovačům na hodnoty 80 ke směrovači č.1 a na hodnotu 70 ke směrovači č.2 oba autonomního systému AS2.

```
neighbor 10.0.1.1 weight 80
neighbor 10.0.2.1 weight 70
```

- směrovač č.7 autonomního systému AS4. Nastavení atributu `MED` na hodnotu 40 k cestám odesílaných do směrovače č.5 autonomního systému AS3.

```
route-map setmetricout permit 10
set metric 40
```

- směrovač č.8 autonomního systému AS4. Nastavení atributu MED na hodnotu 80 k cestám odesílaných do směrovače č.6 autonomního systému AS3.

```
route-map setmetricout permit 10
set metric 80
```

- směrovač č.3 autonomního systému AS2. Nastavení atributu LOCAL\_PREF na hodnotu 200.

```
neighbor 10.0.0.1 route-map setlocpref in
route-map setlocpref permit 10
set local-preference 200
```

- směrovač č.4 autonomního systému AS2. Nastavení atributu LOCAL\_PREF na hodnotu 250.

```
neighbor 10.0.5.1 route-map setlocpref in
route-map setlocpref permit 10
set local-preference 250
```

Příkazem `command` měníme stávající konfiguraci směrovačů, v tomto případě ve směrovači č.5 uplatňujeme atribut MED.

```
$ns at 0.2 "$BGP5 command \"bgp always-compare-med\""
```

Vytvoříme novou síť ve směrovači č.0.

```
$ns at 10 "$BGP0 command \"network 200.80.10.0/24\""
```

Tento příkaz slouží k vytisknutí směrovací tabulky instance BGP0 do souboru `log`.

```
$ns at 199 "$BGP0 command \"show ip bgp\""
```

### 7.2.3. Spuštění simulace

Za předpokladu, že jsme v adresáři se spustitelným `ns`, nebo je nastavená cesta k adresáři s tímto souborem, simulaci spustíme příkazem:

```
ns tlcskript
```

kde `tlcskript` je název souboru skriptu Tcl, který definuje simulační scénář. [1]

#### 7.2.4. Vyhodnocení simulace

Průběh celé simulace je monitorován na každém BGP uzlu. Výsledky ověřování jsou textového charakteru výpisem ze souborů log.

##### Ověření ovlivnění směrování pomocí atributu WEIGHT.

Atribut WEIGHT ověříme výpisem ze směrovací tabulky na směrovači č.7 autonomního systému AS4 (bgp7.log). Jelikož se atribut WEIGHT nastavuje pouze pro lokální směrovač, bude nám k vyhodnocení ovlivnění cesty stačit výpis ze směrovací tabulky daného směrovače.

```
BGP table version is 0, local router ID is 10.0.7.1
Status codes: s suppressed, d damped, h history,
               * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*	200.80.10.0/24	10.0.5.1			0	3 2 1i
*		10.0.2.1			70	1i
*>		10.0.1.1			80	1i
*>	i110.10.0.0/24	10.0.8.1	100		0	i

Z výpisu směrovací tabulky můžeme vidět nastavení atributu WEIGHT. Při směrování do sítě 200.80.10.0 bude směrovač č.7 upřednostňovat (znak > ) cestu přes směrovač 10.0.1.1 (č.1) z důvodu vyšší váhy této cesty.

##### Ověření ovlivnění směrování pomocí atributu MULTI EXIT DISC.

Fungování atributu si ukážeme na zprávách UPDATE, které distribuují IP adresu sítě 110.10.0.0, vytvořené v AS4 směrovači č.8.

K vyhodnocení byli použity výpisy souborů log směrovačů číslo 5, 6, 7 a 8.

Podle nastavených událostí byla v průběhu simulace vytvořena ve směrovači č.8 autonomního systému AS4 síť 110.10.0.0/24. Následně s tohoto směrovače byla odeslána zpráva UPDATE do směrovače č.6 autonomního systému AS3 s distribucí směrovacích informací k nově vzniklé síti. Jednou z přenášených informací je informace o nastaveném atributu MED s hodnotou 80:

```
9.829198 BGP: 10.0.6.1 send UPDATE 110.10.0.0/24 nexthop 10.0.8.1,
              origin i, metric 80, path
```

Směrovač č.6 autonomního systému AS3 přijal zprávu UPDATE v čase 9.834030 vyslanou směrovačem č.8 autonomního systému AS4. Tato zpráva obsahuje směrovací informace do sítě 110.10.0.0/24, tato zpráva obsahuje také informaci o nastaveném atributu MED s hodnotou 80. Dále směrovač tuto informaci

o dostupnosti sítě redistribuuje ve zprávě UPDATE do směrovače č.5 v rámci autonomního systému AS3. Hodnotu atributu LOCAL\_PREF(100) si v rámci autonomního systému nastavuje směrovací algoritmus sám:

