



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

POROVNANIE DISTANCE-VECTOR SMEROVACÍCH PROTOKOLOV

COMPARISON OF DISTANCE-VECTOR ROUTING PROTOCOL

BAKALÁRSKA PRÁCA
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MAROŠ COCUĽA

VEDÚCI PRÁCE
SUPERVISOR

Ing. VLADIMÍR VESELÝ, Ph.D.

BRNO 2017

Zadání bakalářské práce

Řešitel: **Cocuľa Maroš**

Obor: Informační technologie

Téma: **Porovnání distance-vector směrovacích protokolů
Comparison of Distance-Vector Routing Protocol**

Kategorie: Počítačové sítě

Pokyny:

1. Analyzujte směrovací protokoly pracující na principu distance-vector, konkrétně protokoly RIP, EIGRP a Babel.
2. Zjistěte aktuální stav a nasazení těchto protokolů v simulátoru OMNeT++ a na reálných zařízeních.
3. Dle doporučení vedoucího navrhnete simulační scénáře pro porovnávání těchto protokolů mezi sebou, zaměřte se na rychlost konvergence, množství vyměněných zpráv a jiné protokolové metriky.
4. Proveďte měření a porovnání, diskutujte a analyzujte dosažené výsledky.

Literatura:

- J. Chroboczek, *The Babel Routing Protocol*, <http://tools.ietf.org/html/rfc6126>.
- D. Savage, D. Slice, *Enhanced Interior Gateway Routing Protocol*, <http://tools.ietf.org/html/draft-savage-eigrp-00>.
- Cisco Systems, *Document ID: 16406 - Enhanced Interior Gateway Routing Protocol*, [Online], http://www.cisco.com/en/US/tech/tk365/technologies_white_paper09186a00800

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2 včetně.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

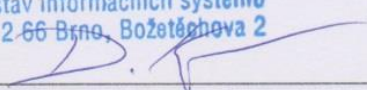
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Veselý Vladimír, Ing., Ph.D.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2


doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Táto práca sa zaoberá porovnaním Distance vector smerovacích protokolov. Presnejšie povedané EIGRP, RIP a Babel. Cieľom je získanie informácií pomocou simulačného nástroja OMNeT++, ale aj reálneho zapojenia. Súčasťou práce je detailný popis protokolov, princípov simulátoru OMNeT++ a jeho knižníc INET a ANSAINET. Následne sú zhrnuté výsledky experimentov a samotné porovnanie protokolov medzi sebou.

Abstract

This work deals with comparing the distance vector routing protocols. More accurately, protocols EIGRP, RIP and Babel. The aim is to compare information using the simulation OMNeT++, as well as the physical connection. Part of the work is a detailed description of the protocols, simulator OMNeT++ and the libraries INET and ANSAINET. Finally, we summarized the results of experiments and the comparison of protocols with each other.

Kľúčové slova

EIGRP, Babel, RIP, simulácie sietí, smerovací protokol, OMNeT++, INET, ANSA

Keywords

EIGRP, Babel, RIP, network simulation, routing protocol, OMNeT++, INET, ANSA

Citácie

COCULA, Maroš. Porovnanie distance-vector smerovacích protokolov. Brno, 2017. Bakalárska práca. Vysoké učenie technické v Brne, Fakulta informačných technológií. Vedúci práce Ing. Vladimír Veselý, Ph.D.

Porovnanie distance-vector smerovacích protokolov

Prehlásenie

Prehlasujem, že som túto prácu vypracoval samostatne pod vedením pána Ing. Vladimíra Veselého, Ph.D.

Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Maroš Cocuľa
16.5.2017

PodĎakovanie

Týmto by som sa chcel poďakovať Ing. Vladimírovi Veselému, Ph.D. za vzornú odbornú pomoc a vedenie v projekte spojené s konzultáciami. Ako odmenu za jeho snahu by som sa s ním rád podelil o môj obľúbený recept na bravčovú panenku na víne plnenú sušenými slivkami. Na prípravu potrebujeme bravčovú panenku, sušené slivky, slivkový lekvár, cibuľu, slaninu, 1 dcl červeného vína a korenie na ochutenie. Cibuľu nakrájame na drobno, opražíme ju s nakrájanou slaninou. Keď je zmes opražená do zlatista, pridáme lyžicu lekváru a necháme prevrieť. Zmes sa zvarí na hustejšiu omáčku. Do panenky nožom vykrojíme tunel po celej dĺžke. Tunel vyplníme sušenými slivkami. Konce upevníme špáradlami. Následne ju osolíme, okoreníme a opečieme na panvici z každej strany. Po opečení panenku vložíme do rúry vyhriatej na 180°C na 20 minút. Hotovú panenku nakrájame na plátky a prelejeme omáčkou. Ako prílohu môžeme zvoliť pečené zemiaky alebo ryžu.

Obsah

1	Úvod.....	3
1.1	Použité pojmy	3
	Diffusing Update Algorithm	6
2	Distance Vector protokoly	7
2.1	Enhanced Interior Gateway Routing Protocol	7
	2.1.1 EIGRP správy.....	7
	2.1.2 Dátové štruktúry.....	8
	2.1.3 Vzťah susedstva.....	8
	2.1.4 Metrika	9
	2.1.5 Spravovanie ciest	10
	2.1.6 Reliable Transport Protocol	10
2.2	Babel	11
	2.2.1 Babel správy.....	11
	2.2.2 Dátové štruktúry.....	12
	2.2.3 Metrika	12
	2.2.4 Podmienky vhodnosti	13
2.3	Routing Information Protokol.....	14
	2.3.1 RIP správy.....	14
	2.3.2 Dátové štruktúry.....	14
	2.3.3 Metrika	14
3	Prostredie OMNeT++	15
	3.1 Moduly.....	15
	3.2 Jazyk NED.....	15
	3.3 INET	16
	3.4 ANSAINET	16
4	Porovnanie jednotlivých protokolov.....	17
	4.1 Topológia porovnáwanej siete.....	17
	4.2 Test ustanovenia stavu susedstva	18

4.2.1 Zhrnutie testu	24
4.3 Porovnanie správ jednotlivých protokolov	25
4.4 Záverečné zhrnutie porovnania	26
5 Záver.....	27
Literatúra	28
A Obsah priloženého CD	30
B Konfigurácia Babeld.....	31
C Konfigurácia EIGRP.....	33
D Konfigurácia RIP.....	35

1 Úvod

Fenomén zvaný Internet je celosvetovo rozšírená komunikačná sieť. Každodenné používanie internetu má na svedomí vyššie nároky na správu a efektivitu internetu. Jednou z metód zefektívniť internet sú aj smerovacie protokoly. Šíria v sieti informácie o smerovaní a umožňujú tak sieti reagovať na zmeny automaticky.

Po nasadení smerovacieho protokolu do siete môžu nastať problémy spôsobené zlou konfiguráciou. Následkom toho sa môže stať sieť nepoužiteľná. Aby sme zabránili problémom, je výhodné konfiguráciu otestovať v simulačnom prostredí ešte pred nasadením. Po vytvorení modelu v simulátore vieme testovať sieť, ale aj vykonávať experimenty. Takto sa dozvieme všetky informácie, ktoré sú potrebné k bezproblémovému behu siete. Jedným z takýchto simulátorov je OMNeT++. Ponúka viaceré možnosti experimentovania a je jednoduchý na používanie.

V tejto práci rozoberáme jednotlivé smerovacie protokoly. V druhej kapitole sú detailne popísané protokoly, ktorými sa budeme zaoberať, a to EIGRP, Babel a RIP. V tretej kapitole je popísaný simulátor OMNeT++ a jeho prostredie. Následne sú v závere práce zhodnotené dosiahnuté výsledky a prínos tejto práce.

1.1 Použité pojmy

IPv4

Je verzia 4 IP protokolu popísaného v RFC791 [18]. Tvorí základ väčšej časti internetu. Používa 32-bitové IP adresy. Súčasne sú všetky adresy vyčerpané a do popredia sa pomaly dostáva IPv6.

IPv6

Je verzia 6 IP protokolu, pôvodne IPng. Hlavným cieľom je nahradenie zastaraného IPv4. Používa 128-bitové adresy. Je popísaná v RFC2460 [19].

Unicast

Metóda posielania IP datagramov jedinému cieľu. Používa sa pre priamu komunikáciu medzi dvoma uzlami v sieti. Cieľ je daný adresou.

Broadcast

Metóda posielania IP datagramov všetkým dostupným cieľom v danej sieti. Používa sa pre šírenie správ v danej podsieti. Pri nesprávnom použití môže zahltiť sieť. Táto metóda je popísaná v RFC919 [20].

Multicast

Je metóda posielania IP datagramov skupine príjemcov. Prenášané dáta sú vo väčšine multimedialne. Prehľad Multicastovej komunikácie nájdeme v RFC6308 [21].

Link-state protokoly

Je jedna z dvoch hlavných kategórií smerovacích protokolov. Automaticky hľadajú susedov na linke a pomocou paketu *Hello* pravidelne testujú ich dostupnosť. Následne vysielajú do siete obežníky s informáciami o susedoch, tým pádom každý smerovač pozná celú topológiu siete.

Classless routing

Smerovače posielajú s adresou siete aj masku, tým pádom sa šetrí adresný priestor. Pre počítanie variabilnej dĺžky masky sa používa VLSM, pre skrátený zápis sa využíva CIDR.

Classfull routing

Smerovače neposielajú v *update* správe údaje o maske. Ak smerovač prijme adresu, rozpozná masku z prvého oktetu podľa jednotlivých tried. Prehľad tried pre IPv4 adresy je v tabuľke 1.1.

Trieda	Prvý oktet
A	1-126
B	128-191
C	192-223
D	224-239
E	240-254

Tabuľka 1.1: Rozdelenie adries podľa tried

Trieda D je určená pre multicast a trieda E pre experimentálne účely.

VLSM (Classless subnetting)

Variable-Length Subnet Mask je metóda smerovania, ktorá umožňuje deliť siete na menšie podsiete. Masky majú variabilnú dĺžku. Sieť sa postupne delí od najväčšej podsiete k najmenšej. Podsiete môžu mať narozdiel od Classfull subnettingu rôzne masky. Rozdelenie siete podľa počtu hostov môžeme nájsť v RFC1878 [22].

CIDR

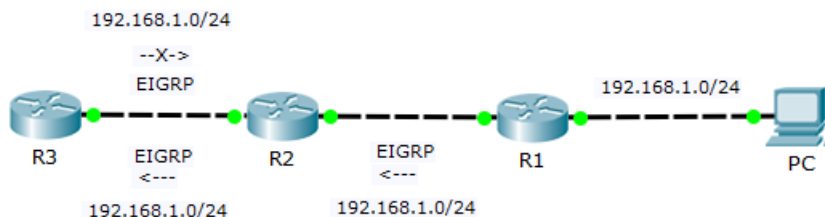
Classless Inter-Domain Routing je štandard zápisu podsiete v tvare *adresa/dĺžka_masky*. Je založený na VLSM smerovaní, umožňuje delenie sietí na viacero podsietí, a to tým, že maska je vyjadrená počtom bitov. Jej cieľom bolo spomaliť vyčerpanie IPv4 adries. Dĺžka adresy môže byť ľubovoľná, tým pádom nie sú potrebné triedy. Bližšie informácie môžeme nájsť v RFC1519 [23].

Administrative distance

Administrative distance (AD) je v kontexte dynamických smerovacích protokolov preferencia zdroju smerovacích informácií.

