ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ

KATEDRA TECHNOLOGIÍ A MĚŘENÍ

BAKALÁŘSKÁ PRÁCE

Současné routovací protokoly

Marek Švamberg **2014**

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: **2013/2014**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Marek ŠVAMBERG**
Osobní číslo:	**E10B0124P**
Studijní program:	**B2612 Elektrotechnika a informatika**
Studijní obor:	**Komerční elektrotechnika**
Název tématu:	**Současné routovací protokoly**
Zadávající katedra:	**Katedra technologií a měření**

Zásady pro vypracování:

1. Popište algoritmy používané routovacími protokoly na třetí vrstvě ISO/OSI modelu.
2. Sestavte přehled v současnosti používaných protokolů.
3. Porovnejte vlastnosti vybraných protokolů.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah pracovní zprávy: **20 - 30 stran**

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.

Vedoucí bakalářské práce: **Ing. Jan Broulím**
Regionální inovační centrum elektrotechniky

Datum zadání bakalářské práce: **14. října 2013**

Termín odevzdání bakalářské práce: **9. června 2014**

Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan

Doc. Ing. Vlastimil Skočil, CSc.
vedoucí katedry

Abstrakt

Bakalářská práce se zabývá směrováním a směrovacími protokoly na třetí vrstvě ISO/OSI modelu. V první části jsou popsané algoritmy pro hledání nejlepší cesty v grafu. Dále popis jednotlivých druhů směrování a k nim využívaných směrovacích protokolů. V poslední části je ukázán návrh a konfigurace topologie sítě.

Klíčová slova

Matice sousednosti, Incidenční matice, Dijkstrův algoritmus, Bellman-Fordův algoritmus, směrovací protokol, RIPv1, RIPv2, EIGRP, OSPF, IS-IS, BGP

Abstract

The bachelor thesis deals with routing and routing protocols on 3rd layer of ISO/OSI model. In the first part algorithms for finding the best path through graph are described. The next part is focused on routing and types of routing protocols. In the end the thesis shows an example of a design and network topology configuration.

Key words

Adjacency matrix, Incidence matrix, Dijkstra's algorithm, Bellman-Ford's algorithm, routing protocol, RIPv1, RIPv2, EIGRP, OSPF, IS-IS, BGP

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

.....

podpis

V Plzni dne 3.6.2014 Marek Švamberg

Poděkování

Tímto bych chtěl poděkovat vedoucímu mé bakalářské práce Ing. Janu Broulímovi za veškeré rady a pomoc při vypracování práce.

Obsah

Seznam symbolů a zkratek

RIP .....routing information procol

AS .....autonomous system

IGP .....interior gateway protocol

EGP.....exterier gateway protocol

LSP.....link-state protocol

LSDB .....link-state database

EIGRP.....Enhanced Interior Gateway Routing Protocol

OSPF.....Open Shortest Path First

LSA.....Link-state advertisment

LSP.....Link-state packet

BGP.....Border Gateway Protocol

IS-IS .....Intermediate System to Intermediate System

DBD.....Database descriptor

IANA .....Internet Assigned Numbers Authority

Úvod

Dnešní svět už si jen s obtížemi dokáže představit den bez internetu, kdy ho využíváme pro naši práci. Ať už se jedná například o internetové obchodování, komunikování anebo pouze udržování vnitřní sítě ve společnosti nebo školách. Tato bakalářská práce se zabývá právě principy fungování směrování v počítačových sítích.

V první části jsou nejdříve uváděny základy k teorii grafů, ze kterých vychází samotné směrování. Jde o základní definice používaných pojmů, matice pro popis grafů a algoritmy sloužící k nalezení cesty skrze graf. Poté jsou v druhé kapitole vysvětleny druhy dynamického směrování, které vycházejí z algoritmů z předchozí části. Ve třetí kapitole jsou popsané jednotlivé směrovací protokoly 3. vrstvy ISO/OSI modelu zaručující správnou komunikaci v síti i mezi autonomními systémy. V poslední části je ukázaná konfigurace RIP a BGP protokolů na příkladu topologie s vnitřním i vnějším systémem.

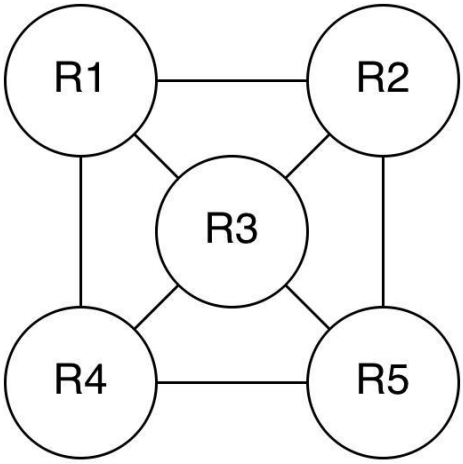
Cílem bakalářské práce tedy bylo shrnout principy směrování a popsat jednotlivé protokoly k tomu používané a nakonfigurovat praktickou ukázku příkladu topologie sítě s použitím směrovacích protokolů.

1 Algoritmy a popis grafů

1.1 Základní pojmy

Ačkoliv to možná na první pohled nevidíme, v našem každodenním životě nenajdeme alespoň jednu činnost, při které by se nevyužilo znalosti grafu nebo jakékoliv jeho se týkající vlastnosti. Ať už se jedná například o znázornění hierarchie pracovních pozic nebo hodně zjednodušenou mapu, ve které budou zakresleny pouze města, jako body a silnice, které je spojují, jako hrany grafu. V souvislosti s touto bakalářskou práci jsou jednotlivé body zastoupeny všemi zařízeními, ať už to jsou routery, switche nebo koncové počítače. Hrany grafu naopak pozměníme na linky spojující zařízení.

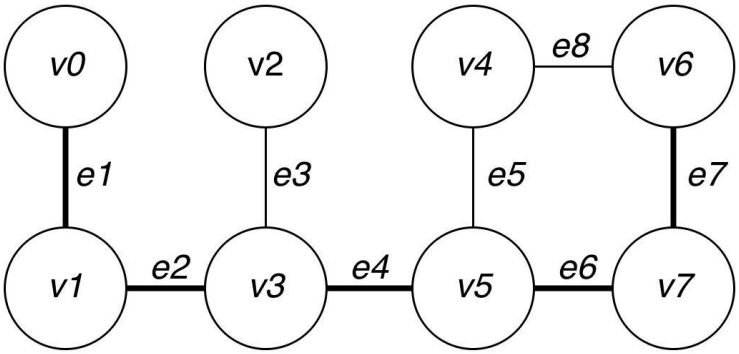
Graf je tedy ideálním prvkem, jak zjednodušeně zakreslit počítačovou síť viz obr. 1, kde známe celkový počet zařízení, tedy bodů a spojeních mezi nimi, hran. Graf je zobrazení bodů a vzájemných vztahů mezi nimi.



Obr. 1: Zobrazení sítě směrovačů jako graf a jejich vzájemné propojení

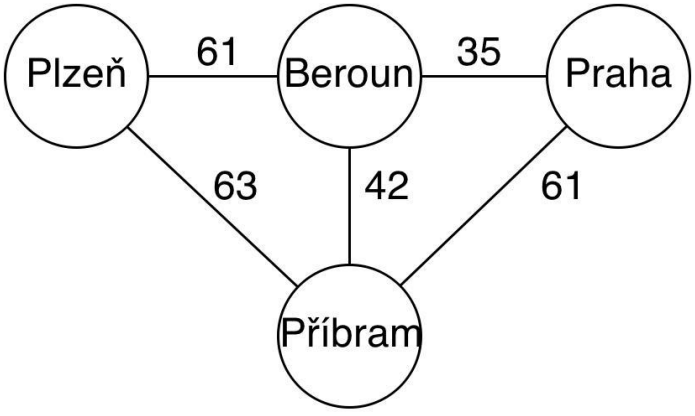
Naproti tomu se nám může stát, že při průchodu grafu budeme muset někde použít jednu konkrétní cestu. V praxi je to to samé, jako když nám dvě křižovatky spojuje jednosměrná silnice a máme přikázaný směr jízdy. V takovém případě se graf nazývá orientovaný graf a definovat ho lze jako uspořádanou dvojici vrcholů a hran (*V*, *E*), kde *E* je podmnožina kartézského součinu $V \times V \times V$. Prvky množiny *E* se nazývají orientované hrany. Orientovaná hrana *e* má tvar (*x*, *y*), tedy vychází z *x* a končí v *y*.

Další důležitou součástí při průchodu grafu je jeho cesta. Při přenesení do praxe je to stejné, jako když chceme dojet do nějakého vzdáleného města, kam neznáme cestu, a tak si zjistíme všechna města na cestě, kterými musíme projet, abychom správně dojeli. Stejně tak v počítačových sítích zařízení musí znát cesty, přes které posílat zprávy ke svým cílům. Využití cesty je dále vidět v kapitole 1.5.1. Definovat cestu v grafu můžeme jako posloupnost vrcholů a hran (*v0*, *e1*, *v1*,…, *et*, *vt*), kde vrcholy *v0*,…, *vt* jsou navzájem různé vrcholy grafu *G* a pro každé *i* = 1, 2,…, *t* je *ei* = {*vi-1*, *vi*} ϵ E(*G*).