```
9.834030 BGP: 10.0.8.1 rcvd UPDATE w/ attr: nexthop 10.0.8.1,
              origin i, metric 80, path 4
9.834030 BGP: 10.0.8.1 rcvd 110.10.0.0/24
10.362794 BGP: 10.0.5.1 send UPDATE 110.10.0.0/24 nexthop 10.0.8.1,
              origin i, localpref 100, metric 80, path 4
```

Směrovač č.5 autonomního systému AS3 přijal zprávu UPDATE v čase 10.367682 vyslanou směrovačem č.6 se směrovacími informacemi do sítě 110.10.0.0/24. Tato zpráva obsahuje také informace o nastaveném atributu MED s hodnotou 80. Další přijatou zprávou je zpráva UPDATE ze směrovače č.7 autonomního systému AS4 se směrovacími informacemi do sítě 110.10.0.0/24, tato zpráva obsahuje také informaci o nastaveném atributu MED s hodnotou 40. Dále směrovač č.5 tuto směrovací informaci přepošle ve zprávě UPDATE do směrovače č.6 v rámci autonomního systému AS3:

```
10.367682 BGP: 10.0.6.1 rcvd UPDATE w/ attr: nexthop 10.0.8.1,
              origin i, localpref 100, metric 80, path 4
10.367682 BGP: 10.0.6.1 rcvd 110.10.0.0/24
10.863634 BGP: 10.0.7.1 rcvd UPDATE w/ attr: nexthop 10.0.7.1,
              origin i, metric 40, path 4
10.863634 BGP: 10.0.7.1 rcvd 110.10.0.0/24
10.863634 BGP: 10.0.6.1 send UPDATE 110.10.0.0/24 nexthop 10.0.7.1,
              origin i, localpref 100, metric 40, path 4
```

Směrovač č.8 autonomního systému AS4 odesílá v čase 10.857250 zprávu UPDATE do směrovače č.7 v rámci autonomního systému AS4 o dostupnosti sítě 110.10.0.0/24 s nastaveným atributem LOC\_PREF(100):

```
10.857250 BGP: 10.0.7.1 send UPDATE 110.10.0.0/24 nexthop 10.0.8.1,
              origin i, localpref 100, path
```

V čase 10.861898 směrovač č.7 autonomního systému AS4 přijímá zprávu UPDATE vyslanou ze směrovače č.8 stejného autonomního systému s informacemi o dostupnosti sítě 110.10.0.0/24, které následně redistribuuje do směrovače č.5 autonomního systému AS3 s přidáním atributem MED s hodnotou 40:

```
10.861898 BGP: 10.0.8.1 rcvd UPDATE w/ attr: nexthop 10.0.8.1,
              origin i, localpref 100, path
10.861898 BGP: 10.0.8.1 rcvd 110.10.0.0/24
10.861898 BGP: 10.0.5.1 send UPDATE 110.10.0.0/24 nexthop 10.0.7.1,
              origin i, metric 40, path
```

Směrovač č.6 autonomního systému AS3 přijímá v čase 10.865426 zprávu UPDATE se směrovacími informacemi o dostupnosti sítě 110.10.0.0/24 ze směrovače č.5 stejného autonomního systému. Součástí těchto informací je i nastavený atribut MED na hodnotu 40. Následuje redistribuce cesty do směrovače č.8 autonomního systému AS4:

```
10.865426 BGP: 10.0.5.1 rcvd UPDATE w/ attr: nexthop 10.0.7.1,
              origin i, localpref 100, metric 40, path 4
10.865426 BGP: 10.0.5.1 rcvd 110.10.0.0/24
10.865426 BGP: 10.0.8.1 send UPDATE 110.10.0.0/24 nexthop 10.0.6.1,
              origin i, metric 40, path 4
```

Směrovač č.8 autonomního systému AS4 přijímá zprávu UPDATE v čase 10.867122 vyslanou směrovačem č.6 autonomního systému AS3 se směrovacími informacemi do sítě 110.10.0.0/24. Po přijetí této zprávy byla detekována smyčka, tím pádem není cesta povolena k zápisu do směrovací tabulky:

```
10.867122 BGP: 10.0.6.1 rcvd UPDATE w/ attr: nexthop 10.0.6.1,
              origin i, path 3 4
10.867122 BGP: 10.0.6.1 rcvd UPDATE about 110.10.0.0/24
              -- DENIED due to: as-path contains our own AS;
```

Níže vidíme dva výpisy ze směrovacích tabulek, konkrétně ze směrovače č.5 a směrovače č.6 oba s autonomního systému AS3. Zde můžeme vidět jakým způsobem ovlivňuje atribut MED výběr cesty ze sousedního autonomního systému.

Ve výpisech jsou uvedeny IP adresa daného směrovače, prefixy sítí, atributy Next hop, MED (metric), LocPref a atribut AS\_Path. Tabulka udává svou verzi - kolikrát byla měněna (BGP table version is). Na začátku záznamu jsou uvedeny kódy, zda je cesta platná (\*), nejlépe vybraná (>), nebo naučená přes iBGP (i).

Směrovač č.5 AS3: Výpis ze směrovací tabulky.