Split Horizon

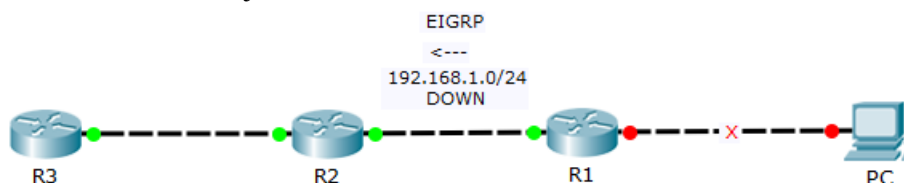
Split horizon je obecné pravidlo implicitně zabraňující posílání počítačové komunikace zpět v směru odkiaľ přišla, vid' obrázok 1.1.



Obrázok 1.1 : Split Horizon

Poison reverse

Poison reverse je obecné pravidlo, ktoré implicitně nutí smerovač oznámit' o nedostupnosti cesty na rozhraní v ceste k cieľu, ako naznačuje obrázok 1.2.



Obrázok 1.2 : Poison Reverse

SNC

Source Node Condition [1] je podmienka pre výber najvhodnejšej cesty k cieľu. Pokiaľ uzol musí zmeniť cestu k cieľu, vyberá medzi susednými uzlami. Vyberá uzol, kde má cesta najnižšiu metriku. Zároveň musí byť nová metrika menšia ako aktuálna metrika v tabuľke. Ak táto podmienka neplatí, smerovač ponechá starú cestu k cieľu. Nemôže sa stať, aby bola metrika zvýšená.

Bellman-Fordov algoritmus

Bellman-Fordov [2] algoritmus sa používa na získanie najkratších ciest k cieľu v orientovanom grafe. Uzly v grafe sú jednotlivé zariadenia a hrany sú linky. Bellman-Fordov algoritmus dokáže pracovať aj so záporným ohodnotením hrán. Každý uzol si pre cieľ ukladá odhad vzdialenosti a susedný uzol v smere k cieľu. Postupne sa odhad vzdialenosti znižuje, až sa dôjde k cieľu.

Ak označíme množinu uzlov V a množinu hrán E , potom sa tento algoritmus dá opísať kódom z obrázku 1.3. Každý uzol si pre cieľ S udržuje odhad vzdialenosti k cieľu S , označený ako $D[A]$, a zvoleného následníka značeného ako $NH[A]$. Hodnota $D[A]$ je vždy horným odhadom najkratšej cesty k cieľu.

```

    BELLMAN-FORD ((V, E), C, S)
1  for každý  $v \in V$  do
2       $D[v] \leftarrow \infty$ 
3       $NH[v] \leftarrow NIL$ 
4   $D[S] \leftarrow 0$ 
5  for  $i \leftarrow 1$  to  $n-1$  do
6      for každá hrana  $(u,v) \in E$  do
7          if  $D[v] > D[u] + C(u,v)$  then
8               $D[v] \leftarrow D[u] + C(u,v)$  then
9                   $NH[v] \leftarrow u$ 
10 for každá hrana  $(u,v) \in E$  do
11     if  $D[v] > D[u] + C(u,v)$  then
12         return FALSE
13 return TRUE

```

Obrázok 1.3 : Bellman-Ford algoritmus

Z pohľadu daného uzlu potom vyzerá nasledovne. Najprv sa nastaví hodnota $D[A]$ na nekonečno a $NH[A]$ na nedefinovanú hodnotu. Hodnota $D[S]$ sa nastaví na nulu. Každý uzol B periodicky posiela všetkým susedom informácie o jemu dostupných cestách obsahujúcich hodnotu $D[B]$. Po prijatí takejto správy uzlom A sa najprv overí, či je sused B zvolený nasledovník pre daný cieľ. Ak sa podmienka splní, ide o aktualizáciu cesty a hodnota $D[A]$ je nastavená na $C(A, B) + D[B]$. Inak sa porovná aktuálna hodnota $D[A]$ s hodnotou $C(A, B) + D[B]$. Ak je oznamovaná hodnota nižšia, ide o oznámenie lepšej cesty, ako je aktuálna. Hodnota $NH[A]$ je nastavená na B , a $D[A]$ na hodnotu $C(A, B) + D[B]$.

Algoritmus využívajú napríklad protokoly RIP a Babel.

Diffusing Update Algorithm

DUAL (Diffusing Update Algorithm) [1, strana 8] slúži pre výber najlepších ciest a odstránenie cyklov v topológií. Bol vyvinutý v SRI International¹. Používa difúzne výpočty a reaguje na všetky možné udalosti pri výpočtoch. Sieť spracúva ako neorientovaný graf, kde hrany sú linky a uzly jednotlivé smerovače. Výmena potrebných informácií medzi uzlami je riadená správami *query* a *reply*. Výsledkom používania DUAL je rýchla konvergencia siete. Tento algoritmus používa protokol EIGRP.

Pre šírenie informácií sú potrebné distribuované smerovacie algoritmy, ako aj koordinácia informácií medzi všetkými uzlami v sieti. DUAL namiesto vektorovej vzdialenosti, akú používa Bellman-Ford, používa prístup k šíreniu smerovacích protokolov pomocou spätnej väzby známej ako difúzny výpočet. Výpočet rastie zahŕňaním uzlov, ktoré sú ovplyvnené zmenou v topológií. Zmenšuje sa vylučovaním tých, ktoré ovplyvnené nie sú. Takýto stav umožňuje dynamický výpočet a jeho rýchle ukončenie.

Algoritmus je popísaný konečným stavovým automatom, ktorý je uvedený v literatúre RFC 7868 [1, strana 10].

¹ Stanford Research Institute International

2 Distance Vector protokoly

Smerovacie protokoly slúžia na preposielanie smerovacích informácií medzi smerovacími zariadeniami. Distance Vector protokoly patria k dynamickým smerovacím protokolom. Pri dynamických smerovacích protokoloch sa na rozdiel od statických nemusia vkladať cesty ručne, a nemusia sa meniť pri zmene topológie. Aby smerovače mohli šíriť informácie, potrebujú poznať topológiu siete, v ktorej sa nachádzajú. Protokoly zo skupiny Distance Vector nepoznajú štruktúru siete za svojimi susedmi. Pre rozhodovanie o smerovaní správ používajú vzdialenosť a smer k cieľu.

2.1 Enhanced Interior Gateway Routing Protocol

Enhanced Interior Gateway Routing Protocol, ďalej len **EIGRP** [4] je Cisco² proprietárny smerovací protokol patriaci do skupiny Internet Gateway Protocol, ktorá je určená pre šírenie smerovacích informácií vo vnútri autonómneho systému. Vznikol ako náhrada staršieho IGRP [5] protokolu. Podporujú ho však len zariadenia značky Cisco.

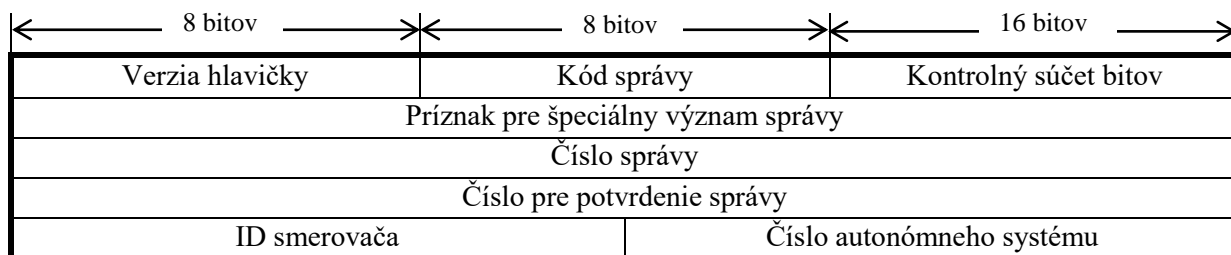
EIGRP patrí k distance vector smerovacím protokolom. K smerovaniu používa vzdialenosť a smer k cieľu. Napriek tomu, že patrí k distance vector protokolom, má aj vlastnosti link-state protokolov. Informácie o smerovaní posielajú len pri zmene v topológiách a neposielajú celú tabuľku, ale len zmenené dáta. Keďže má vlastnosti z oboch skupín smerovacích protokolov, nazýva sa Advanced Distance Vector protokol. EIGRP je classless protokol, prenáša s informáciami aj masku siete. Napriek tomu ho je možné nakonfigurovať ako classfull protokol. Podporuje VLSM aj CIDR.

Pracuje na sieťovej vrstve a podporuje IPv4 aj IPv6, IPX [24] a AppleTalk [24]. V EIGRP je možné použiť autentizáciu aj sumarizáciu ciest.

EIGRP používa podmienku Source Node Condition (SNC) pre výber najlepšej cesty. Pre obmedzenie cyklov kombinuje Split horizon a Poison reverse. Pre výber najlepších ciest používa protokol DUAL.

2.1.1 EIGRP správy

EIGRP používa pre šírenie informácií, ale aj pre vlastnú réžiu, správy zapuzdrené do paketov prenášaných ako unicast alebo multicast. Ako multicast adresa sa používa pre IPv4 adresa 224.0.0.10, pre IPv6 adresa ff02::a. Správy sa skladajú z hlavičky a užitočného obsahu podľa správy. Hlavička je u všetkých správ rovnaká. Obsahuje štruktúru z obrázka 2.1 s popisom polí [1].



Obrázok 2.1: Štruktúra hlavičky EIGRP

Užitočný obsah správy [1] sa líši od daného typu správy. Obsahuje položky vo formáte TLV (Type Length Value) [1, kap.A.6], ktorý umožňuje rozšíriť protokol o ďalšie funkcie.

Ďalej sa budeme venovať jednotlivým správam, ktoré EIGRP používa.

² Cisco Systems, Inc.

- **Hello správa** – používa sa na vyhľadanie susedov a udržanie vzťahu susedstva. Je posielaná ako multicast. Nie je potvrdzovaná;
- **Acknowledgement správa** – správa sa používa na potvrdzovanie iných správ. Je posielaná ako unicast. Nie je potvrdzovaná;
- **Query správa** – využívaná algoritmom DUAL pre difúzny výpočet v aktívnom stave. Je posielaná ako multicast. Je potvrdzovaná;
- **Reply správa** – využíva sa na odpovedanie k správe *Query*. Je posielaná ako unicast. Je potvrdzovaná;
- **Update správa** - používa sa na prenos smerovacích informácií. Je posielaná ako unicast, ale aj ako multicast. Je potvrdzovaná;
- **Request** – používa sa ako žiadosť o špecifické informácie. Je posielaná ako unicast, ale aj ako multicast. Nie je potvrdzovaná.

2.1.2 Dátové štruktúry

Protokol EIGRP obsahuje dve hlavné dátové štruktúry. Tabuľku topológie, ktorá obsahuje všetky cesty k cieľom, ktoré zariadenie pozná. Narozdiel od smerovacej tabuľky, ukladá iba cesty nadobudnuté protokolom EIGRP. Všetky záznamy sa po výpočte metriky vložia do smerovacej tabuľky. Ďalšou štruktúrou je tabuľka susedstva. Táto tabuľka obsahuje adresy rozhraní smerovačov, s ktorými je nadviazaný vzťah susedstva.