Obr. 2: Příklad cesty v grafu (*v0*, *e1*, *v1*, *e2*, *v3*, *e4*, *v5*, *e6*, *v7*, *e7*, *v6*)

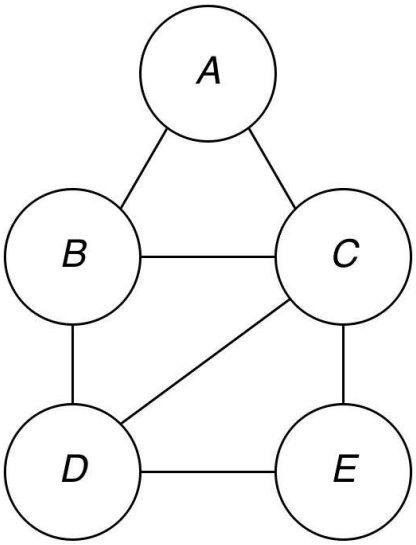
V neposlední řadě je také další důležitou vlastností pro určení správné cesty metrika grafu. Jde o ohodnocení hran grafu. Může se jednat například o vzdálenost cesty, nebo jestli jsou na cestě nějaké poplatky za její použití, případně nějaká omezení. Metrika bude dále v textu používaná u směrovacích protokolů při hledání cest. Jde tedy o číslo, které nám reprezentuje vzdálenost vrcholů mezi danou hranou. Matematická definice pro souvislý graf *G* je: pro vrcholy v, v' definujme číslo dG(v,v'), jako délku nejkratší cesty z v do v' v grafu *G*. Číslo dG(v, v') se nazývá vzdálenost vrcholů v a v' v grafu *G*.



Obr. 3: Příklad metriky ukazující vzdálenosti měst mezi sebou

1.2 Matice sousednosti

Jedním z nejzákladnějších zobrazení grafů je pomocí matice sousednosti. Ta dává informace, které vrcholy jsou mezi sebou propojeny a které nemají společnou hranu. Každá matice bude mít rozměry $n \times n$, kde n je rovno počtu vrcholů grafu. Poté, když spolu dva vrcholy sousedí, napíšeme do matice „1“ v opačném případě „0“ [1, 2]. Například pro graf znázorněný na obr. 4 dostaneme následující matici:



Obr. 4: Neorientovaný graf

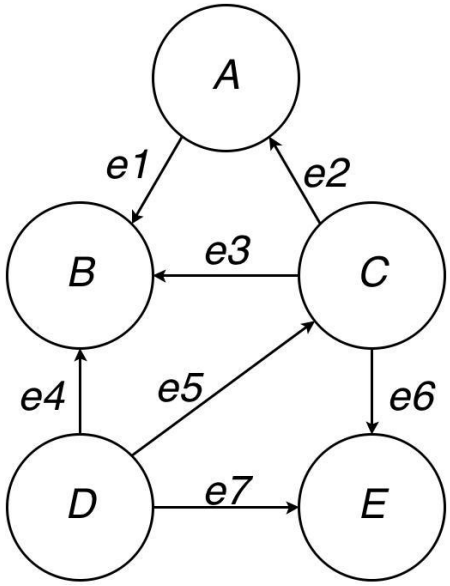
$$A = \begin{matrix} A & B & C & D & E & \\ 0 & 1 & 0 & 1 & 1 & A \\ 1 & 0 & 1 & 1 & 0 & B \\ 0 & 1 & 0 & 1 & 0 & C \\ 1 & 1 & 1 & 0 & 1 & D \\ 1 & 0 & 0 & 1 & 0 & E \end{matrix}$$

Obr. 5: Matice sousednosti pro graf

1.3 Incidenční matice

Další způsob popisu grafu je pomocí incidenční matice. Z incidenční matice zjistíme, jaké vztahy mají vrcholy s přilehlými hranami. Opět zde vznikne stejná matice jako u matice sousednosti, tedy o rozměrech $n \times n$, kde n je počet vrcholů. Zde nebudeme zapisovat pouze, jestli jsou jednotlivé vrcholy s hranami sousedy, ale i jestli hrany do vrcholu vchází nebo z něj vycházejí a to následovně [2]:

- pokud hrana z vrcholu vychází: +1,
- pokud hrana do vrcholu vchází: -1,
- všechno ostatní: 0.



Obr. 6: Orientovaný graf

$$A = \begin{matrix} e1 & e2 & e3 & e4 & e5 & e6 & e7 & \\ +1 & -1 & 0 & 0 & 0 & 0 & 0 & A \\ -1 & 0 & -1 & -1 & 0 & 0 & 0 & B \\ 0 & +1 & +1 & 0 & -1 & +1 & 0 & C \\ 0 & 0 & 0 & +1 & +1 & 0 & +1 & D \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & E \end{matrix}$$

Obr. 7: Incidenční matice pro graf

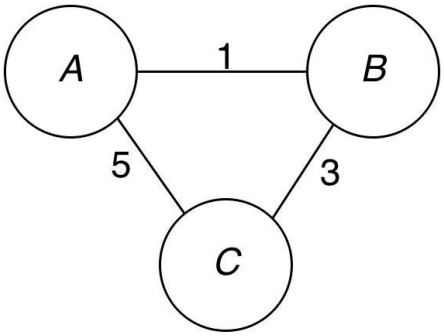
1.4 Dijkstrův algoritmus

Dijkstrův algoritmus byl navržen roku 1959 nizozemským informatikem Edsgerem Wybe Dijkstrem [3]. Slouží ke hledání nejkratší cesty v grafu s kladným oceněním jeho hran. Jeho princip spočívá v tom, že graf prochází podle vzdáleností hran od zdrojového uzlu k cílovému.

Princip algoritmu [3]:

- přiřazení hodnoty „0“ ke zdrojovému uzlu a všem ostatním „∞“,
- přiřazení vzdálenosti hrany mezi zdrojovým a sousedním uzlem,
- určení nejmenší vzdálenosti dané cesty a z tohoto uzlu udělat trvalý,
- pokud je možno cílový uzel dosáhnout více cestami, zvolit tu s nejmenší vzdáleností,
- opakování předchozích kroků dokavaď každý uzel nebude prozkoušený.

1.4.1 Příklad k výpočtu Dijkstrova algoritmu

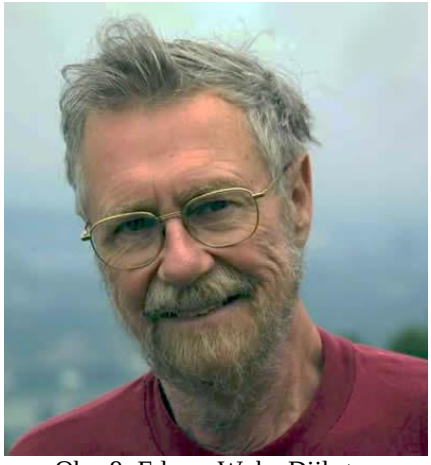


Obr. 8: Graf k příkladu Dijkstrova algoritmu

Máme graf znázorněný na obr. 8. V něm budeme chtít zjistit nejkratší cesty z vrcholu *A* do ostatních. V prvním kroku si označíme vrchol *A* jako zdrojový uzel a tomu přiřadíme hodnotu „0“. Ke všem ostatním uzlům, přiřadíme „∞“. V dalším kroku zjistíme vzdálenosti jeho sousedů. K vrcholu *B* je vzdálenost hrany „1“ a k vrcholu *C* vzdálenost „5“ a tyto hodnoty jim přiřadíme. Ve třetím kroku budeme vycházet se zatím nejbližšího dalšího vrcholu, vrcholu B. Z něj budeme zkoumat jeho sousední vrcholy. Tím, že vrchol *A* už máme určený, budeme se soustředit pouze na vrchol *C*. Do vrcholu *C* vede cesta přes *B* „1“ + „3“, celková vzdálenost je „4“. Tím máme určené všechny vzdálenosti cest a zjistíme, že nejkratší cesta z *A* do *B* je přes jejich přímo spojenou hranou. A cesta z *A* do *C* bude nejkratší při průchodu vrcholem *B* [3].

Tabulka 1 Postup Dijkstrova algoritmu

	A	B	C
1. krok	0	∞	∞
2. krok	0	1	5
3. krok	0	1	4



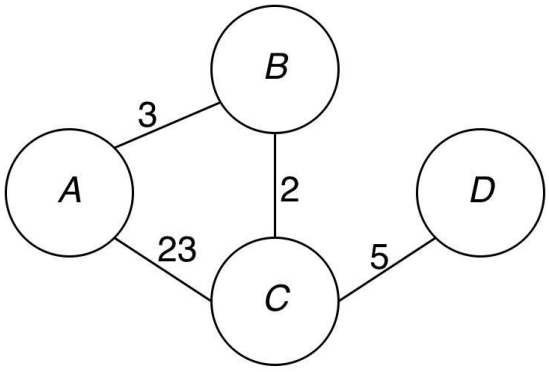
Obr. 9: Edsger Wybe Dijkstra
Zdroj: http://www.thocp.net/biographies/pictures/dijkstra_edgar1.jpg

1.5 Bellman-Fordův algoritmus

Bellman-Fordův algoritmus počítá, stejně jako Dijkstrův algoritmus, nejkratší možné cesty v grafu. Na rozdíl od něj ale umí pracovat i se zápornými hodnotami jednotlivých hran, což ho ale také dělá o něco pomalejším než předešlý algoritmus. Byl představen americkými matematiky Richardem Bellmanem a Lesterem Fordem [4].

1.5.1 Příklad k výpočtu Bellman-Fordova algoritmu

Máme zadaný graf na obr. 10. V prvním kroku si vytvoříme počáteční tabulku. Všem zdrojovým vrcholům a k nim neplatných cest se přiřadí hodnota „0“ a všem přímo připojeným sousedům jejich metriky. Všem ostatním cestám přiřadíme „∞“. A hodnotou „x“ označíme všechny neplatné cesty [4].