```
BGP table version is 0, local router ID is 10.0.5.1
Status codes: s suppressed, d damped, h history,
               * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* 200.80.10.0/24	10.0.7.1	40		0	4 1i
*>	10.0.4.1			0	2 1i
*> 110.10.0.0/24	10.0.7.1	40		0	4i

Směrovač č.6 AS3: Výpis ze směrovací tabulky.

BGP table version is 0, local router ID is 10.0.6.1

Status codes: s suppressed, d damped, h history,

\* valid, > best, i - internal

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
* 200.80.10.0/24	10.0.8.1	80		0	4 1i
*> i	10.0.4.1		100	0	2 1i
*> i110.10.0.0/24	10.0.7.1	40	100	0	4i
*	10.0.8.1	80		0	4i

### Ověření ovlivnění směrování pomocí atributu LOC\_PREF.

Fungování atributu si ukážeme na zprávách UPDATE, které distribuují IP adresu sítě 110.10.0.0, vytvořené v AS4 směrovači č.8.

K vyhodnocení byli použity výpisy souborů log směrovačů číslo 3 a 4.

Směrovač č.4 přijal v čase 10.369378 zprávu UPDATE, kterou odeslal směrovač č.5 autonomního systému AS3, zpráva obsahuje směrovací informace o síti 110.10.0.0/24. Směrovač zprávu zpracuje, nastaví atribut LOC\_PREF na hodnotu 250 a následně distribuuje v rámci autonomního systému AS2 do směrovače č.3:

```
10.369378 BGP: 10.0.5.1 rcvd UPDATE w/ attr: nexthop 10.0.5.1,  
origin i, path 3 4
```

```
10.369378 BGP: 10.0.5.1 rcvd 110.10.0.0/24
```

```
10.369378 BGP: 10.0.3.1 send UPDATE 110.10.0.0/24 nexthop 10.0.5.1,  
origin i, localpref 250, path 3 4
```

Směrovač č.3 v čase 10.371130 přijal zprávu UPDATE vyslanou směrovačem č.4 s informacemi o dostupnosti sítě 110.10.0.0/24 s nastaveným atributem LOC\_PREF s hodnotou 250. Dále v čase 12.001680 přijal směrovač č.3 UPDATE zprávu vyslanou směrovačem č.0 autonomního systému AS1 s informacemi o dostupnosti sítě 200.80.10.0/24. Směrovač zprávu zpracuje nastaví atribut LOC\_PREF na hodnotu 200 a následně distribuuje do směrovače č.4:

```
10.371130 BGP: 10.0.4.1 rcvd UPDATE w/ attr: nexthop 10.0.5.1,  
origin i, localpref 250, path 3 4
```

```
10.371130 BGP: 10.0.4.1 rcvd 110.10.0.0/24
```

```
12.001680 BGP: 10.0.0.1 rcvd UPDATE w/ attr: nexthop 10.0.0.1,  
origin i, path 1
```

```
12.001680 BGP: 10.0.0.1 rcvd 200.80.10.0/24
```

```
12.001680 BGP: 10.0.4.1 send UPDATE 200.80.10.0/24 nexthop 10.0.0.1,  
origin i, localpref 200, path 1
```



V čase 12.003416 přijímá směrovač č.4 zprávu UPDATE vyslanou směrovačem č.3 se směrovacími informacemi o síti 200.80.10.0/24 s nastaveným atributem LOC\_PREF na hodnotu 200:

```
12.003416 BGP: 10.0.3.1 rcvd UPDATE w/ attr: nexthop 10.0.0.1,
              origin i, localpref 200, path 1
12.003416 BGP: 10.0.3.1 rcvd 200.80.10.0/24
```

Na výpisu ze směrovacích tabulek směrovačů č.3 a č.4 autonomního systému AS2 můžeme vidět, jakým způsobem ovlivňuje atribut LOC\_PREF výběr cest.

Ve výpisech jsou uvedeny IP adresa daného směrovače, prefixy sítí, atributy Next hop, MED (metric), LocPref a atribut AS\_Path. Tabulka udává svou verzi - kolikrát byla měněna (BGP table version is). Na začátku záznamu jsou uvedeny kódy, zda je cesta platná (\*), nejlépe vybraná (>), nebo naučená přes iBGP (i).