2.1.3 Vzťah susedstva

Vzťah susedstva je stav dvoch smerovačov, ktoré o sebe vedia všetky potrebné informácie a pravidelne ich kontrolujú. Aby mohlo dôjsť k susedstvu, smerovače musia byť priamo pripojené. Musia mať na rozhraniach, ktorými sú prepojené, nastavené EIGRP. Musia sa zhodovať čísla autonómneho systému oboch smerovačov. Pre IPv4 musia byť adresy rozhraní v rovnakej podsieti. Pre IPv6 sa kontroluje len platná link-local adresa.

Každý smerovač si vedie vlastnú tabuľku susedov pre každý nakonfigurovaný L3 protokol. Je to tabuľka, kde sa nachádzajú všetci priamo pripojení susedia so vzťahom susedstva. Obsahuje tieto údaje [13].

Výpis tabuľky susedstva môžeme vidieť na obrázku 2.2.

- **H** – udáva poradie, akým bol záznam pridaný do tabuľky, začína 0;
- **Address** – IP adresa suseda;
- **Interface** – rozhranie, cez ktoré je sused pripojený;
- **Hold** – časovač, ktorý sa znižuje. Ak sa vynuluje, nastal problém so susedom. Časovač sa resetuje pri každej *hello* správe;
- **Uptime** – čas, kedy bol sused pridaný do vzťahu susedstva;
- **SRTT** – informácia o tom, ako dlho trvá susedovi odpovedať na správu;
- **RTO** – značí, ako dlho čakáme, kým spravíme retransmisiu, ak nedošlo ACK;
- **Q Cnt** – ak nie je 0, značí zahltenie linky;
- **Seq Num** – sekvenčné číslo posledného *query*, *reply* alebo *update* obdržaného od suseda.

H	Address	Interface	Hold(sec)	Uptime	SRTT (ms)	RTO	Q Cnt	Seq Num
1	192.168.1.1	Fa0/1	11	00:06:12	1023	5000	0	7
0	192.168.1.2	Fa0/0	12	00:08:10	815	4968	0	7

Obrázok 2.2: Výpis z tabuľky susedstva v EIGRP

Aby mohol byť záznam daný do tabuľky, musí prebehnúť komunikácia medzi smerovačmi. Najprv prebieha komunikácia pomocou *Hello* správ, následne si smerovače pridajú suseda do tabuľky susedov. Následuje výmena informácií a pridanie nových informácií do tabuľky topológie a smerovacej tabuľky.

Pre udržanie vzťahu susedstva smerovač pravidelne kontroluje stav susedov pomocou *Hello* správy. Pre kontrolu používa EIGRP dva časovače. *Hello* časovač určuje, ako často sa budú posielat' *Hello* správy, to závisí na type linky. Prvotne je nastavený na 60 sekúnd. Druhý časovač - *Hold* časovač určuje, kedy označíme suseda za nedostupného. Každou správou od daného suseda sa časovač nastaví na pôvodnú hodnotu. Prvotne je nastavený na trojnásobok *Hello* časovača.

2.1.4 Metrika

EIGRP používa pre ohodnotenie cesty **metriku** [1], ktorá je 32-bitové číslo. Skladá sa z **K-hodnôt** [6], ktoré nadobúdajú hodnoty od 0 po 255. Hodnota 0 je vylúčená z výpočtov. Jednotlivé K-hodnoty sú.

- **K1** – šírka pásma. Je to statická hodnota daná typom linky;
- **K2** – zaťaženie. Počíta sa dynamicky na základe rýchlosti odosielania paketov a šírky pásma;
- **K3** – oneskorenie. Statická hodnota, vyjadruje čas potrebný na prenesenie jednobitovej správy susedovi;
- **K4,K5** – spoľahlivosť. Dynamická hodnota, určuje pravdepodobnosť odoslania a prijatia správy;
- **K6** – Používa sa pri výpočte rozšírenej metriky. Umožňuje pridať do výpočtu parametre smerodatná odchýlka a spotreba energie;

Predvolené hodnoty K-hodnôt sú uvedené v tabuľke 2.1.

K1	1
K2	0
K3	1
K4	0
K5	0
K6	0

Tabuľka 2.1: Predvolené K-hodnoty

Predvolené hodnoty pre šírku pásma a oneskorenie podľa typu linky, sú uvedené v RFC [1].

Rozpoznávame dva výpočty metriky, klasickú a rozšírenú metriku. Klasická metrika je spätne kompatibilná s 24-bitovou IGRP [1] metriku. Preto sa šírka pásma a oneskorenie násobí 256. Medzi každým výpočtom je potrebné medzivýsledok zaokrúhliť smerom dole. Pri linkách so šírkou pásma 1Gb/s a viac klasická metrika nevyhovuje, preto sa používa rozšírená metrika. Pridávajú sa k nej ďalšie parametre a výsledok je 64-bitový. Výsledná metrika sa preto podelí číslom 128 aby mohla byť uložená do tabuľky. Rozšírená metrika sa použije iba pri pomenovanej konfigurácii namiesto autonómneho systému.

Výpočet klasickej metriky vychádza zo vzorca :

Výpočet pre $K5 = 0$, sa odvíja od nasledujúceho vzorca.

$$metric = K1 * bandwidth + K2 * bandwidth / (256 - load) + K3 * delay$$

Pre nenulové $K5$ sa použije vzorec nižšie.

$$metric = metric * K5 / reliability + K4$$

Výpočet rozšírenej metriky :

Oneskorenie pre linku rýchlejšiu ako 1 Gb/s.

$$delay = delay_{sum} * 10^6$$

Oneskorenie pre linku pomalšiu ako 1Gb/s.

$$delay = (10^7 * 10^6)/min_bandwidth$$

Následne sa vypočítajú položky Thr (throughput), Lat (latency). Nové parametre smerodajná odchýlka a energia sa sčítajú.

$$Thr = (10^7 * 65536)/min_bandwidth$$

$$Lat = (delay * 65536)/10^6$$

$$extend = energy + jitter$$

Z toho sa vypočíta metrika ako :

$$metric_{ext} = K1 * Thr + K2 * Thr / (256 - load) * K3 * Lat + K6 * extend$$

Ak je hodnota K5 odlišná od 0, musí sa metrika upraviť nasledovne.

$$metric = \frac{metric * K5}{reliability + K4}$$

2.1.5 Spravovanie ciest

Cesta je hodnota zložená z cieľovej siete a suseda. Ak existuje viac ciest k danému cieľu, do tabuľky sa vloží len tá s najmenšou AD. Cesty sa líšia podľa toho, či sú z rovnakého autonómneho systému alebo nie. Cesty môžu byť prevzaté z iného autonómneho systému, statické cesty, či cesty z iných smerovacích protokolov.

Každý autonómny systém v EIGRP má vlastnú tabuľku topológie. V tabuľke sa nachádzajú informácie o stave cesty, adresa nasledujúceho zariadenia po ceste, či súčasná metrika.

2.1.6 Reliable Transport Protocol

EIGRP používa pre spoľahlivosť prenosu protokol **RTP** (Reliable Transport Protocol) [12]. RTP slúži aj pre nespoľahlivý prenos dát. Používa Automatic repeat request [25] metódu Stop and Wait [25], ktorá detekuje duplicitné správy. RTP pracuje s údajmi v hlavičke správ, a to sekvenčným číslom a číslom potvrdenia.

Pri nespoľahlivom prenose sa sekvenčné číslo nastaví na 0, ináč sa nastaví na nenulovú hodnotu. Ako potvrdenie sa posiela správa *Acknowledgement*, ktorá by mala obsahovať práve to číslo. Pre zjednodušenie môžu potvrdenie obsahovať aj správy *update*, *query* a *reply*. Pri obdržaní potvrdenia môže smerovač odoslať ďalšiu správu. Ak neobdrží potvrdenie a vyprší časovač *RTO*, správa sa odošle znovu.

2.2 Babel

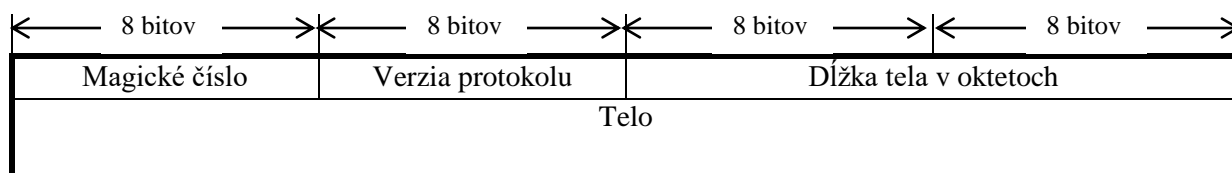
Babel [7] je pôvodne navrhnutý s vlastnosťami protokolov DSDV [2], AODV[2] a EIGRP. Protokol Babel patrí do skupiny distance vektor protokolov. Metriku Babel počíta tak, že každá linka je ohodnotená. Výsledná metrika k cieľu sa rovná sume všetkých liniek po ceste k cieľu. Hlavnou výhodou nasadenia protokolu Babel je jeho rýchla reakcia na nežiaduce stavy v sieti, ako sú slučky. Vo väčšine prípadov vie Babel skonvergovať sieť do stavu bez slučiek v krátkej chvíli. Používa na to Bellman-Fordov algoritmus [2]. Tabuľky susedov si udržiava periodickým zasielaním *hello* paketov, ale aj okamžite po zmene topológie. Pre zabezpečenie pred slučkami kombinuje Split horizon a Poison reverse

Pracuje na sieťovej vrstve a podporuje IPv4 aj IPv6 adresovanie. Protokol nie je podporovaný na zariadeniach od firmy Cisco. Na rozdiel od EIGRP nepoužíva spoľahlivý prenos, ale pravidelné zasielanie správ. To robí Babel neefektívny vo veľkých stabilných sieťach.

2.2.1 Babel správy

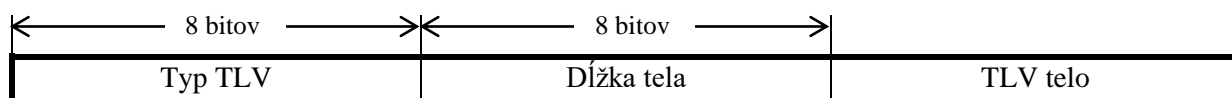
Babel používa pre šírenie informácií pakety UDP [26]. Cieľová adresa správ môže byť unicast alebo multicast. Ako multicastová adresa sa používa adresa 224.0.0.111.

Každá správa obsahuje položky z obrázka 2.3 [2]. Ak prvý oktet nie je 42, paket musí byť ticho ignorovaný.



Obrázok 2.3: Formát správy Babel

Telo správy tvoria sekvencie TLV [1, kap.A.6]. Štruktúra tela správy [2] protokolu Babel je znázornená na obrázku 2.4.



Obrázok 2.4: Štruktúra tela správy

Posielané správy môžu obsahovať žiadosť o potvrdenie. Potvrdzovanie je voliteľné, ale každý smerovač na to musí vedieť reagovať v prípade potreby. Takéto potvrdenie je vždy posielané ako unicast. Následne si ukážeme základné typy TLV protokolu Babel :

- **Hello** – používa sa na objavenie susedov, vždy je posielaná ako multicast;
- **IHU** – používa sa pre potvrdenie obojstrannej dosiahnuteľnosti susedov a udržanie susedstva. Ďalej sa používa pri zisťovaní metriky. Hoci je navrhnutá pre unicast, môže sa posielat' aj multicastovo;
- **Acknowledgement request** – používa sa na vyžiadanie zaslania *acknowledgement*;
- **Acknowledgement** – slúži ako potvrdzovacia správa, vždy je posielaná ako unicast;
- **Router-id** – zavádza ID smerovača, vyplýva z následného *update*;
- **Next hop** – zavádza adresu Next-hop pre danú adresu, vyplýva z následného *update*;
- **Update** – slúži ako aktualizácia smerovacích informácií, môže aj zavádzať nové ID smerovača či predvolený prefix;
- **Route request** – pomocou nej sa dá vyžiadať zaslania *update*;
- **Seqno request** – slúži na vyžiadanie *update* pre daný prefix s daným poradovým číslom.