Obr. 10: Graf k přikladu Bellman-Fordova algoritmu (převzato z [3])

Tabulka 2 Počáteční tabulky pro vrcholy

z A	přes A	přes B	přes C	přes D
A	0	0	0	0
B	0	3	∞	x
C	0	∞	23	x
D	0	∞	∞	x

z B	přes A	přes B	přes C	přes D
A	3	0	∞	x
B	0	0	0	0
C	∞	0	2	x
D	∞	0	∞	x

z C	přes A	přes B	přes C	přes D
A	23	∞	0	∞
B	∞	2	0	∞
C	0	0	0	0
D	∞	∞	0	5

z D	přes A	přes B	přes C	přes D
A	x	x	∞	0
B	x	x	∞	0
C	x	x	5	0
D	0	0	0	0

V dalším kroku se všechny vrcholy dozví nové vzdálenosti od všech přímo připojených sousedních vrcholů. Z nových informací původní vrcholy zjistí přes jejich vrcholy nově

dosažitelné další vrcholy. Pokud nově zjištěné cesty budou v tuto chvíli známé jako nejkratší, jednotlivé vrcholy si je prozatím uloží. Po provedení tohoto na každém vrcholu vzniknou nové průběžné tabulky [4].

Tabulka 3 Tabulky vzdáleností po prvním přepočítání

z A	přes A	přes B	přes C	přes D
A	0	0	0	0
B	0	3	25	x
C	0	5	23	x
D	0	∞	28	x

z B	přes A	přes B	přes C	přes D
A	3	0	25	x
B	0	0	0	0
C	26	0	2	x
D	∞	0	7	x

z C	přes A	přes B	přes C	přes D
A	23	5	0	∞
B	26	2	0	∞
C	0	0	0	0
D	∞	∞	0	5

z D	přes A	přes B	přes C	přes D
A	x	x	25	0
B	x	x	7	0
C	x	x	5	0
D	0	0	0	0

Tím, že vrcholy získali opět nové nejkratší cesty k ostatním, tak opět pošlou svoje hodnoty svým sousedům, aby přepočítali své vzdálenosti právě s nově objevenými cestami. Pokud po novém přepočítání vrcholy objeví nové nejkratší cesty, tak si je uloží a předchozí cesty zapomenou [4].

Tabulka 4 Tabulky vzdáleností po druhém přepočítání

z A	přes A	přes B	přes C	přes D
A	0	0	0	0
B	0	3	25	x
C	0	5	23	x
D	0	10	28	x

z B	přes A	přes B	přes C	přes D
A	3	0	7	x
B	0	0	0	0
C	8	0	2	x
D	31	0	7	x

z C	přes A	přes B	přes C	přes D
A	23	5	0	33
B	26	2	0	12
C	0	0	0	0
D	51	9	0	5

z D	přes A	přes B	přes C	přes D
A	x	x	10	0
B	x	x	7	0
C	x	x	5	0
D	0	0	0	0

Po tomto novém přepočtu zjistili své nové nejkratší cesty pouze vrcholy *A* a *D*. Opět vyšlou nové informace svým sousedům, aby znovu přepočítali své vzdálenosti podle nových hodnot [4].

Tabulka 5 Výsledné tabulky pro jednotlivé vrcholy

z A	přes A	přes B	přes C	přes D
A	0	0	0	0
B	0	3	25	x
C	0	5	23	x
D	0	10	28	x

z B	přes A	přes B	přes C	přes D
A	3	0	7	x
B	0	0	0	0
C	8	0	2	x
D	13	0	7	x

z C	přes A	přes B	přes C	přes D
A	23	5	0	15
B	26	2	0	12
C	0	0	0	0
D	33	9	0	5

z D	přes A	přes B	přes C	přes D
A	x	x	10	0
B	x	x	7	0
C	x	x	5	0
D	0	0	0	0

Při posledním přepočítávání cest už žádný z vrcholů nezjistil žádné nové nejkratší cesty k ostatním, a tak ani nikdo z nich už nemusí upravovat své tabulky. Po tomto kroku se algoritmus může zastavit a vznikne konečná tabulka [4].

Tabulka 6 Konečná celková tabulka

	A	B	C	D
A	0	3	5	10
B	3	0	2	7
C	5	2	0	5
D	10	7	5	0

2 Směrování

2.1 Statické směrování

Statické směrování je založeno na pevně a především ručně napsaných směrovacích tabulkách jednotlivých zařízeních. Jde o celkem jednoduché nastavování jednotlivých hodnot, ale naproti tomu je toto směrování vhodné především pro malé sítě, kde nedochází k velkým změnám v topologii sítě. Z toho vyplývají nevýhody statického směrování, jako že správce sítě musí znát její topologii a každou změnu musíme aplikovat na každém zařízení samostatně [5].

2.2 Dynamické směrování

Druhým, a ve většině případů používanějším, způsobem směrování je dynamické směrování. Úloha dynamického směrování je na rozdíl od statického směrování v tom, že v případě například připojení nového zařízení do sítě si jednotlivé okolní zařízení zjistí změnu v topologii sítě a následném zjištěním nových směrovacích tabulek. Používají se především dva hlavní typy dynamického směrování [6]:

- algoritmus stavu linky,
- algoritmus vektoru vzdálenosti.

2.2.1 Algoritmus stavu linky

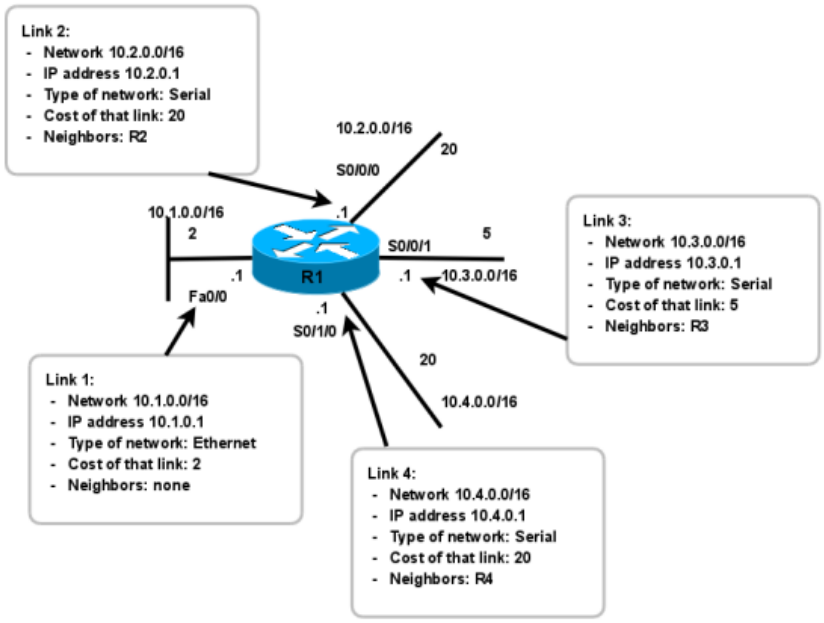
První ze dvou typů dynamického směrování je tzv. směrování typu stavu linky (link-state algorithm). Také je možné ho nazývat jako protokol nejkratší cesty, což je vlastně odvozeno podle principu jeho fungování založeném na Dijkstrovu algoritmu, kterého využívá. Pro správné pracování tohoto směrování je používáno především dvou protokolů [6]:

- OSPF protokol,
- IS-IS protokol.

Dijkstrův algoritmus je založen na hledání nejkratší cesty v síti od zdrojového k cílovému zařízení. Každé zařízení si zde vypočítává své vlastní cesty a tím si určí hodnoty jednotlivých cest, neboli metriku (cost) daného spojení ze svého konkrétního umístění v topologii sítě. Na základě dané hodnoty celkové cesty pak určí tu nejkratší možnou cestu [6].

Při provádění tohoto algoritmu každý směrovač vytváří LSP (Link-state packet), které znají všechny přímo připojené linky. Pakety stavu linky pak znají data o lince mezi dvěma

zařízeními, tj. ID sousedního zařízení, typ linky, adresu sítě, masku sítě, přenosovou kapacitu a metriku. Poté při jakékoliv změně v topologii sítě se LSP pakety posílají znovu, aby se zjistilo nové rozložení sítě a vytvořili se nové nejkratší cesty. Z takto vytvořených LSP se sestaví jedna celková databáze LSDB (Link-state database). Při rozsáhlejší oblasti se pak celková topologie rozděluje do menších oblastí a LSP se rozesílají jen v dané oblasti, aby nedocházelo ke zbytečnému zatížení procesoru [6].



Obr. 11: Informace o stavu linky pro směrovač [6]

Při porovnání protokolů stavu linky s protokoly typu vektor vzdálenosti může nalézt pár výhod [6]:

- každé zařízení si sestavuje svou vlastní topologii a metriky linek,
- LSP jsou posílány pouze při změně linky, takže zbytečně nezahlcují neustále síť,

ale i nevýhod:

- zvýšená operační paměť pro databázi,
- procesorový čas pro vypočet algoritmu,
- přenosová kapacita pro rozesílání LSP.

2.2.2 Algoritmus vektoru vzdálenosti

Druhým způsob dynamického směrování používá algoritmus vektoru vzdálenosti. Stejně jako v matematice, kde je vektor definován velikostí a směrem, tak je i stejně definován ve směrování, kde velikost představuje metrika cesty a směr je reprezentován následujícím zařízením. Na rozdíl od algoritmu stavu linky, který používá Dijkstrův algoritmus, se zde využívá principu Bellman-Fordova algoritmu. Ten stejně jako Dijkstrův algoritmus hodnotí nejkratší možné cesty, ale na rozdíl od něj umožňuje používat i záporné hodnocení cest [3].