Směrovač č.3 AS2: Výpis směrovací tabulky.

```
BGP table version is 0, local router ID is 10.0.3.1
Status codes: s suppressed, d damped, h history,
              * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 200.80.10.0/24	10.0.0.1		200	0	1i
* 110.10.0.0/24	10.0.0.1		200	0	1 4i
*> i	10.0.5.1		250	0	3 4i

Směrovač č.4 AS2: Výpis směrovací tabulky.

```
BGP table version is 0, local router ID is 10.0.4.1
Status codes: s suppressed, d damped, h history,
              * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> i200.80.10.0/24	10.0.0.1		200	0	1i
*> 110.10.0.0/24	10.0.5.1		250	0	3 4i

## Závěr

Cílem bakalářské práce bylo prostudovat problematiku simulace počítačových sítí, se zaměřením na síťové simulátory NS-2 a NS-3. Dalším cílem byl popis směrovacího protokolu BGP a tento protokol použít při simulaci počítačové sítě na simulátorech NS-3 a NS-2.

První část části teoretické se zaměřuje na popis protokolu IP, rozdělení směrovacích protokolů, na protokol TCP, navazování a ukončování spojení tímto protokolem, na protokol UDP. Dále je v práci rozebráno, co jsou to autonomní systémy a směrovací protokol BGP.

Druhá část teoretické části práce obsahuje popis teorie simulací, na kterou navazuje popis simulátorů NS-2 a NS-3, jaké protokoly podporují, jaké jsou základní objekty těchto simulátorů. U simulátoru NS-3 je popsán nástroj pro přímé vykonávání kódu (DCE) a balíček Quagga, které jsou použity v praktické části bakalářské práce.

Praktická část bakalářské práce se zaměřuje na rozbor výsledků provedených simulací. První simulace se zaměřuje na popis navazování spojení pomocí protokolu TCP, chování protokolu BGP, konkrétně na zřizování vazby mezi směrovači pomocí zprávy OPEN, dále na zasílání aktualizací do směrovacích tabulek pomocí zprávy UPDATE, zvolení směrování na základě atributu AS\_PATH při přenosu dat, na udržování spojení pomocí zprávy KEEPALIVE. Dále se zaměřuje na chování protokolu při výpadku linky v průběhu přenosu dat, na chování při přidání nové linky mezi autonomní systémy a chování při přidání nového autonomního systému do sítě. K této simulaci byl použito simulátoru NS-3. Druhá simulace se zaměřuje na ověření ovlivnění výběru cest při směrování pomocí atributů WEIGHT, LOCAL\_PREF a MED. K této simulaci byl použit simulátor NS-2. Součástí praktické práce je popis instalace simulátorů a popis vytváření simulačních skriptů.

## Reference

- [1] *The Network Simulator - ns-2*, [online] <http://www.isi.edu/nsnam/ns/>
- [2] T. Issariyakul, E. Hossain : *Introduction to Network Simulator NS2*. Springer, 2009.
- [3] *The network simulator NS-3* [online] <http://www.nsnam.org/>
- [4] E. Altman, T. Jimenéz : *NS Simulator for beginners, Lecture notes*. Universita de Los Andes, Venezuela, 2003.
- [5] L. Dostálek, A. Kabelová : *Velký průvodce protokoly TCP/IP a systémem DNS*. Computer Press, Brno, 2012
- [6] J. Jeřábek : *Skripta k předmětu Pokročilé komunikační techniky*. Vysoké učení technické v Brně, Brno, 2009.
- [7] P. Grygárek : *Směrovací protokol BGP*, [online] <http://www.cs.vsb.cz/grygarek/SPS/lect/BGP/BGP.html>
- [8] S. H. Y. Rekhter, T. Li : *A Border Gateway Protocol 4 (BGP-4)*. 2006. [online] <http://www.faqs.org/rfcs/rfc4271.html>
- [9] Wikipedia - Internet Protocol, [online] [http://www.cs.wikipedia.org/wiki/Internet\\_Protocol](http://www.cs.wikipedia.org/wiki/Internet_Protocol)
- [10] O. Wendell : *CCENT/CCNA ICND1 640-822 Official Cert Guide*. Third Edition. Cisco Press, 2011.
- [11] T. Lammler : *CCNA výukový průvodce přípravou na zkoušku 640-802*. Computer Press, 2010.
- [12] *NS-3 Quagga Documentation*, [online] <https://www.nsnam.org/docs/dce/manual-quagga/html/getting-started.html>
- [13] *BGP++* [online] <http://www.ece.gatech.edu/research/labs/MANIACS/BGP++/>