Pri procese získavania susedstva s iným smerovačom je zisťovaná vzájomná obojsmerná dostupnosť smerovačov. Preposielajú si informácie potrebné k výpočtu metriky. Po získaní susedstva smerovač pravidelne posiela *hello* správy. Tie sa používajú na udržanie stavu susedstva medzi dvoma susedmi.

2.2.2 Dátové štruktúry

- **ID smerovača** – slúži na jednoznačné identifikovanie jednotlivých smerovačov. Pozostáva z 8 bajtového reťazca.
- **Sekvenčné číslo** – vyjadruje, kedy boli naposledy potvrdené informácie. Je vkladané do všetkých informácií o cestách.
- **Tabuľka rozhraní** – obsahuje zoznam všetkých rozhraní nakonfigurovaných s protokolom Babel. Každé rozhranie má dva časovače - *hello* časovač pre zasielanie *hello* správ a *update* časovač, ktorý riadi posielanie aktualizácií ciest.
- **Tabuľka susedov** – obsahuje všetky rozhrania susedov, s ktorými má smerovač nadviazaný vzťah susedstva. Obsahuje rozhranie, cez ktoré sa dá dostať k susedovi, adresu suseda, históriu prijatých *hello* správ, cenu TXCOST a sekvenčné číslo očakávané v nasledujúcej *hello* správe.
- **Tabuľka zdrojov** – slúži na ukladanie vhodných vzdialeností k cieľu.
- **Tabuľka topológie** – obsahuje všetky známe cesty pre daný smerovač.
- **Tabuľka čakajúcich požiadaviek** – obsahuje zoznam požiadaviek na zvýšenie sekvenčného čísla odoslaných smerovačom, na ktoré nebolo odpovedané.

2.2.3 Metrika

Protokol Babel počíta metriku z metriky, ktorú mu pošle sused, a ceny linky k susedovi.

Cena linky [2] je hodnotenie cesty k susednému smerovaču. Počíta sa z hodnoty RXCOST, ktorá je odvodená z prijatých *hello* správ a TXCOST, ktorá je získaná od suseda v správe *IHU*. Cena musí byť pozitívna, a ak nejaký čas neboli prijaté *hello* správy, je nekonečná. Pre výpočet sa používa spôsob K-out-of-j alebo ETX.

K-out-of-j [2] je spôsob výpočtu vhodný pre drôtové spojenia. Smerovač si udržuje históriu j správ. Ak bolo k a viac správ prijatých úspešne, je linka funkčná a RXCOST je nastavená na konštantu C. Ináč je RXCOST nastavená na nekonečno. Konštanta C nadobúda hodnoty $C \geq 1$.

ETX [2] je spôsob vhodný pre bezdrôtové spojenia. Zakladá sa na odhade počtu poslaných správ pre úspešné odoslanie. Zo správ *hello* sa určí konštanta β , ktorá značí pravdepodobnosť úspešného prijmu správy *hello*. Z toho sa určí RXCOST.

$$RXCOST = \frac{256}{\beta}$$

Následné sa určí konštanta α . Značí odhad pravdepodobnosti úspešného odoslania *hello* správy.

$$\alpha = \min\left(1, \frac{256}{TXCOST}\right)$$

Z týchto vzťahov sa vypočíta cena linky

$$cost = \frac{256}{\alpha * \beta}$$

Ak už má smerovač hodnotu ceny linky, môže vypočítať metriku k cieľu, a to následne: Ak je cena linky nekonečno, celá metrika musí byť nekonečno. Metrika musí byť vždy väčšia ako cena linky. Metrika sa dá definovať ako suma všetkých liniek, ktoré je potrebné prekonať na ceste k cieľu.

Protokol Babel používa podmienku vhodnosti SNC, ktorá sa podobá podmienke z EIGRP tak, že nikdy nemôže byť prijatá metrika k cieľu, ktorá je vyššia ako aktuálna metrika. Týmto pravidlom sa predíde k vzniku slučky.

2.2.4 Podmienky vhodnosti

Tento protokol rieši aj problémy počítania do nekonečna a problém vyhľadovania [2] tak, že ak oznámenie o ceste nesplňuje podmienku vhodnosti, je ignorovaná. Používa na to podmienku SNC takisto ako aj protokol EIGRP. Metrika cesty sa nikdy nemôže zvýšiť, ale iba znížiť oproti pôvodnej hodnote.

Pri použití SNC môže nastať problém vyhľadovania. Tento problém nastáva, ak sú vyčerpané všetky vhodné cesty k cieľu, a jediná dostupná cesta nesplňuje podmienku vhodnosti. V protokole EIGRP sa tento problém rieši nastavením všetkých uzlov do stavu aktív [1]. Babel však využíva menej náročnú metódu sekvenčných čísel [2, strana 7].

2.3 Routing Information Protokol

RIP (Routing Information Protokol) [14] slúži na preposielanie informácií o sieti. Patrí k Distance Vector protokolom. Je to jeden z najpoužívanejších smerovacích protokolov. Pôvodne bol navrhnutý ako smerovací protokol v sieti univerzity v Berkeley.

Pre výpočet metriky používa Belmann-Fordov algoritmus, podobne ako protokol Babel. Delí účastníkov komunikácie na pasívnych a aktívnych. Aktívni účastníci posielajú informácie, zatiaľ čo pasívni len aktualizujú svoje tabuľky. Počítače v sieti nemôžu byť aktívne. Metrika sa skladá z počtu zariadení po ceste k cieľu. Aktualizovaná informácia nemôže mať horšiu hodnotu, akú mala dovtedy. To znamená, že ak dostane informáciu o ceste k cieľu s vyššou metrikou, v tabuľkách ponechá starú hodnotu. Metrika sa skladá z dvoch častí - IP adresy cieľa a počtu skokov. Využíva aj Poison reverse a Split horizon.

Protokol RIP je známy v troch verziách. **RIPv1** [3] je classfull, to znamená, že v aktualizáciách neposiela sieťovú masku. Avšak verzia **RIPv2** [15] je classless. Pre IPv6 slúži protokol vo verzii **RIPng** [16].

Konvergenca po zmene je oproti ostatným protokolom oveľa pomalšia.

2.3.1 RIP správy

Protokol používa pre posielanie smerovacích informácií pakety založené na UDP, na port 520. RIP môže využívať takzvaný „**silence**“ mód, ak router chce počúvať komunikáciu, ale nechce mať funkciu brány.

Aktualizácie posiela svojim susedom primárne každých 30 sekúnd pomocou broadcastu. Pri posielaní aktualizácie sa posiela celá tabuľka. V jednej správe môže byť uvedené maximálne 25 sietí. Ak je potreba odoslať informácie o viac než 25 sieťach, posiela sa viacero správ.

Protokol používa hlavne následné správy, ostatné správy nájdeme v RFC [3, strana 18] :

- **Request** – žiadosť, aby mu systém poslal aktuálnu tabuľku;
- **Response** – odpoveď na správu `request`, obsahuje aktuálnu smerovaciu tabuľku.

2.3.2 Dátové štruktúry

RIP je používaný pre svoju jednoduchosť, to sa týka aj dátových štruktúr, ktoré udržiava. Informácie o vzťahoch sú ukladané do smerovacej tabuľky. V tejto tabuľke si uchováva potrebné informácie o dostupných cestách k susedom a metriku ciest.

V protokole sa používajú aj časovače:

- **Update** – používa sa pre periodické zasielanie aktualizácií, všeobecne je nastavený na 30 sekúnd;
- **Timeout** – čas, po ktorom je daná cesta bez odozvy označená za nedostupnú;
- **Flush** – čas, po ktorom sa nedostupná cesta vymaže celkovo.

2.3.3 Metrika

Metrika je v protokole RIP vypočítaná ako počet skokov, ktoré musí paket vykonať na ceste k cieľu. Skok je počet zariadení, cez ktoré musí paket prejsť. Pri metrike sa predpokladá, že bude kladná.

Pre zabezpečenie pred slučkami je aplikovaný algoritmus split horizon. Maximálny počet hopov je 15. Ak je stanica nedosiahnuteľná, je metrika nastavená na 16, čo značí nekonečno. Obmedzené je aj počítanie do nekonečna, a to tak, že pokiaľ príde aktualizácia s vyššou metrikou, v tabuľke ostane pôvodná hodnota.

3 Prostredie OMNeT++

OMNeT++ [9] je objektovo orientované diskkrétne simulačné prostredie so širokou škálou využitia. Výhodou je aj otvorená architektúra a prepracovaná podpora pre grafické rozhranie, či vyhodnocovanie simulácií. Môže sa využívať pre modelovanie drôtových, ale aj bezdrôtových sietí a sledovanie šírenia informácií v nich. Zahrňuje však aj modelovanie rôznych sieťových protokolov, či multiprocessorové architektúry.

Momentálne je pre prostredie OMNeT++ dostupných mnoho projektov, ktoré sú prístupné pod open-source licenciou. Medzi najpoužívanejšie dostupné knižnice patrí INET Framework [8] a Mobility Framework [27]. Venujú sa simuláciám sietí s protokolmi TCP/UDP, IP a taktiež 802.11, Ethernet, IPv6, či pre nás dôležité RIP.

OMNeT++ je diskkrétne prostredie. Diskkrétne znamenie znamená, že zmeny v systéme prebiehajú skokovo, tak, že netrvajú žiadnu dobu. Simulátor používa dátovú štruktúru - kalendár udalostí. Udalosť je daná aktivačným časom a prioritou. Aktivačný čas je čas, kedy sa má udalosť vybrať z kalendára a vykonať sa. Priorita zase určuje, ktorá udalosť sa vykoná, ak majú viaceré udalosti rovnaký aktivačný čas. Simulačný čas je časová os simulácie. Nemusí byť synchronná s reálnym časom. Systém sa mení len vykonaním udalosti.

Začiatkom každej simulácie je vytvorenie kalendára udalostí a vloženie udalostí. Postupne sú udalosti podľa aktivačného času vyberané z kalendára a sú vykonávané. Po vybratí udalosti z kalendára je zmenený simulačný čas na aktivačný čas udalosti. OMNeT++ pracuje s modulmi, ktoré medzi sebou komunikujú. Vykonanie udalosti je zaslanie správy modulom inému modulu. Ten správu spracuje podľa vopred definovaných inštrukcií. Simulácia končí, ak sa v kalendári nenachádzajú žiadne udalosti, alebo simulačný čas sa rovná časovej zarážke na osi modelu.

3.1 Moduly

Simulačné moduly sú tvorené do seba vnorenými modulmi. Moduly medzi sebou komunikujú zasielaním správ. Jednotlivé inštancie modulov sú nazývané sieť. Hlavný modul nachádzajúci sa najvyššie v hierarchii, sa nazýva systémový modul. Môže obsahovať rôzne ďalšie submoduly. Tie môžu obsahovať ďalšie submoduly alebo jednoduché moduly. Jednoduchý modul už neobsahuje žiadne ďalšie moduly. Štruktúra modulov je popísaná jazykom NED.