Fungování vektoru vzdálenosti je založeno na pravidelném shromažďování informací o celé síti, kde ale jednotlivé uzly komunikují jen se svými sousedními zařízeními. Komunikační zprávy, které si mezi sebou posílají, jsou ve tvaru {<zdrojový uzel> <cílový uzel> <následující uzel> <celková metrika>}. Celková metrika je zde vyjádřena pomocí počtu přeskoků. Cílem předávání těchto zpráv je vytvořit jejich směrovací tabulku s tím, že při duplikování stejných cest udržuje jen tu, která má nejmenší metriku. Takto vytvořená směrovací tabulka je po nějakém čase aktualizována a sdílena se sousedními uzly [3].

Směrování vektoru vzdálenosti funguje za pomocí následujících protokolů:

- RIPv1
- RIPv2
- IGRP
- EIGRP
- BGP

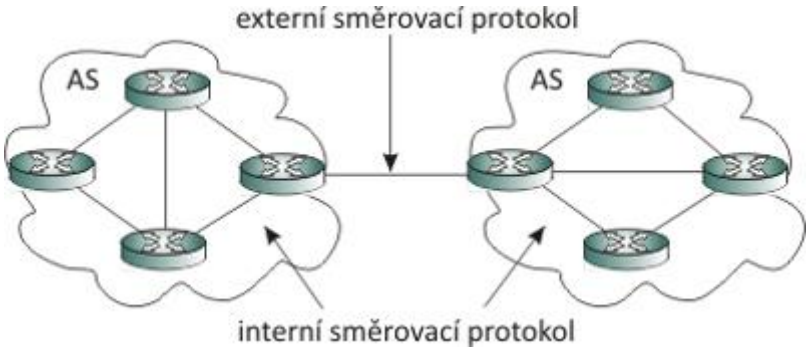
Úkolem směrovacích protokolů je, aby periodicky posílali kompletní směrovací tabulky svým sousedům. Takovéhle aktualizace sebírají šířku pásma a systémové zdroje, a tak protokoly mají některé společné vlastnosti [3]:

- Periodické aktualizace – nezávislé na změně topologie sítě, u RIP každých 30 sekund, u IGRP každých 90 sekund,
- Sousedi – používají stejný směrovací protokol. Směrovač ví jen síťové adresy svého rozhraní a adresy vzdálené sítě, které může dosáhnout přes sousedy,
- Všesměrové aktualizace – vysílány na adresu broadcastu (255.255.255.255).

3 Směrovací protokoly

3.1 Autonomní systém

Jelikož v dnešní době je internet jedna obrovská síť a bylo by takřka nemožné na jednotlivých směrovačích držet veškeré informace, byla celá síť hierarchický rozdělená na jednotlivé menší skupiny sítí – autonomní systémy (AS). Tím například může být jeden poskytovatel internetu nebo velká společnost. Autonomní systém je tedy pod jednou správou a má svoje pravidla, jako např. určený vnitřní směrovací protokol (IGP). Každý AS obdrží od organizace IANA (Internet Assigned Numbers Authority) svoje specifické 16-bitové číslo (1-65536), které ji jednoznačné poté identifikuje. Směrování uvnitř AS tedy zařizuje některý z IGP protokolů, oproti tomu ale jednotlivé AS používají pro komunikace vnější směrovací protokoly (EGP) [6, 7].



Obr. 12: Schéma dvou AS a použitých protokolů mezi nimi
Zdroj: http://access.feld.cvut.cz/storage/201001131458_fig1.jpg

3.2 Vnitřní (IGP)

3.2.1 RIPv1

Protože v raném počátku protokolu si každá společnost vytvořila svoji vlastní verzi protokolu, roku 1988 byl napsán Charlesem Hedrickem standard pro tento protokol. Routing information protocol (RIP) je prvním směrovacím protokolem vektoru vzdálenosti, který je založen na Bellman-Fordově algoritmu. Pomoci tohoto protokolu si jednotlivé směrovače rozesílají mezi sebou zprávy, kterými si navzájem říkají o případných změnách v topologii sítě [6].

Hlavní vlastnosti RIPv1 jsou [6, 8]:

- jako metriku používá počet přeskoků,
- maximální počet přeskoků může být 15, jinak je síť neplatná,

- opakování rozesílání zpráv je každých 30 sekund,
- automaticky spouští aktualizace při změně sítě,
- směrovače s RIPv1 jsou omezeny pouze na stejné masky podsítí ve stejných třídách,
- používá se pouze v malých sítích.

Zprávy rozesílané směrovači jsou zapouzdřeny do UDP segmentu. UDP segment je rozdělen na čtyři části: záhlaví linkové vrstvy, záhlaví paketu IP, záhlaví UDP datagramu a samotná zpráva RIP. V záhlaví linkové vrstvy je pouze zdrojová MAC adresa a cílová, která je rovna adrese broadcastu sítě. Záhlaví paketu IP obsahuje zdrojovou IP adresu, cílovou IP adresu, která je opět adresou broadcastu, tedy 255.255.255.255 a protokol „17“. UDP datagram zná jen zdrojový a cílový port, které jsou nastaveny na „520“. Samotná RIP zpráva pak obsahuje příkaz, který je buď „žádost“, anebo „odpověď“. Dále o jakou verzi se jedná, v tomto případě tedy RIPv1. Identifikátor skupiny adres, který pro IP je roven „2“. Předposlední položkou jsou trasy, což je IP adresa sítě a naposledy metriku, kterou zastupuje počet přeskoků. Jedna taková zpráva může znát až 25 řádek tras [6, 8].

Základní příkazy pro konfiguraci Cisco směrovače s RIP [6, 9]:

```
Router(config)#router rip      / zapnutí RIP
Router(config-router)#network <přímo připojená síť   / určení, které sítě se mají RIPu zúčastnit, je-li to víc sítí, příkaz budeme opakovat pro každou síť znovu
Router(config-router)#no network    / odebere síť
Router(config-router)#show ip route / zobrazí routovací tabulku
Router(config-router)#version <verze>     / zvolení mezi RIPv1 a RIPv2
```

3.2.2 RIPv2

Hlavním důvodem, proč vznikla druhá verze protokolu RIP je že RIPv1 nepodporoval beztřídní směrování. Kromě této výhody, ale navíc získal i informaci o adrese dalšího přeskoku, použití skupinových adres, tzv. multicastu a možnost ověřování. Naopak ke zpětné kompatibilaci s RIPv1 má stejný limit maximálního počtu přeskoků, použití časovačů, aby nedocházelo k nekonečným smyčkám a automatické aktualizace. Z důvodu ochrany před přijetím nesprávné aktualizace byla navíc ještě zavedena autentizace. Ta zajišťuje, že směrovače budou přijímat zprávy od okolí pouze v případě, že budou mít nastaveny stejně heslo [6].

3.2.3 EIGRP

EIGRP (Enhanced interior gateway routing protocol) patří do skupiny vnitřních směrovacích protokolů. Byl vyvinut společností Cisco v roce 1992 jako vylepšený IGRP protokol. Je založen na principu vektoru vzdálenosti. Hlavním důvodem proč EIGRP vznikl, bylo vytvoření beztřídního IGRP a přidání funkcí, které společně s RIP neměl, jako [6, 10]:

- spolehlivý transportní protokol RTP,
- omezené aktualizace,
- konvergentní algoritmus DUAL,
- tabulky sousedů a topologie.

V porovnání s RIP protokolem, EIGRP nepoužívá Bellman-Fordův algoritmus, ale algoritmus DUAL. Jeho hlavní předností je, že nemusí posílat pravidelné aktualizace jednotlivých směrovacích tabulek. Oproti tomu rozesílá tzv. Hello pakety, kterými kontroluje pouze dostupnost svých sousedů. Aktualizaci směrovací tabulky poté pošle jen v případě, že Hello paket zjistí nějaký rozdíl u jeho sousedů. Další výhodou EIGRP je pamatování si všech možný cest v podsíti, což následně urychlí zorientování se při změně v topologii sítě. To zajišťuje oddělená směrovací tabulka od tabulky topologie. Počet přeskoků byl navýšen na maximálně 255. Protože byl ale vyvinut firmou Cisco, funguje jen na jejich směrovačích [6, 10].

EIGRP je založeno na fungování 3 tabulek [6]:

- **sousedů** – zná všechny přímo připojené směrovače v jednom AS,
- **topologie** – kromě nejlepší známé cesty ví i o případných dalších cestách, kdyby došlo k poruše na nejlepší cestě,
- **směrovací** – obsahuje pouze cesty s nejnižší metrikou za pomoci DUAL algoritmu a předchozích dvou tabulek.

Výpočet metrik jednotlivých cest je u EIGRP poměrně složitý, protože v jeho vzorci se bere v potaz i šířka pásma, prodleva rozhraní, aktuální zátěž a spolehlivost linky. Výsledná metrika se pak spočte podle následujícího vzorce [10]:

$$\left(K_1 * \text{šířka pásma} + \frac{K_2 * \text{šířka pásma}}{256 - \text{aktuální zátěž}} + K_3 * \text{prodleva}\right) * \frac{K_5}{\text{spolehlivost linky} + K_4} \quad (1.1)$$

- $\text{šířka pásma} = \frac{10^7}{\text{šířka pásma nejpomalejší linky}}$, (1.2)

- prodleva odpovídá sumě všech prodlev v linkách,
- aktuální zátěž je z intervalu <1;255>,
- spolehlivost linky je pravděpodobnost nefunkčnosti, taktéž z intervalu <1;255>,
- koeficienty K_2, K_4 a K_5 jsou implicitně nastaveny na 0.