3.2 Jazyk NED

Jazyk NED slúži na popis topológie v simuláciách. Umožňuje modulárny popis siete, čo znamená, že sieť môže byť zložená z viacerých častí. Tieto sa potom spoja dokopy. Tvoria ich moduly a kanály pre komunikáciu. Jednotlivé modulárne časti sa môžu využiť aj pre iné zapojenia. Súbory definujúce popis siete pomocou jazyka NED majú príponu .ned. Môžu byť preložené do C++ jazyka a následne skompilované v spustiteľné súbory.

3.3 INET

INET [8] framework je open source balík. Bol vyvinutý v rokoch 2000-2001 na univerzite v Karlsruhe. Neskôr, po malom úpadku, sa ho ujal Andras Varga, ktorý mu pridal mnoho implementácií. Následne bol INET postupne doplňovaný o ďalšie protokoly študentmi univerzít po celom svete.

Architektúra vychádza z OMNeT++. Kombinuje moduly tak, aby vznikli sieťové zariadenia. Súčasťou frameworku sú hotové modely sieťových zariadení a niektorých protokolov. Na popis modulu sa využíva jazyk NED. Funkcionalita jednotlivých modulov sa popisuje pomocou jazyka C++. Každá správa prechádza jednotlivými vrstvami TCP/IP modelu.

Obsahuje implementáciu protokolov Ethernet, PPP, IPv4, IPv6, TCP, UDP či RIP, OSPF, OSPFv2 a iné.

3.4 ANSAINET

Projekt ANSA (Automated Network Simulation and Analysis) [17] vznikol na Fakulte informačných technológií Vysokého učenia technického v Brne. Rieši ho výskumná skupina NES@FIT. Hlavnou náplňou skupiny sú automatizované metódy vytvárania simulačných modelov, formálna analýza a verifikácia sietí na základe znalosti topológie. V rámci tohto výskumu vznikol aj balíček rozšírenia frameworku INET, ANSAINET [11]. Balíček rozširuje INET o protokoly HSRP, VRRRv2, GLBP, IS-IS CDP, LLDP, STP, LISP, ale hlavne smerovacie protokoly RIPv2, RIPng, EIGRP [10] či Babel.

4 Porovnanie jednotlivých protokolov

V tejto kapitole sú porovnané jednotlivé protokoly v simulátore OMNeT++ voči správaniu sa protokolov v reálnych zariadeniach. Pre testovanie reálnych zariadení bolo použité virtuálne prostredie UNetLab [28]. Protokoly EIGRP, RIPv1 a RIPv2 boli testované na zariadeniach I86BI_LINUX-Adventerprise-M vo verzii 15.4. Pre Babel bolo použité zariadenie Linux Ubuntu 17.04 s verziou protokolu babeld 1.7.0. Na tomto zariadení bolo nutné najprv Babel nakonfigurovať a virtuálny stroj exportovať do prostredia UNetLab.

Postupnosť komunikácie medzi zariadeniami a časové značky v reálnej topológii siete boli zisťované z ladiacich výpisov pomocou príkazov uvedených v tabuľke 4.1. Komunikácia bola odchytená pomocou programu Wireshark.

Protokol	Príkaz
EIGRP	debug eigrp packets
RIPv1	debug ip rip
RIPv2	debug ip rip
Babel	babeld -d [1-3]

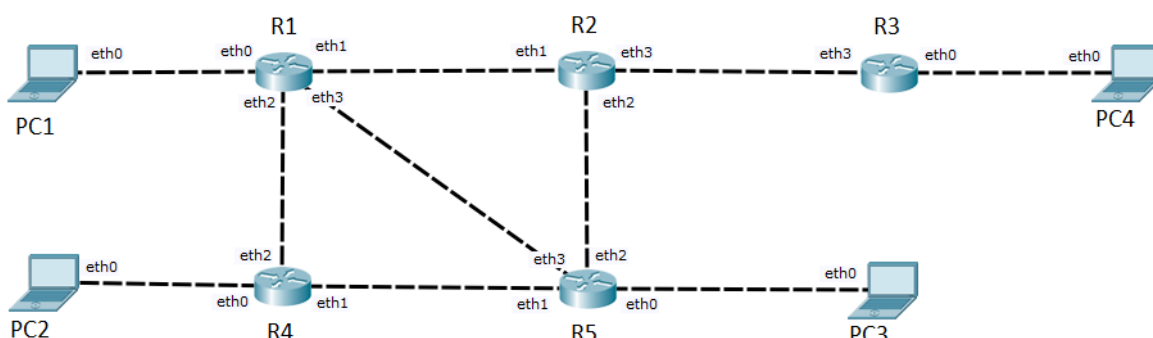
Tabuľka 4.1 : Príkazy pre zobrazenie ladiacich výpisov

Pri porovnaní reálneho zapojenia a simulácie v OMNeT++ je treba brať do úvahy vlastnosti simulátorov. Základnou vlastnosťou je nulový čas spracovania udalostí. Jediné oneskorenie je spôsobené prenosom správ, čo je v reálnej topológii nereálne dosiahnuť. Niektoré časové rozdiely sú pri simulácii menšie ako jedna tisícina sekundy, preto boli takéto rozdiely zaokrúhlené na jednu tisícinu. Čas T_0 značí začiatok merania. Popis významu počiatočného času je popísaný pri každom meraní.

V nasledujúcich porovnaníach sú uvedené výstupy z jedného zariadenia, pretože výpisy všetkých zariadení by boli zdlhové a zahlcujúce. Všetky výpisy sú uvedené v prílohe A.

4.1 Topológia porovnáwanej siete

Pre meranie rozdielov smerovacích protokolov bola použitá topológia zobrazená na obrázku 4.1. V topológii sa nachádza päť smerovacích zariadení a štyri koncové zariadenia. Koncové zariadenia môžu byť nahradené sieťou o veľkosti sieťovej masky /24. IP adresy priradené k jednotlivým rozhraniám sú uvedené v tabuľke 4.2.



Obrázok 4.1 : Topológia meranej siete

Zariadenie	Rozhranie	Adresa IPv4
PC1	Eth0	192.168.1.1
PC2	Eth0	192.168.2.1
PC3	Eth0	192.168.3.1
PC4	Eth0	192.168.4.1
R1	Eth0	192.168.1.2
	Eth1	192.168.5.9
	Eth2	192.168.5.1
	Eth3	192.168.5.5
R2	Eth1	192.168.5.10
	Eth2	192.168.5.13
	Eth3	192.168.5.17
R3	Eth0	192.168.4.2
	Eth3	192.168.5.18
R4	Eth0	192.168.2.2
	Eth1	192.168.5.21
	Eth2	192.168.5.2
R5	Eth0	192.168.3.2
	Eth1	192.168.5.22
	Eth2	192.168.5.14
	Eth3	192.168.5.6

Tabuľka 4.2 : Pridelenie IPv4 adres zariadeniam v topológii

4.2 Test ustanovenia stavu susedstva

V tomto teste bol sledovaný proces, v ktorom zariadenie nájde nového suseda a ustanoví s ním stav susedstva. V tomto stave medzi sebou začnú zdieľať smerovacie informácie. Pre testovanie bola použitá topológia z obrázku 4.1. Pri testovaní v simulátore boli všetky zariadenia zapnuté v rovnakom momente. To je pri reálnych zariadeniach ťažko dosiahnuteľné. Z toho dôvodu bol použitý skript, ktorý v jednom momente nastavil všetky rozhrania do stavu up. Zaznamenané informácie boli sledované medzi zariadeniami R1 a R2. Počas pozorovania výsledkov je treba brať do úvahy vlastnosti simulácie, ale aj zaťaženie reálnych zariadení. Preto sú časové odstupy správ odlišné.

Údaje získané z protokolu EIGRP sú uvedené v tabuľke 4.3.

P.č.	Typ správy	Smer správy	Čas reálny [s]	Čas simulácie [s]
1	Hello	R2 → R1	1,006	0,014
2	Hello	R1 → R2	1,010	0,020
3	Hello	R2 → R1	1,011	0,026
4	Update (INIT)	R2 → R1	1,019	0,039
5	Update (INIT)	R1 → R2	3,018	0,040
6	Update	R2 → R1	3,022	0,060
7	Ack	R1 → R2	3,031	0,077
8	Update	R1 → R2	3,039	0,061
9	Ack	R2 → R1	3,044	0,076
10	Update	R1 → R2	3,053	0,083
11	Ack	R2 → R1	3,054	0,099
12	Hello	R2 → R1	8,028	4,001
13	Hello	R1 → R2	8,061	4,001

Tabuľka 4.3 : Čas prenosu správ pri nadviazaní nového susedstva pre protokol EIGRP

Priebeh nadviazania a udržanie susedstva medzi dvoma zariadeniami začína zaslaním správ Hello. Následne sa posielajú správy Update s príznakom INIT. Tieto správy slúžia ako žiadosť o zaslanie smerovacích informácií od susedného zariadenia. Odpoveď na to je zaslanie správy Update so svojimi smerovacími údajmi. Na správy zariadenia odpovedajú správou Ack, ktorá slúži ako potvrdenie o prijatí správy. Po vymenení všetkých smerovacích informácií odosiľajú zariadenia správy Hello v danom časovom intervale pre udržanie stavu susedstva.

Čas T_0 je čas, kedy je na zariadeniach zapnutý proces EIGRP. Resp. čas, kedy sú rozhrania uvedené do stavu up. Ako môžeme vidieť, v reálnom zapojení je prvá Hello správa odoslaná po určitom čase. Tento jav je spôsobený zavedením procesu EIGRP na zariadeniach. V simulátore je tento čas náhodne zvolený v rozmedzí 0-1 sekunda.

EIGRP-IPv4 Neighbors for AS(1)							
H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RTO	Q Cnt Seq Num
2	192.168.5.10	Et0/1	14	00:07:22	9	100	0 12
1	192.168.5.6	Et0/3	13	00:19:31	818	4908	0 11
0	192.168.5.2	Et0/2	10	00:19:33	1	100	0 12

(a.)R1 – reálne zapojenie

```

elements[3] (inet::EigrpNeighbor *)
├── [0] = ID:1 Address:192.168.5.10 IF:eth1(102) HoldInt:15 SeqNum:13
├── [1] = ID:2 Address:192.168.5.2 IF:eth2(103) HoldInt:15 SeqNum:9
└── [2] = ID:3 Address:192.168.5.6 IF:eth3(104) HoldInt:15 SeqNum:16
  
```

(b.)R1 – zapojenie v OMNeT++

Obrázok 4.2 : Tabuľka susedstva pre zariadenie R1 protokolu EIGRP

```

192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.1.0/24 is directly connected, Ethernet0/0
L    192.168.1.2/32 is directly connected, Ethernet0/0
D    192.168.2.0/24 [90/307200] via 192.168.5.2, 00:04:25, Ethernet0/2
D    192.168.3.0/24 [90/307200] via 192.168.5.6, 00:04:23, Ethernet0/3
192.168.5.0/24 is variably subnetted, 8 subnets, 2 masks
C    192.168.5.0/30 is directly connected, Ethernet0/2
L    192.168.5.1/32 is directly connected, Ethernet0/2
C    192.168.5.4/30 is directly connected, Ethernet0/3
L    192.168.5.5/32 is directly connected, Ethernet0/3
C    192.168.5.8/30 is directly connected, Ethernet0/1
L    192.168.5.9/32 is directly connected, Ethernet0/1
D    192.168.5.12/30 [90/307200] via 192.168.5.6, 00:04:23, Ethernet0/3
D    192.168.5.20/30 [90/307200] via 192.168.5.6, 00:04:25, Ethernet0/3
      [90/307200] via 192.168.5.2, 00:04:25, Ethernet0/2

```

(a.)R1 – reálne zapojenie

```

elements[13] (inet::IPv4Route *)
- [0] = C 192.168.5.0/30 is directly connected, eth2
- [1] = C 192.168.5.4/30 is directly connected, eth3
- [2] = C 192.168.5.8/30 is directly connected, eth1
- [3] = D 192.168.5.12/30 [90/30720] via 192.168.5.10, eth1
- [4] = D 192.168.5.12/30 [90/30720] via 192.168.5.6, eth3
- [5] = D 192.168.5.16/30 [90/30720] via 192.168.5.10, eth1
- [6] = D 192.168.5.20/30 [90/30720] via 192.168.5.2, eth2
- [7] = D 192.168.5.20/30 [90/30720] via 192.168.5.6, eth3
- [8] = C 192.168.1.0/24 is directly connected, eth0
- [9] = D 192.168.2.0/24 [90/30720] via 192.168.5.2, eth2
- [10] = D 192.168.3.0/24 [90/30720] via 192.168.5.6, eth3
- [11] = D 192.168.4.0/24 [90/33280] via 192.168.5.10, eth1
- [12] = C 127.0.0.0/8 is directly connected, lo0

```

(b.)R1 – zapojenie v OMNeT++

Obrázok 4.2 : Smerovacia tabuľka pre zariadenie R1 protokolu EIGRP

Na obrázku 4.2 sú zobrazené tabuľky susedov pre zariadenie R1. Tabuľka topológie pre protokol EIGRP je zobrazená na obrázku 4.3. Tabuľky boli získané po ukončení výmen smerovacích informácií na všetkých zariadeniach.

Údaje získané z protokolu Babel sú uvedené v tabuľke 4.4. Podmienky získania informácií boli rovnaké ako pri protokole EIGRP. Zapnutie procesu `babeld` na všetkých zariadeniach naraz bolo veľmi ťažké dosiahnuť. Preto správy v porovnaní so simulátorom môžu byť v odlišnom poradí.

Čas T_0 je moment, v ktorom bol zapnutý proces `babeld` na všetkých zariadeniach. V simulátore to znamenalo zapnutie všetkých zariadení. Implementácia protokolu v simulátore nepodporuje postupy pre zvýšenie spoľahlivosti pri prenose na stratových linkách. Z tohto dôvodu sú správy, ktoré sú posielané viacnásobne, ignorované pri porovnaní.

P.č.	Typ správy	Smer správy	Čas reálny [s]	Čas simulácie [s]
1	Hello, Route Request	R2 → R1	0,238	0,029
2	Hello, IHU, Update	R1 → R2	0,267	0,229
3	Hello, Route Request	R1 → R2	3,771	0,029
4	Hello, IHU, Update	R2 → R1	3,818	0,229
5	Hello, IHU	R2 → R1	3,819	0,429
6	Hello, IHU	R1 → R2	3,842	0,429
7	Route request	R1 → R2	3,845	0,429
8	Route request	R2 → R1	4,641	0,429
9	Update	R1 → R2	6,020	5,889
10	Hello, IHU	R2 → R1	7,898	25,089
11	Update	R2 → R1	11,730	4,529
12	Hello	R1 → R2	11,861	24,159
13	Hello, IHU	R2 → R1	15,590	65,009
14	Hello	R2 → R1	17,400	44,639

Tabuľka 4.4 : Čas prenosu správ pri nadviazaní nového susedstva pre protokol Babel

Pri nadviazaní stavu susedstva posiela zariadenie správu Hello so žiadosťou o zaslanie smerovacích údajov. Prijemca tejto správy na to odpovedá ohlásením sa a zaslaním jemu dostupných ciest. V tomto prípade sú obe zariadenia zapnuté naraz a preto obe žiadajú o zaslanie informácií. Z tohto pohľadu je to pre zariadenie, ktoré žiada o informácie, prvá obdržaná Hello správa. Preto odpovedá správou Hello IHU. V tomto bode sa linka stane dostupnou. Nasleduje žiadosť o zaslanie všetkých dostupných ciest. Na rozdiel od predošlej správy so žiadosťou, je táto adresovaná priamo konkrétnemu zariadeniu na linke. Odpoveďou je správa Hello IHU. Zariadenie posiela správu Hello po vypršaní časovača.

Tabuľka susedstva a smerovacia tabuľka je zobrazená na obrázku 4.4 resp. 4.5.

```

192.168.5.6 dev ens6(eth3) lladdr 50:06:00:05:00:03 STALE
192.168.5.2 dev ens5(eth2) lladdr 50:06:00:04:00:02 STALE
192.168.5.10 dev ens4(eth1) lladdr 50:06:00:03:00:01 STALE

```

(a.) R1 – reálne zapojenie

```

elements[3] (inet::BabelNeighbour *)
├── [0] = 192.168.5.10 on eth1 H:1111110000000000 cost:96 txc:96 rxc:96 eHsn:58554 Hint:2000 lint:6000
├── [1] = 192.168.5.2 on eth2 H:1111110000000000 cost:96 txc:96 rxc:96 eHsn:2568 Hint:2000 lint:6000
└── [2] = 192.168.5.6 on eth3 H:1111110000000000 cost:96 txc:96 rxc:96 eHsn:60984 Hint:2000 lint:6000

```

(b.) R1 – zapojenie v OMNeT++

Obrázok 4.3 : Tabuľka susedstva pre zariadenie R1 protokolu Babel

```

Kernel IP routing table
Destination      Gateway         Genmask        U         Iface
192.168.1.0      0.0.0.0        255.255.255.0 U         ens3(eth0)
192.168.3.2      192.168.5.6   255.255.255.255 UGH      ens6(eth3)
192.168.5.0      0.0.0.0        255.255.255.252 U         ens5(eth2)
192.168.5.4      0.0.0.0        255.255.255.252 U         ens6(eth3)
192.168.5.6      192.168.5.6   255.255.255.255 UGH      ens6(eth3)
192.168.5.8      0.0.0.0        255.255.255.252 U         ens4(eth1)
192.168.5.10     192.168.5.6   255.255.255.255 UGH      ens6(eth3)
192.168.5.13     192.168.5.6   255.255.255.255 UGH      ens6(eth3)
192.168.5.14     192.168.5.6   255.255.255.255 UGH      ens6(eth3)
192.168.5.17     192.168.5.6   255.255.255.255 UGH      ens6(eth3)
192.168.5.22     192.168.5.6   255.255.255.255 UGH      ens6(eth3)

```

(a.) R1 – reálne zapojenie

```

elements[11] (inet::IPv4Route *)
- [0] = C 192.168.5.0/30 is directly connected, eth2
- [1] = C 192.168.5.4/30 is directly connected, eth3
- [2] = C 192.168.5.8/30 is directly connected, eth1
- [3] = ba 192.168.5.12/30 [125/96] via 192.168.5.10, eth1
- [4] = ba 192.168.5.16/30 [125/96] via 192.168.5.10, eth1
- [5] = ba 192.168.5.20/30 [125/96] via 192.168.5.2, eth2
- [6] = C 192.168.1.0/24 is directly connected, eth0
- [7] = ba 192.168.2.0/24 [125/96] via 192.168.5.2, eth2
- [8] = ba 192.168.3.0/24 [125/96] via 192.168.5.6, eth3
- [9] = ba 192.168.4.0/24 [125/192] via 192.168.5.10, eth1
- [10] = C 127.0.0.0/8 is directly connected, lo0

```

(b.) R1 – zapojenie v OMNeT++

Obrázok 4.4 : Smerovacia tabuľka pre zariadenie R1 protokolu Babel

Ďalší z porovnávaných protokolov bol RIP v oboch verziách dostupných pre IPv4 smerovanie. Údaje získané z komunikácie zariadení sú uvedené v tabuľke 4.5 pre verziu RIPv1, a v tabuľke 4.6 pre RIPv2.

Čas T_0 bol opäť čas, kedy boli spustené všetky zariadenia v simulátore, a všetky rozhrania nastavené na stav up v reálnom zapojení.

P.č.	Typ správy	Smer správy	Čas reálny [s]	Čas simulácie [s]
1	Request	R2 → R1	1,004	2,964
2	Response	R1 → R2	1,005	2,965
3	Response	R1 → R2	27,830	7,176
4	Response	R2 → R1	27,832	6,143

Tabuľka 4.5 : Čas prenosu správ pri nadviazaní nového susedstva pre protokol RIPv1

P.č.	Typ správy	Smer správy	Čas reálny [s]	Čas simulácie [s]
1	Request	R2 → R1	1,005	2,744
2	Response	R1 → R2	4,757	2,963
3	Response	R2 → R1	23,714	4,223
4	Response	R1 → R2	28,371	6,270

Tabuľka 4.6 : Čas prenosu správ pri nadviazaní nového susedstva pre protokol RIPv2

Protokol RIP má jednoduchú skladbu zasielania správ. Router po pripojení zasiela správu Request susedovi. Očakáva odpoveď Response, ktorá nesie smerovacie informácie suseda. Po uplynutí časovača nasleduje zaslanie správy Response s informáciami. RIP ako jediný zo sledovaných protokolov posielala informácie bez ohľadu na zmenu v topológii.

Na obrázku 4.5 je tabuľka susedstva a na obrázku 4.6 je uvedená smerovacia tabuľka pre protokol RIP.

Device ID	Local Infrfce	Holdtme	Capability	Platform	Port ID
R2	Eth 0/1	163	R B	Linux Uni	Eth 0/1
R4	Eth 0/2	121	R B	Linux Uni	Eth 0/2
R5	Eth 0/3	135	R B	Linux Uni	Eth 0/3

(a.) R1 – reálne zapojenie

```

elements[3] (inet::EigrpNeighbor *)
├── [0] = ID:1 Address:192.168.5.10 IF:eth1(102) HoldInt:15 SeqNum:13
├── [1] = ID:2 Address:192.168.5.2 IF:eth2(103) HoldInt:15 SeqNum:9
└── [2] = ID:3 Address:192.168.5.6 IF:eth3(104) HoldInt:15 SeqNum:16
    
```

(b.) R1 – zapojenie v OMNeT++

Obrázok 4.5 : Tabuľka susedstva pre zariadenie R1 protokolu RIP

```

192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.1.0/24 is directly connected, Ethernet0/0
L    192.168.1.2/32 is directly connected, Ethernet0/0
R    192.168.2.0/24 [120/1] via 192.168.5.2, 00:00:19, Ethernet0/2
R    192.168.3.0/24 [120/1] via 192.168.5.6, 00:00:04, Ethernet0/3
R    192.168.4.0/24 [120/2] via 192.168.5.10, 00:00:26, Ethernet0/1
192.168.5.0/24 is variably subnetted, 9 subnets, 2 masks
C    192.168.5.0/30 is directly connected, Ethernet0/2
L    192.168.5.1/32 is directly connected, Ethernet0/2
C    192.168.5.4/30 is directly connected, Ethernet0/3
L    192.168.5.5/32 is directly connected, Ethernet0/3
C    192.168.5.8/30 is directly connected, Ethernet0/1
L    192.168.5.9/32 is directly connected, Ethernet0/1
R    192.168.5.12/30 [120/1] via 192.168.5.10, 00:00:26, Ethernet0/1
      [120/1] via 192.168.5.6, 00:00:04, Ethernet0/3
R    192.168.5.16/30 [120/1] via 192.168.5.10, 00:00:26, Ethernet0/1
R    192.168.5.20/30 [120/1] via 192.168.5.6, 00:00:04, Ethernet0/3
      [120/1] via 192.168.5.2, 00:00:19, Ethernet0/2

```

(a.) R1 – reálne zapojenie

```

elements[11] (inet::IPv4Route *)
- [0] = C 192.168.5.0/30 is directly connected, eth1
- [1] = C 192.168.5.4/30 is directly connected, eth2
- [2] = C 192.168.5.8/30 is directly connected, eth3
- [3] = R 192.168.5.12/30 [255/2] via 192.168.5.2, eth1
- [4] = R 192.168.5.16/30 [255/2] via 192.168.5.2, eth1
- [5] = R 192.168.5.20/30 [255/2] via 192.168.5.10, eth3
- [6] = C 192.168.1.0/24 is directly connected, eth0
- [7] = R 192.168.2.0/24 [255/2] via 192.168.5.6, eth2
- [8] = R 192.168.3.0/24 [255/2] via 192.168.5.10, eth3
- [9] = R 192.168.4.0/24 [255/3] via 192.168.5.2, eth1
- [10] = C 127.0.0.0/8 is directly connected, lo0

```

(b.) R1 – zapojenie v OMNeT++

Obrázok 4.6 : Smerovacia tabuľka pre zariadenie R1 protokolu RIP

4.2.1 Zhrnutie testu

Výsledky testu ustanovenia stavu susedstva sú zobrazené v tabuľke 4.7.