Základní Cisco přikazy ke konfiguraci EIGRP [6, 9]:

```
(config)# router eigrp <číslo AS>   / zapnutí EIGRP a zadání čísla AS
(config-router)# network <sít>      / určení sítě
(config-if)# bandwidth x      / nastavení šířky pásma v kilobitech
(config-router)# no network <sít>   / odebrání sítě
(config)# no router eigrp <číslo AS>   / vypnutí EIGRP
(config-router)# metric weights tos k1 k2 k3 k4 k5     / změna hodnot
koeficientů
```

3.2.4 OSPF

OSPF (Open shortest path first) je typickým příkladem protokolu typu stavu linky a jedním z nejpoužívanějších IGP (Interior gateway protocol) protokolů. Byl vyvinut v letech 1988 – 1991 pro rozsáhlejší sítě jako reakce na protokol RIP. Oproti RIPu nabízí rychlou konvergenci na změny v topologii sítě a použitelnost u mnohem větších sítí [6].

Práce OSPF protokolu začíná v okamžiku, kdy směrovač rozešle Hello pakety. Když se dva přímo propojené směrovače dohodnou na vzájemných vlastnostech, stanou se sousedy. Některé směrovače se poté mohou stát přilehlými, pokavaď se mezi nimi vytvoří užší vazby. Ty si pak mezi sebou začnou předávat pakety s aktualizacemi obsahující oznamovače LSA, které informují o stavu rozhraní nebo seznam směrovačů připojených k síti. Každý směrovač si došlé LSA uloží do databáze topologie LSDB a přepošle ji svým přilehlým směrovačům. V okamžiku, kdy se tohle stane na všech směrovačích, tak na každém bude uložená stejná LSDB. Z hotové LSDB a použití Dijkstrova algoritmu si jednotlivé směrovače zjistí své nejkratší cesty do ostatních sítí. Pokud poté nastane jakákoliv změna v topologii, směrovač, na kterém k tomu došlo, rozešle svým sousedům aktualizace a celý princip se spustí znovu až do okamžiku než opět všichni budou vědět o všem. Tento proces se však provádí jen v určité oblasti sítě a tím je zajištěno „nenáročné“ fungování ve větších sítích. Mezi jednotlivými oblastmi se pak vyměňují jenom finální informace [6, 11].

Pro správnou komunikaci mezi směrovači používá OSPF 5 druhů paketů [6, 12]:

- **Hello** – prvotní paket, který vytváří vztahy se sousedy,
- **DBD** – stručný výpis LSDB k ověření a synchronizaci na přijímacím routeru,
- **LSR** – žádost o další řádku databáze,

- **LSU –** aktualizace s až 11 různými typy LSA, např.: směrovač, síť, agregace, externí AS, Multicast OSPF, externí parametry protokolu BGA,
- **LSAck** – potvrzení přijetí LSU.

U OSPF je metrika zastupována cenou. Čím nižší bude cena, tím více bude dané rozhraní upřednostňováno. Může nabývat hodnoty 1-65535. Metrika se zde vypočte podle následujícího vzorce [11]:

$$\text{cena} = \frac{10^8}{\text{šířka pásma v bitech za sekundu}} \tag{2.3}$$

Cisco příkazy pro konfiguraci OSPF [6, 9]:

```
(config)# router ospf <číslo procesu>  / zapnutí OSFP s ID procesu
(config-router)# network <síť> <pseudomaska> area <area ID>   / určí, která rozhraní mají být vložena do oblasti „area ID", pro páteřní oblast je „0"
(config-router)#  log-adjacency-changes  detail       / zapnutí posílání logovacích zpráv při změně stavu mezi sousedy
(config-if)#ip OSF priority <priorita>   / změní prioritu rozhraní na 0-255
(config)#interface  serial  0/0      / změní z režimu směrovače na režim konfigurace rozhraní
(config-if)#bandwith <šířka pásma>   / změna ceny linky podle šířky pásma
(config-if)#ip OSF cost <cena>   / přímé nastavení ceny linky
(config-router)#area <area ID> authentication   / zapnutí autentizac
(config-router)#ip  OSF  message-digest-key  1  md5  7  <heslo>    / nastavení hesla pro autentizaci
(config-if)#ip  OSF  hello  interval  <čas>     / nastavení intervalu pro posílání hello paketů
```

3.2.5 IS-IS

Protokol IS-IS (Intermediate System to Intermediate System) patří do skupiny vnitřních směrovacích protokolů typu stavu linky. Stejně jako protokol OSPF využívá Dijkstrova algoritmu na určení nejkratší možné cesty v síti a podle zjištěných výsledků si sestaví topologii sítě. Mezi jeho hlavní vlastnosti patří jeho hierarchické směrování a uspořádání routerů, zjišťování stavu sousedů pomocí LSP paketu, rychlá konvergence, škálovatelnost, flexibilní nastavení časovače a beztřídní směrování, čímž umožňuje podporu VLSM, které dovoluje rozdělit síť na různě velké podsítě [13].

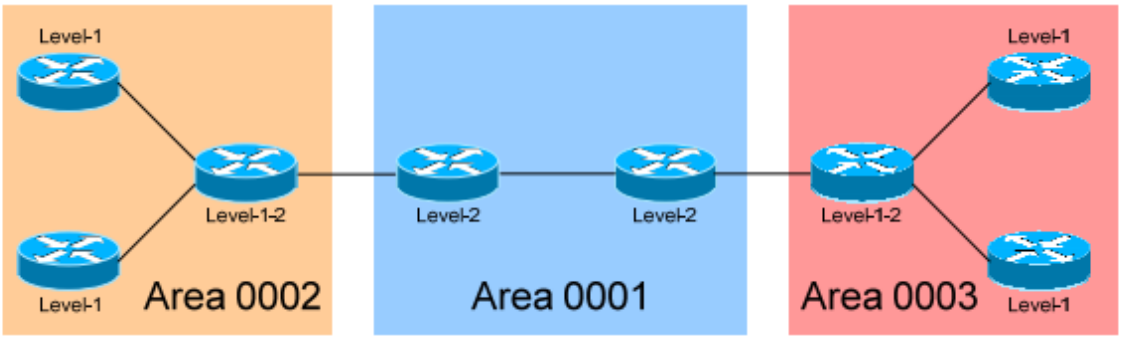
Opět s podobností protokolu OSPF, IS-IS používá jako metriku cenu linky. Může ale mít navíc ještě další volitelné metriky: metriku zpoždění, nákladovou metriku a metriku chyby. Ty ale nejsou podporovány například směrovači Cisco. Na rozdíl od ostatních protokolů se zde cena linky nepočítá pomocí vzorce, ale cenou rozhraní. Ta může být 1-63, kdy výchozí

hodnota je nastavena na 10. Celková metrika bude tedy dána součtem všech rozhraní, které jsou na cestě k cíli. Je ale omezena maximální hodnotou 1023. Ta ale v případě větších sítí může být nedostatečná. Tento problém vyřešila firma Cisco IOS Software zavedením 24bitového metrického pole. Tím se maximální hodnota zvýšila na 4 261 412 864. Naopak není vůbec nijak limitován počtem přeskoků [13, 14].

Hierarchie směrovačů, které používají protokol IS-IS je rozdělena do několika úrovní na které pracují a to následovně: [13, 15]:

- Level 1 – směrovače komunikující mezi sebou pouze v jedné určité oblasti,
- Level 2 – routery směrující mezi více oblastmi, ale ve stejné doméně,
- Level 1-2 – routery na pomezí Level 1 a Level 2
- Level 3 – směrovače mezi jednotlivými doménami.

Jak vidíme na obr. 13, směrovače na jednotlivých úrovních mohou komunikovat pouze s úrovněmi o jednu výš, nebo níž. Směrovač na levelu 1 tedy nemůže komunikovat se směrovačem levelu 2 a výš [13].



Obr. 13: Hiearchie routerů s IS-IS protokolem (převzato z [13]

Stejně jako u OSPF, tak i zde komunikace začíná rozesíláním Hello paketů. Ty jsou zde ale třech druhů, podle toho s kým si je přeposílají [13]:

- **IIH (IS-IS Hello)** – výměna pouze mezi routery,
- **ESH (ES Hello)** – posílána z koncového zařízení k objevení routerů,
- **ISH (IS Hello)** - z routerů, které oznamují svoji přítomnost koncovým zařízení.

V dalším kroku si sousední směrovače začínají vyměňovat LSA zprávy k vytvoření topologických tabulek. Stejně tak IS-IS používá podobné zprávy, které se ovšem nazývají LSP (Link-State pakety). Podle úrovně směrovače poté posílají buď LSP levelu 1, nebo 2. Směrovače na levelu 1 tedy sdílejí LSP a vytvářejí topologické tabulky pouze s ostatními směrovači na prvním levelu. Pokud ovšem směrovač má posílat do jiné oblasti, musí nejdřív

Směrovač na levelu 1-2 nastavit v LSP „Attach“. Routery levelů 1-2 tedy znají jak topologii prvního levelu, tak i druhého. Na rozdíl od protokolu OSPF, jsou zde prováděny aktualizace každých 15 minut. Naopak proti tomu routery na druhé úrovni si staví tabulku topologie opět jen s routery na stejné úrovni. Díky tomu pomezní směrovače na levelu 1-2 budou sdílet topologické tabulky z obou dvou okolních úrovní [13].