Protokol	Čas potrebný v simulátore [s]	Čas potrebný v reálnych zariadeniach [s]
EIGRP	0,099	3,054
Babel	0,429	3,842
RIPv1	7,176	27,832
RIPv2	4,223	23,714

Tabuľka 4.7 : Čas potrebný pre ustanovenie susedstva pre všetky protokoly

Čas uvedený v tabuľke 4.7 je doba, ktorú potrebujú zariadenia na vzájomnú komunikáciu a ustanovenie podmienok výmeny smerovacích informácií. Posledná správa v tomto časovom rozmedzí je správa, v ktorej si zariadenia vymenia smerovacie informácie.

Najväčší vplyv na test, hlavne pri reálnych zariadeniach, malo zaťaženie liniek. Jedno zariadenie komunikovalo naraz s viacerými susedmi. Ďalej záležalo na čase, za ktorý sa dokázal zapnúť proces smerovania. Najdlhšie to trvalo pri testovaní Babel protokolu s verziou babeld na Linux Ubuntu. Bolo to spôsobené tým, že Babel nie je dostupný na CISCO zariadeniach. Z toho

dôvodu sa nedalo testovať rovnakým spôsobom, ako ostatné protokoly. Najprv sa museli všetky rozhrania nastaviť do stavu up a následne sa na nich zapol proces babeld. Zatiaľ čo na CISCO zariadeniach proces protokolu bežal, rozhrania boli nastavené na stav down. Pomocou skriptu sa tieto rozhrania naraz nastavili do stavu up. To ušetrilo čas nutný na zapnutie procesu.

Test potvrdil, že najrýchlejšie nadviazať komunikáciu a vymeniť informácie dokáže protokol EIGRP.

4.3 Porovnanie správ jednotlivých protokolov

V tomto bol použitý rovnaký scenár ako pri teste nadviazania susedstva. Narozdiel od prvého testu sme sledovali veľkosť prenesených správ nutných k uzatvoreniu stavu. Pre testovanie bola použitá topológia siete z obrázku 4.1.

Správy boli sledované od nastavenia rozhrania medzi zariadeniami R1 a R2 na stav up, až po stav, v ktorom si obe zariadenia vymenili smerovacie informácie. Následne zmeny v ich smerovacích tabuľkách, či správy slúžiace k udržaniu susedstva, neboli brané do úvahy. Výpis poslaných správ a ich veľkostí sú v tabuľkách 4.7 až 4.9.

Protokol	
EIGRP	
Typ správy	Veľkosť [bytes]
Hello	74
Hello	84
Hello	84
Update(INIT)	60
Update(INIT)	54
Update	144
Ack	54
Update	188
Ack	54
Update	144
Ack	54
Spolu :	994

Tabuľka 4.7 : Veľkosť poslaných správ pre EIGRP

Protokol	
Babel	
Typ správy	Veľkosť [bytes]
Hello, Route Request	83
Hello, IHU, Update	86
Hello, Route Request	83
Hello, IHU, Update	86
Hello, IHU	90
Hello, IHU	90
Route Request	91
Route Request	155
Update	242
Hello, IHU	90
Update	126
Hello, IHU	90
Spolu :	1312

Tabuľka 4.8 : Veľkosť poslaných správ pre Babel

Protokol			
RIPv1		RIPv2	
Typ správy	Veľkosť [bytes]	Typ správy	Veľkosť [bytes]
Request	66	Request	66
Response	226	Response	226
Response	226	Response	146
Spolu:	518	Spolu:	438

Tabuľka 4.9 : Veľkosť poslaných správ pre RIP

Protokoly RIP v oboch verziách posielajú pri pravidelných aktualizáciách vždy celú tabuľku. Preto je veľkosť správ podobná po celý čas behu tohto protokolu. Protokoly EIGRP a Babel po prvotnom vymenení informácií posielajú správy značne menšie. EIGRP posielajú správy iné ako Hello len pri zmene v topológii. Babel posielajú pravidelne aj správy Update, ale neobsahujú celú tabuľku.

4.4 Záverečné zhrnutie porovnania

Všetky tri porovnávané protokoly používajú metódy Split horizon a Poison reverse. EIGRP, na rozdiel od ostatných, posielajú smerovacie informácie o topológii len pri zmene v nej. RIP a Babel využívajú pravidelné zasielanie informácií. RIP posielajú vždy celú smerovaciu tabuľku, EIGRP a Babel len zmenené informácie. Protokoly EIGRP a Babel podporujú komunikáciu vo verzii IPv6. Pre podporu IPv6 protokolom RIP je nutné použiť jeho verziu RIPng. Vo všeobecnosti je momentálne najrozšírenejší protokol RIP, zatiaľ čo najmenej sa používa Babel.

Ďalšie rozdiely medzi protokolmi v používaných technológiách či ich vlastnostiach sú uvedené v tabuľke 4.8.

	EIGRP	Babel	RIPv1	RIPv2
Šírenie správ	multicast	multicast	broadcast	multicast
IPv4 adresa	224.0.0.10	224.0.0.111	255.255.255.255	224.0.0.9
Transportná vrstva	TCP/IP	UDP	UDP	UDP
IPv6	✓	✓	RIPng	RIPng
Použitý algoritmus	DUAL	Bellman-Ford	Bellman-Ford	Bellman-Ford
Podpora CISCO	✓	✗	✓	✓
Interval posielania dotazovacích správ	60 sec ³	20 sec ⁴	30 sec	30 sec
Nekonzistentné siete	✓	✗	✗	✓
Autentifikácia	✓	✗	✗	✓
VLSM	✓	✓	✗	✓
Classless	✓	✓	✗	✓
Classfull	✓	✓	✓	✓

Tabuľka 4.8 : Porovnanie protokolov

³ Linky rýchlejšie ako 1,54 Mbps majú interval 5 sekúnd

⁴ Bezdrôtové linky majú interval 4 sekundy

5 Záver

Pre analýzu a verifikáciu siete sú potrebné nástroje na simuláciu. V tejto práci sme využívali nástroj OMNeT++ s knižnicou INET a ANSAINET. Knižnice nám poskytli potrebné moduly so skúmanými protokolmi aj vďaka skvelej práci bývalých študentov, ktorí implementovali Babel a protokol EIGRP. Cieľom práce je porovnať medzi sebou smerovacie protokoly zo skupiny Distance Vector.

Podrobne sme opísali jednotlivé protokoly a vysvetlili, v čom sa od seba líšia a naopak, čím sú si podobné. Pri všetkých protokoloch sme sledovali parametre, ktoré sú potrebné pre pochopenie ich funkcionality. Skúmali sme prostredie simulátora OMNeT++ a naučili sme sa ho používať pri porovnávaní. Po naštudovaní prostredia simulátora sme sa pustili do samotného porovnania protokolov v simulátore, ako aj v reálnom zapojení. Pri reálnom zapojení sme sa zamerali na samotnú virtualizáciu, ktorá je menej náročná na časové, ako aj finančné podmienky.

Literatúra

- [1] SAVAGE, D.; Slice, D. RFC 7868: Enhanced Interior Gateway Routing Protocol [Online]. August 2013 [cit.2017-01-02].
URL <http://tools.ietf.org/html/draft-savage-igrp-00>
- [2] CHROBOCZEK, Juliusz. RFC 6126: The Babel Routing Protocol [online]. 2011 [cit.2017-01-02].
URL <https://tools.ietf.org/html/rfc6126>
- [3] HEDRICK, C. RFC 1058: Routing Information Protocol [online]. 1988 [cit.2017-01-02].
URL <https://tools.ietf.org/html/rfc1058>
- [4] Cisco System, Inc.: Enhanced Interior Gateway Routing Protocol [online]. 2016 [cit.2017-01-02].
URL <http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-igrp/16406-igrp-toc.html>
- [5] Cisco System, Inc.: An Introduction to IGRP [online]. 2015 [cit.2017-01-20].
URL <http://www.cisco.com/c/en/us/support/docs/ip/interior-gateway-routing-protocol-igrp/26825-5.html>
- [6] Cisco System, Inc.: EIGRP Wide Metrics [online]. Prosinec 2012 [cit. 2014-02-05].
URL http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_egrp/configuration/15-sy/ire-15-sy-book/ire-wid-met.html
- [7] CHROBOCZEK, Juliusz. Babel: A flexible routing protocol [online]. 2014 [cit.2017-01-02].
URL <http://www.pps.univ-paris-diderot.fr/~jch/software/babel/babel-20140311.pdf>
- [8] INET Framework: INET Framework for OMNeT++, Manual. 2015 [cit.2017-01-02].
URL <https://inet.omnetpp.org/DocumentationGuidelines.html>
- [9] Varga, A.: OMNeT++, User Guide, Version 5.0. 2016 [cit.2017-01-02].
URL <http://www.omnetpp.org/doc/omnetpp/manual/usman.html>
- [10] VESELÝ, Vladimír, Vít Rek, Ondrej Ryšavý. Enhanced Interior Gateway Routing Protocol with IPv4 and IPv6 Support for OMNeT++. 2014 [cit. 2017-01-02].
URL <http://www.fit.vutbr.cz/~ivesely/pubs.php?file=%2Fpub%2F11092%2Feigrp-vesely.pdf&id=11092>
- [11] Git repozitář: kvetak/ANSA [online]. [cit.2017-01-02].
URL <https://github.com/kvetak/ANSA>
- [12] SCHULZRINNE, H., Casner, S., Frederick, R. , Jacobson, V.; RFC 3550: RTP: A Transport Protocol for Real-Time Applications [Online]. July 2003 [cit.2017-01-28].
URL <https://tools.ietf.org/html/rfc3550>
- [13] Teare, D.: Implementing Cisco IP Routing (ROUTE). Cisco Press, 2010, ISBN-13:978-1-58705-82-0 [cit.2017-01-28].

- [14] Cisco System, Inc.: Routing Information Protocol [online]. 2012 [cit.2017-01-28].
URL http://docwiki.cisco.com/wiki/Routing_Information_Protocol
- [15] MALKIN, G. RFC 2453: RIP Version 2 [online]. 1998 [cit.2017-01-28].
URL <https://tools.ietf.org/html/rfc2453>
- [16] MALKIN, G; Minnear, R., RFC 2080: RIPng for IPv6 [online]. 1997 [cit.2017-01-28].
URL <https://tools.ietf.org/html/rfc2080>
- [17] ANSA: Automated Network Simulation and Analysis [online]. 2008 [cit.2017-01-28].
URL <https://ansa.omnetpp.org/>
- [18] DEL REY, M. RFC 791: Internet Protocol [Online]. September 1981 [cit.2017-05-07].
URL <https://www.ietf.org/rfc/rfc791.txt>
- [19] DEERING, S., Hinden, R. RFC 2460: Internet Protocol, Version 6 (IPv6) [Online].
December 1998 [cit.2017-05-07].
URL <https://www.ietf.org/rfc/rfc2460.txt>
- [20] MOGUL, J.. RFC 919: Broadcasting Internet Datagrams [Online].
October 1984 [cit.2017-05-07].
URL <https://tools.ietf.org/html/rfc919>
- [21] SAVOLA, P.. RFC 6308: Overview of the Internet Multicast Addressing Architecture [Online].
June 2011 [cit.2017-05-07].
URL <https://tools.ietf.org/html/rfc6308>
- [22] PUMILL, T.; Manning, B. RFC 1878: Variable Length Subnet Table for IPv4 [Online].
December 1995 [cit.2017-05-07].
URL <https://tools.ietf.org/html/rfc1878>
- [23] FULLER, V., Li, T., Yu, J., Varadhan, K. RFC 1519: Classless Inter-Domain Routing (CIDR).
September 1993 [cit. 2017-05-07].
URL <https://tools.ietf.org/html/rfc1519>
- [24] Cisco System, Inc.: Cisco Doc Wiki [online]. December 2011 [cit.2017-05-07].
URL <http://docwiki.cisco.com/wiki>
- [25] FAIRHURST, G., Wood, L. RFC 3366: Advice to link designers on link Automatic Repeat
request (ARQ) [Online]. August 2002 [cit.2017-05-07].
URL <https://tools.ietf.org/html/rfc3366>
- [26] POSTEL, J. RFC 3366: User Datagram Protocol [Online]. August 1980 [cit.2017-05-07].
URL <https://www.ietf.org/rfc/rfc768.txt>
- [27] Simulator OMNeT++: Mobility Framework for OMNeT++ [Online]. 2009 [cit.2017-05-07].
URL <http://mobility-fw.sourceforge.net/hp/index.html>
- [28] Unified Networking Labs: The Emulated Virtual Environment [Online]. 2009 [cit.2017-05-07].
URL <http://mobility-fw.sourceforge.net/hp/index.html>

Príloha A

Obsah priloženého CD

Obsah adresárov na priloženom CD nosiči je uvedený v tabuľke A.1.

Adresár	Popis obsahu adresára
/Babel	Kópia virtuálneho stroja Linux Ubuntu s implementovaným protokolom babeld
/config	Konfigurácie jednotlivých protokolov
/labs	Vytvorené laboratória v OMNeT++ ako aj UNetLab
/outputs	Výstupné súbory programu Wireshark a výstupy ladiacich výpisov
/doc	Bakalárska práca

Tabuľka A.1 : Obsah priloženého CD

Príloha B

Konfigurácia Babeld

V tabuľke B.1 sa nachádza konfigurácia pre protokol Babel vo verzii babeld.

Kľúčové slovo	Parameter	Popis
protocol-group group	-m	Multicastova adresa pre zasielanie správ
protocol-port port	-p	UDP port pre zasielanie správ
kernel-priority priority	-k	Priorita použitá pri zadaní cesty do jadra
allow-duplicates priority		Duplikácia ciest s prioritou minimálnej priority
keep-unfeasible [true false]	-u	Určuje, či budú nevhodné cesty udržiavané v pamäti
random-id [true false]	-r	Použitie náhodného identifikátora smerovača
debug level	-d	Určuje úroveň ladiacich výpisov
local-port port	-g	TCP port, pre pripojenie grafickej nadstavby
export-table table	-t	Smerovacia tabuľka pre ukladanie ciest
import-table table	-T	Smerovacia tabuľka ktorej cesty sú distribuované
link-detect [true false]	-l	Použitie CS pre určenie dostupnosti rozhrania
diversity [true false kind]		Algoritmus pre určovanie ciest rušených bezdrôtových spojov
diversity-factor factor		Faktor určujúci zvýhodnenie nerušených bezdrôtových liniek
smoothing-half-life seconds	-M	Polčas zmeny exponenciálneho oneskorenia, používa sa pri zohľadňovaní histórie metrik
daemonise [true false]	-D	Program pobeží ako démon
state-file filename	-S	Umiestnenie súboru obsahujúceho stav
log-file filename	-L	Umiestnenie súboru obsahujúceho log
pid-file filename	-I	Umiestnenie súboru obsahujúceho identifikátor procesu
interface name [param]		Konfigurácia sieťového rozhrania name. Možné parametre sú uvedené v tabuľke B.2.
default name [param]		Štandardná konfigurácia sieťových rozhraní . Možné parametre sú uvedené v tabuľke B.2.

Tabuľka B.1 : Konfigurácia babeld

Kľúčové slovo	Popis
wired [true false auto]	Optimalizácia špecifická pre drôtové rozhrania
link-quality [true false auto]	Odhad kvality linky, v štandardnom nastavení sa používa len na bezdrôtové rozhrania
split-horizon [true false auto]	Použitie mechanizmu split-horizon, štandardne je na bezdrôtových rozhraniach vypnutý
rxcost cost	Určuje cenu príjmu správ za ideálnych podmienok, štandardná hodnota je 96 pre drôtové a 256 pre bezdrôtové rozhrania
channel channel	Kanál používaný bezdrôtovým rozhraním, štandardne je detekovaný automaticky
faraway [true false]	Určuje vzdialenosť siete aby nedošlo k rušeniu
hello-interval interval	Interval zasielania správ hello
update-interval interval	Interval zasielania periodických aktualizácií ciest
enable-timestamps [true false]	Odosielanie časových značiek so správami hello a IHU pre výpočet RTT
rtt-decay decay	Faktor úpadku, hodnota v rozmedzí 1 až 256
rtt-min rtt	Minimálne RTT v milisekundách, prvotne 10ms
rtt-max rtt	Maximálna RTT v milisekundách, prvotne 120ms
max-rtt-penalty cost	Maximálna penalizácia ceny linky z dôvodu oneskorenia linky, prvotne 0

Tabuľka B.2 : Konfigurácia parametrov sieťových rozhraní pre babeld

Príloha C

Konfigurácia EIGRP

Popis príkazov používaných pri konfigurácii protokolu EIGR je v tabuľke C.1. Príkazy používané pre výpis informácií o smerovaní protokolom EIGRP sú v tabuľke C.3.

Príkaz	Popis
<code>router eigrp AS</code>	Zapnutie EIGRP procesu, AS je číslo autonómneho systému
<code>network network wildcard</code>	Zapojenie rozhrania do procesu EIGRP
<code>bandwidth num</code>	Maximálny dátový tok rozhraním, používa sa pre výpočet metriky
<code>auto-summary</code>	Autosumarizácia sietí
<code>variance number</code>	Nerovnomerné rozvažovanie
<code>passive-interface interface</code>	Vypnutie šírenia smerovacích informácií na danom rozhraní
<code>eigrp stub [param]</code>	Nastavenie rozposielania smerovacích nastavení, bez parametra posiela sumárne a priamo pripojené siete. Parametre sú opísané v tabuľke C.2
<code>eigrp router-id address</code>	Nastavenie ID zariadenia
<code>default-metric num</code>	Nastavenie metriky použitej pri smerovaní
<code>distance num</code>	Nastavenie administratívnej distance
<code>default-information [in out] [Access list]</code>	Predvolené smerovacie informácie ktoré sa budú šíriť
<code>ip authentication key-chain eigrp AS key</code>	Zapnutie autentifikácie paketov pre EIGRP
<code>ip authentication mode eigrp AS md5</code>	Nastavenie módu autentifikácie paketov
<code>ip hello-interval eigrp AS sec</code>	Nastavenie časovača odosielania hello správ
<code>ip hold-time eigrp AS sec</code>	Nastavenie hold-time časovača
<code>ip split-horizon eigrp AS</code>	Nastavenie mechanizmu split horizon
<code>neighbor address int-type int-num</code>	Definovanie susedného zariadenia

Tabuľka C.1 : Konfigurácia EIGRP

Príkaz	Popis
receive-only	Nepreposiela smerovacie informácie
connected	Posiela len priamo pripojené siete
static	Posiela len staticky pripojené siete
summary	Posiela len sumárne siete

Tabuľka C.2 : Parametre príkazu eigrp stub

Príkaz	Popis
Show ip eigrp neighbors	Zobrazí tabuľku susedov. Môže sa použiť s parametrom detail, ktorý zobrazí detailne informácie
Show ip eigrp interfaces <i>interface</i>	Zobrazí informácie pre rozhranie. Bez parametru zobrazí informácie o rozhraniach zapojených do procesu
Show ip eigrp topology	Zobrazí tabuľku topológie
Show ip eigrp traffic	Zobrazí informácie o posielaných a prijatých správach
Show ip route eigrp	Zobrazí smerovaciu tabuľku pre EIGRP
Debug eigrp packets	Zobrazí ladiace výpisy pre EIGRP smerovanie

Tabuľka C.3 : Príkazy pre zobrazenie informácií o smerovaní EIGRP

Príloha D

Konfigurácia RIP

Popis príkazov používaných pri konfigurácii protokolu RIP je v tabuľke D.1. Príkazy používané pre výpis informácií o smerovaní protokolom EIGRP sú v tabuľke D.2.

Príkaz	Popis
auto-summary	Nastavenie autosumarizácie sietí
default-information originate [map map-name]	Nastavenie predvolenej trasy pre protokol RIP
default-metric num	Nastavenie metriky pre smerovanie RIP
ip rip authentication key-chain name	Nastavenie autentifikácie správ pre protokol RIPv2
ip rip authentication mode [text md5]	Nastavenie módu autentifikácie správ
ip rip v2-broadcast	Nastavenie posielania správ broadcastom namiesto multicastu pre verziu RIPv2
ip split-horizon	Nastavenie mechanizmu split horizon
ip summary-address rip address mask	Konfigurácia súhrnnej adresy
neighbor address	Definovanie susedného zariadenia
network address	Špecifikovanie siete ktorá sa zapojí do procesu smerovania
passive-interface interface	Vypnutie šírenia smerovacích informácií na danom rozhraní
router rip	Zapnutie procesu smerovania protokolom RIP
version [1 2]	Nastavenie verzie protokolu RIP

Tabuľka D.1 : Konfigurácia RIP

Príkaz	Popis
Show ip rip neighbors	Zobrazí tabuľku susedov. Môže sa použiť s parametrom detail, ktorý zobrazí detailne informácie
Show ip route rip	Zobrazí smerovaciu tabuľku pre RIP
Debug ip rip	Zobrazí ladiace výpisy pre RIP smerovanie

Tabuľka D.2 : Príkazy pre zobrazenie informácií o smerovaní pre RIP