Základní Cisco příkazy pro konfiguraci IS-IS [13]:

```
(config)#router isis   / zapnutí IS-IS
(config-router)#net <sít>   / nastavení adresy směrovače
(config-router)#is-type level-1     / určení úrovně směrovače, případně „level-1-2“, nebo „level-2-only“
(config)# interface fa0/0 -> (config-if)# ip router isis   / povolení IS-IS na rozhraní a přidání do směrovací tabulky
(config)# interface e0/0 -> (config-if)# isis metric <metrika>     / nastavení metriky rozhraní
```

3.3 Vnější (EGP)

3.3.1 BGP

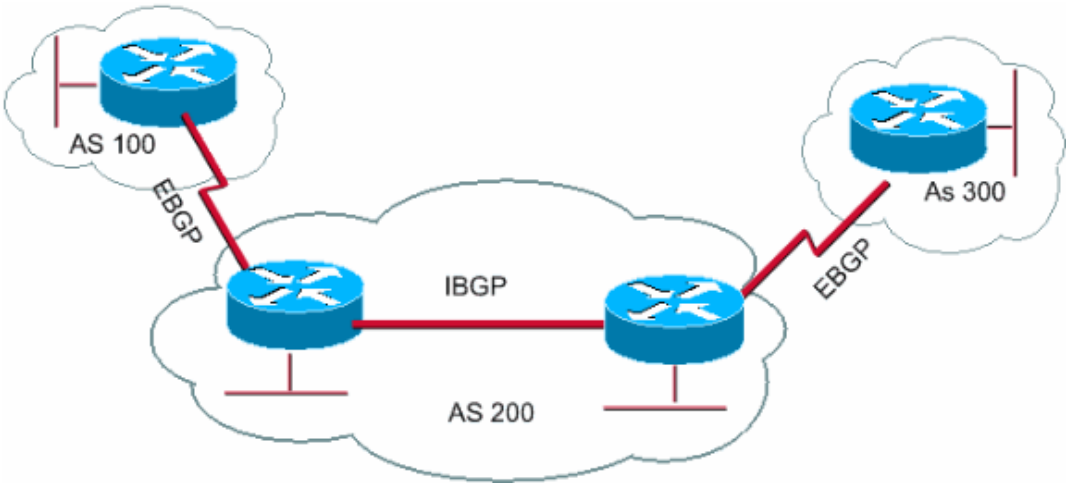
Protokol BGP patří v dnešní době prakticky mezi jediný používaný vnější směrovací protokol mezi autonomními systémy. Nedá se ale u něj jednoznačně říct, jestli patří do skupiny protokolů stavu linky, nebo vektoru vzdálenosti, a proto se dá nazvat jako path-vector protokol [7]. Svůj název dostal podle svojí činnosti, kdy přeposílá zprávy skrze hraniční routery, anglicky border gateway, tedy Border Gateway protokol – BGP. První verze BGP byla definována v roce 1989, ale v současnosti se od roku 2006 používá jeho čtvrtá verze BGP-4, která je definována v RFC 4271 Pro spolehlivý přenos paketů jako jediný používá TCP na portu 179 a podporuje beztřídní adresování CIDR/VLSM [16].

BGP by měl být používán v následujících případech [17]:

- Vícenásobně připojení k AS přes různé poskytovatele,
- AS je připojen do více AS,
- Ovladatelný datový tok skrz AS.

Prvním krokem při práci protokolu je přijmutí směrovacího updatu, které zajistí TCP spojení se sousedním směrovačem. Směrovače s BGP jsou zde nazývány jako „speakers“ a po vytvoření spojení se sousedem jako „peers“. Při vytvoření vztahu se sousedem vznikne jedna ze dvou vazeb, eBGP, nebo iBGP. Vytvoření jednoho z těchto spojení závisí na umístění spojovaných sousedů. Jestliže jsou sousedi ve stejném AS, vznikne iBGP, naopak jestliže jsou

v různých AS, jde o eBGP. Následně dojde k vytvoření BGP tabulky do které jsou vkládány veškeré příchozí aktualizace. Z těchto došlých zpráv se poté vytvoří cesty podle zvolených kritérii. Ty mohou být například jako vyřazení cesty s nedostupným parametrem „další skok“, preferování nejvyšší „Local_Preference“, nejkratší cesta a mnoho dalších. Nejlepší vybrané cesty jsou pak rozesílány svým sousedům a jsou zaznamenány ve směrovacích tabulkách. Ty zde nejsou pravidelně aktualizovány, ale jednotlivé směrovače si opakovaně zkouší dostupnost svých sousedů a až při případné nedostupnosti upraví informace a přepošle je ostatním svým sousedům [7, 18].



Obr. 14: Topologie s rozdělením na eBGP a iBGP
Zdroj: http://www.cisco.com/c/dam/en/us/support/docs/ip/border-gateway-protocol-bgp/26634-bgp-toc1.gif

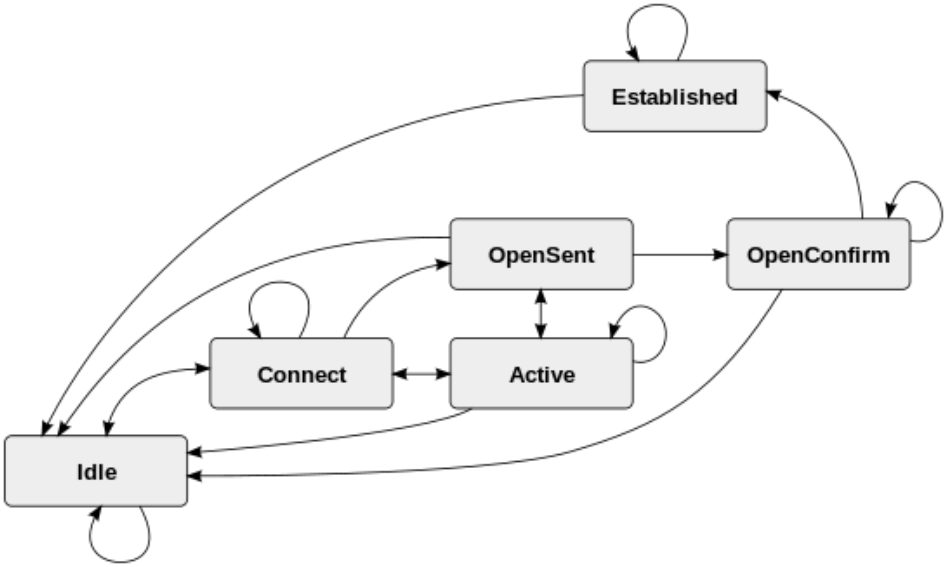
Protokol BGP vytváří a udržuje jednotlivá spojení pouze za pomocí 4 druhů zpráv [7, 17]:

- **OPEN** – zařizuje spojení mezi sousedy a informuje o číslech AS, do kterých patří, verzi BGP a identifikátor směrovače,
- **UPDATE** – obsahuje směrovací informace, které jsou sestaveny z dvojic prefix a délka prefixu, případně změny směrovacích cest,
- **KEEPALIVE** – zpráva pro ověřování dostupnosti linek mezi sousedy. Většinou opakována každých 60 vteřin. Pokud nedostane žádnou odpověď po dobu nastavenou jako „HoldTime“, většinou 180 sekund, spojení se považuje za mrtvé,
- **NOTIFICATION** – značí chybu při práci protokolu, po jejím vyslání se přeruší spojení mezi sousedy

Naopak jednotlivé směrovače si při tomto procesu projdou několika stavy, kterým se říká Finite-State Machine (FSM) [7, 17]:

- **Idle** – počáteční stav, při kterém se připravuje na vysílání a zároveň odmítá jiná spojení,

- **Connect** - vytváří TCP spojení a odesílá OPEN zprávu. V případě úspěšného pokusu přejde do stavu OpenState. V případě neúspěchu přejde do Active stavu,
- **Active** – zkouší dále vytvořit spojení a v případě navázání ho přejde do OpenSent stavu. Při neúspěchu čeká na vypršení časovače ConnectRetry a po jeho uplynutí se vrátí do stavu Connect,
- **OpenSent** – čeká na odpověď od souseda a při přijetí vyzkouší, jestli je platná, odešle KEEPALIVE,
- **OpenConfirm** - čeká na KEEPALIVE od souseda,
- **Established** - bylo vytvořeno oboustranné spojení a směrovače si začnou vyměňovat UPDATE a KEEPALIVE zprávy.



Obr. 15: Schéma FSM
Zdroj: http://upload.wikimedia.org/wikipedia/commons/thumb/a/a8/BGP_FSM.svg/549px-BGP_FSM.svg.png

Oproti předchozím IGP protokolům, BGP nemá jednoznačně danou metriku. Ta je zde zastoupena koeficientem, v kterém se bere v potaz počet AS k cíli, rychlost linky a jednotlivě přiřazené jim priority a pravidla, které si určují administrátoři daných AS.

Při hledání nejlepších a nejkratších cest se u BGP využívá jeho parametrů. Ty umožňují administrátorům nastavovat jednotlivá pravidla a preferování cest při směrování. Atributy jsou podle povinnosti definování jich rozděleny do 4 skupin [7]:

- **Well-known mandatory** - povinné ke každé cestě a každá implementace BGP je musí znát,
- **Well-known discretionary** - nepovinné, ale každá implementace BGP je musí stejně znát,
- **Optional transitive** - BGP jim nemusí rozumět, ale i tak je předá dál,

- **Optional nontransitive** - opět jim BGP nemusí rozumět, ale v tomhle případě ho dál nepředá.

Základními atributy jsou [16]:

- weight – povinný lokální atribut pro upřednostnění cesty,
- AS_Path - Well-known mandatory, řada čísel AS, které vedou k cíli,
- Next hop - Well-known mandatory, další AS v řadě,
- origin - Well-known mandatory, značí, jakým způsobem se dozvěděl o směrovací cestě, může být „IGP“, EGP“, nebo „Incomplete“,
- local preference - Well-known discretionary**,** výběr opuštění cesty z AS,
- Multiple Exit Discriminator (MED) - Optional nontransitive**,** informuje sousedy o preferované cestě do AS při více možnostech.

Při sestavování celkové topologie se dají jednotlivé autonomní systémy rozdělit do třech základních skupin [18]:

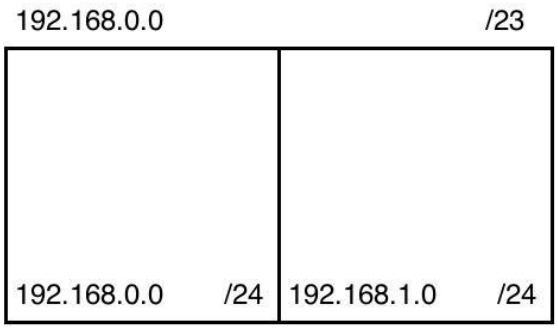
- **tranzitní sít,** přes kterou jsou pakety jenom přeposílány, má několik vnějších i vnitřních sousedů a zná celkovou směrovací tabulku,
- **AS s více výstupy**, které přeposílají jen zdrojové, nebo cílové pakety a je možnost se k nim dostat přes více cest,
- **jednoduché AS**, které mají pouze jednu přístupovou linku.

Základní Cisco příkazy pro konfiguraci BGP [7, 17]:

```
(config)#router bgp <číslo AS>   / zapnutí BGP
(config-router)#neighbor <adresa> remote-as <číslo AS>    / určení sousedů pro vytvoření spojení
(config-router)#neighbor <adresa> next-hop-self     / nastaví adresu jako next hop
(config-router)#bgp always-compare-med      / vynucené porovnávání metrik s ostatních AS
(config-router)# neighbor <adresa> timers <keepalive> <hold-time>      / nastaví časovače pro Keepalive a Holt-time v sekundách
```

4 Příklad na konfiguraci protokolů

Využívání a konfiguraci směrovacích protokolů ukážeme na návrhu topologie sítě. Při návrhu topologie budeme vycházet z toho, že budou tři autonomní systémy a jeden z nich bude zobrazen podrobněji, kdy v něm budou další dva routery a koncové počítače. Jednotlivé autonomní systémy budou označeny jako AS10, AS20 a AS30. AS10 a AS30 jsou společně propojeny přes AS20. V AS10 dále bude vytvořena podsíť, ve které budou umístěny do hvězdy směrovače, ze kterých už vycházejí jen switche, do kterých jsou připojeny počítače. Návrh počítá, že v sítích bude moci být až 254 zařízení, proto jsou u obou sítí masky /24. První síť mezi AS je tedy pod IP adresou 10.10.10.0 a druhá síť je následující možná, tedy 10.10.11.0. K vzájemnému propojení směrovačů R1, R2 a R3 v AS10 pro každou síť stačí mít pouze dvě IP adresy na jejich spojení. Maska zde bude tedy použita /30. IP adresy těchto tří sítí jsem zvolil po sobě následující 195.1.1.0, 195.1.1.4 a 195.1.1.8. V koncových podsítích vycházíme z adresního schématu viz. obr. 16, kde IP adresu 192.168.0.0/23 rozdělíme pro začátek na dvě další podsítě. V nich bude moci být maximálně 254 zařízení, maska bude tedy /24. Adresy jejich sítí budou 192.168.0.0 a 192.168.1.0 s maskou /24. Výsledný návrh topologie sítě i s IP adresami a maskami všech sítí, rozhraní i koncových počítačů je na obr. 17.



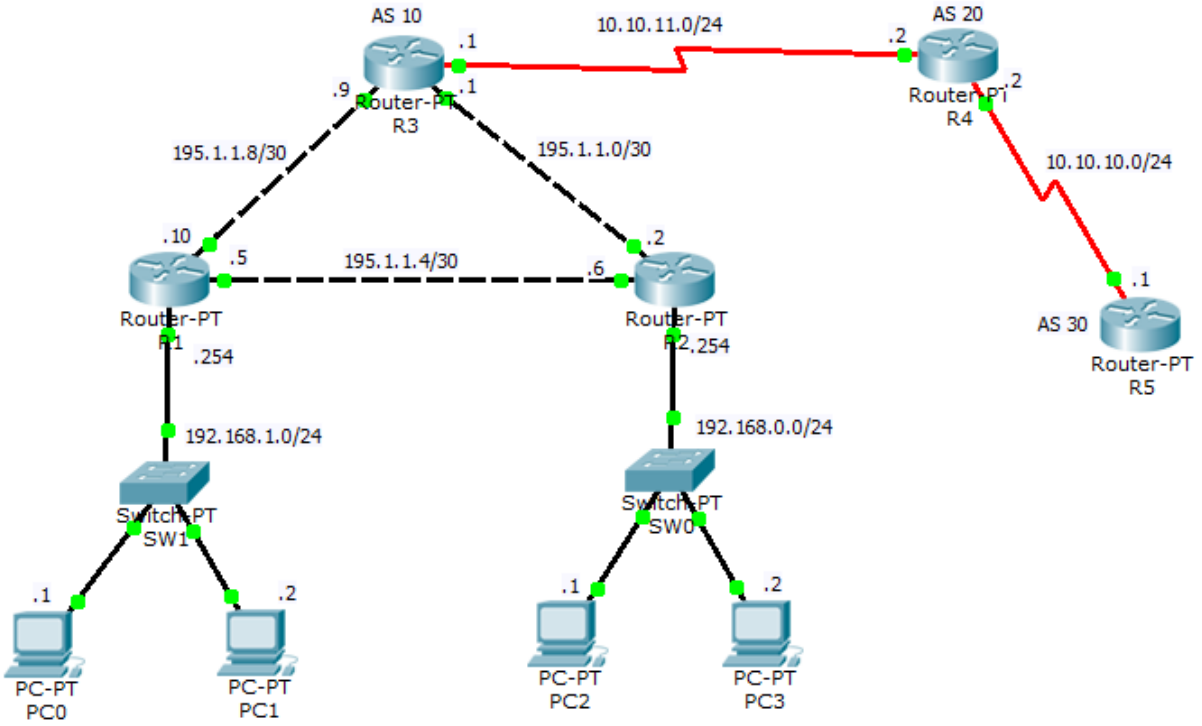
Obr. 16: Návrh adresního schématu

Při konfigurování sítě se nejprve nastavili všechny IP adresy na všech rozhraních. Směrování mezi autonomními systémy je zařízeno pomocí vnějšího směrovacího protokolu BGP . Ten se nastavil na směrovačích R3,R4 a R5 pomocí příkazů, zde například pro R3:

```
(config)#router bgp 10
(config-router)#neighbor 10.10.11.2 remote-as 20
(config-router)#network 195.1.1.8 mask 255.255.255.252
```

Uvnitř AS se komunikuje podle vnitřního protokolu RIP. Nastavení se provedlo na směrovačích R1,R2,R3 podle následujících příkazů, zde pro router R1:

```
(config)#router rip
(config-router)#network 192.168.1.0
(config-router)#network 195.1.1.8
(config-router)#no auto-summary
```



Obr. 17: Topologie sítě k příkladu směrovacích protokolů

Úplné konfigurace všech směrovačů jsou viděny v příloze. Po provedení nastavení všech zařízení si můžeme výsledek zkontrolovat po vypsání tabulek na směrovačích pomocí příkazů „show ip bgp“ a „show ip route“. Pro ukázku jsou příkazy vypsány ze směrovače R4. Při použití příkazu „show ip bgp“, obr. 18, uvidíme, které sítě jsou dostupné a zároveň i které jsou nejlepší k použití směrovaní. Položka Next hop říká, přes které rozhraní je daná síť dostupná. Druhým příkazem „show ip route“ se vypíše směrovací tabulka na obr. 19. Z ní zjistíme dostupné sítě z daného routeru, ale také v jakém vztahu jsou ke směrovači připojeny. Na R4 například vidíme dvě přímo připojené sítě, to jsou sítě, které spojují autonomní systémy, tedy 10.10.10.0 a 10.10.11.0. Poté také dvě sítě, ke kterým se dostaneme pomocí BGP, tedy 195.1.1.0 a 195.1.1.8.

```
BGP table version is 5, local router ID is 10.10.11.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 195.1.1.0/30     10.10.11.1               0     0      0 10 i
*> 195.1.1.8/30     10.10.11.1               0     0      0 10 i
```

Obr. 18: Výpis příkazu "show ip bgp" na R4

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 2 subnets
C       10.10.10.0 is directly connected, Serial3/0
C       10.10.11.0 is directly connected, Serial2/0
     195.1.1.0/30 is subnetted, 2 subnets
B       195.1.1.0 [20/0] via 10.10.11.1, 02:03:30
B       195.1.1.8 [20/0] via 10.10.11.1, 02:03:30
```

Obr. 19: Výpis směrovací tabulky na R4

Závěr

Cílem této práce bylo popsání jednotlivých druhů směrování, směrovacích protokolů a následná ukázka jejich konfigurace. K pochopení dynamického směrování byly nejdříve vysvětlené základní pojmy ke grafům, ale především Dijkstrův a Bellman-Fordův algoritmus k nalezení nejlepší cesty při průchodu grafem. Na jejich principech jsou založeny následně popsané druhy dynamického směrovaní, tedy algoritmus stavu linky a vektoru vzdálenosti. Statické směrování zde bylo zmíněné jenom okrajově, neboť není tolik používané ve velkých sítích, ale především proto, že se hlavně zabývalo dynamickým směrováním. Ve třetí části byl vysvětlen pojem autonomního systému, který dále určuje, jaké směrovací protokoly budeme chtít používat. Ke vztahu k autonomnímu systému byly jednotlivé protokoly rozděleny na vnitřní a vnější a každý z nich více popsán. V poslední části je praktická ukázka konfigurace směrovačů uvnitř jednoho autonomního systému, ale i mezi jednotlivými autonomními systémy. Zde byly pro zprovoznění sítě použity protokoly RIP a BGP.

Seznam literatury a informačních zdrojů

[1] ČERNÝ, Jakub. Reprezentace grafu [online]. Praha, 2010 [cit. 2014-03-10]. Dostupné z: http://kam.mff.cuni.cz/~kuba/ka/reprezentace_grafu.pdf

[2] ČADA, Roman, Tomáš KAISER a Zdeněk RYJÁČEK. *Diskrétní matematika*. 1. vyd. Plzeň: Západočeská univerzita v Plzni, 2004, 170 s. ISBN 80-708-2939-7.

[3] VUSKOVIC, Marco. Routing protocols [online]. San Diego, 2004. Dostupné z: http://medusa.sdsu.edu/network/CS576/Lectures/ch13_RIP.pdf

[4] EFRAT, Alon. Bellman-Ford Algorithm [online]. Arizona, 2009 [cit. 2014-03-03]. Dostupné z: http://www.cs.arizona.edu/classes/cs545/fall09/ShortestPath2.prn.pdf

[5] LOMNICKÝ, Marek a Vladimír VESELÝ. *Směrování a směrovací protokoly* [online]. Brno, 2007 [cit. 2014-03-23]. Dostupné z: http://netacad.fit.vutbr.cz/texty/ccna-moduly/ccna2-6.pdf

[6] PÁV, Miroslav, Jan SYŘÍNEK a Jana HOŠKOVÁ. *CCNA Exploration - Směrování, koncepce a protokoly* [online]. Plzeň, 2011. Dostupné z: http://jonatan.spse.pilsedu.cz/CISCO/CCNA_Exploration_2.pdf

[7] GRYGAREK, Petr. *Směrovací protokol BGP* [online]. Ostrava [cit. 2014-04-22]. Dostupné z:http://www.cs.vsb.cz/grygarek/SPS/lect/BGP/BGP.html

[8] HEDRICK, Charles. *Routing Information Protocol* [online]. New Jersey, 1988 [cit. 2014-04-20]. Dostupné z: http://www.ietf.org/rfc/rfc1058.txt

[9] Příkazy Cisco IOS [online]. [cit. 2014-04-25]. Dostupné z:http://cs.wikibooks.org/wiki/P%C5%99%C3%ADkazy_Cisco_IOS#EIGRP_sm.C4.9Brov.C3.A1n.C3.AD

[10] Směrovací protokoly "Distance Vector" [online]. [cit. 2014-04-22]. Dostupné z: http://www.neoo72.ic.cz/doc/pos/15_rip.pdf

[11] GRYGAREK, Petr. *Směrovací protokol OSPF* [online]. Ostrava, 2005 [cit. 2014-04-22]. Dostupné z:http://www.cs.vsb.cz/grygarek/SPS/lect/OSPF/ospf.html

[12] KMONÍČEK, Tomáš. *Analýza principů IGP a EGP routovacích protokolů* [online]. Pardubice, 2013 [cit. 2014-04-22]. Dostupné z:http://dspace.upce.cz/bitstream/10195/51622/2/KmonicekT_AnalyzaPrincipu_JH_2013.pdf

[13] BALCHUNAS, Aaron. Router Alley: IS-IS. [online]. 2007 [cit. 2014-04-27]. Dostupné z: http://www.routeralley.com/ra/docs/isis.pdf

[14] Cisco: Intermediate System-to-Intermediate System Protocol. [online]. [cit. 2014-04-27]. Dostupné z:http://www.cisco.com/en/US/products/ps6599/products_white_paper09186a00800a3e6f.shtml#wp39006

[15] BOUŠKA, Petr. Cisco Routing 4: IS-IS - Intermediate System to Intermediate System. [online]. [cit. 2014-04-25]. Dostupné z: http://www.samuraj-cz.com/clanek/cisco-routing-4-is-is-intermediate-system-to-intermediate-system/

[16] BOUŠKA, Petr. Cisco routing: BGP - Border Gateway Protocol. [online]. 2009 [cit. 2014-04-28]. Dostupné z: http://www.samuraj-cz.com/clanek/cisco-routing-5-bgp-border-gateway-protocol/

[17] BALCHUNAS, Aaron. *Router Alley: Border Gateway Protocol* [online]. 2007 [cit. 2014-04-28]. Dostupné z:http://www.routeralley.com/ra/docs/bgp.pdf

[18] NĚMEČEK, Vladimír. *Border Gateway Protocol* [online]. Liberec [cit. 2014-04-28]. Dostupné z:http://www.nti.tul.cz/~satrapa/vyuka/site/rs/Pr3_BGP_basic.pdf

Přílohy

Konfigurace R1:

```
Router>enable
Router#configure terminal
Router(config)#hostname R1
R1(config)#
R1(config)#interface FastEthernet0/0
R1(config-if)#no shutdown
R1(config-if)#ip address 192.168.1.254 255.255.255.0
R1(config-if)#exit

R1(config)#interface FastEthernet1/0
R1(config-if)#no shutdown
R1(config-if)#ip address 195.1.1.5 255.255.255.252
R1(config-if)#exit
R1(config)#interface FastEthernet6/0
R1(config-if)#no shutdown

R1(config-if)#ip address 195.1.1.10 255.255.255.252
R1(config-if)#exit

R1(config)#router rip
R1(config-router)#network 192.168.1.0
R1(config-router)#network 195.1.1.8
R1(config-router)#network 195.1.1.4
R1(config-router)#no auto-summary
R1(config-router)#exit
```

Konfigurace R2:

```
Router>enable
Router#configure terminal
Router(config)#hostname R2

R2(config)#interface FastEthernet0/0
R2(config-if)#no shutdown
R2(config-if)#ip address 192.168.0.254 255.255.255.0
R2(config-if)#exit

R2(config)#interface FastEthernet1/0
R2(config-if)#no shutdown
R2(config-if)#ip address 195.1.1.6 255.255.255.252
R2(config-if)#exit

R2(config)#interface FastEthernet6/0
R2(config-if)#no shutdown
R2(config-if)#ip address 195.1.1.2 255.255.255.252
R2(config-if)#exit

R2(config)#router rip
R2(config-router)#network 192.168.0.0
R2(config-router)#network 195.1.1.0
R2(config-router)#network 195.1.1.4
R2(config-router)#no auto-summary
R2(config-router)#
```

Konfigurace R3:

```
Router>enable
Router#configure terminal
Router(config)#hostname R3

R3(config)#interface FastEthernet0/0
R3(config-if)#no shutdown
R3(config-if)#ip address 195.1.1.9 255.255.255.252
R3(config-if)#exit

R3(config)#interface FastEthernet1/0
R3(config-if)#no shutdown
R3(config-if)#ip address 195.1.1.1 255.255.255.252
R3(config-if)#exit

R3(config)#router rip
R3(config-router)#network 195.1.1.8
R3(config-router)#network 195.1.1.0
R3(config-router)#no auto-summary
R3(config-router)#
R3(config-router)#exit

R3(config)#interface Serial2/0
R3(config-if)#no shutdown
R3(config-if)#ip address 10.10.11.1 255.255.255.0

R3(config-if)#exit
R3(config)#router bgp 10
R3(config-router)#neighbor 10.10.11.2 remote-as 20
R3(config-router)#%BGP-5-ADJCHANGE: neighbor 10.10.11.2 Up

R3(config-router)#network 195.1.1.8 mask 255.255.255.252
R3(config-router)#network 195.1.1.0 mask 255.255.255.252
R3(config-router)#exit
R3(config)#exit
```

Konfigurace R4:

```
Router>enable
Router#configure terminal
Router(config)#hostname R4

R4(config)#interface Serial2/0
R4(config-if)#no shutdown
R4(config-if)#ip address 10.10.11.2 255.255.255.0
R4(config-if)#exit

R4(config)#interface Serial3/0
R4(config-if)#no shutdown
R4(config-if)#ip address 10.10.10.2 255.255.255.0
R4(config-if)#exit

R4(config)#router bgp 20
R4(config-router)#neighbor 10.10.11.1 remote-as 10
R4(config-router)#%BGP-5-ADJCHANGE: neighbor 10.10.11.1 Up
```

```
R4(config-router)#neighbor 10.10.10.1 remote-as 30
R4(config-router)#%BGP-5-ADJCHANGE: neighbor 10.10.10.1 Up
R4(config-router)#exit
R4(config)#exit
```

Konfigurace R5:

```
Router>enable
Router#configure terminal
Router(config)#hostname R5

R5(config)#interface Serial3/0
R5(config-if)#no shutdown
R5(config-if)#ip address 10.10.10.1 255.255.255.0
R5(config-if)#exit

R5(config)#router bgp 30
R5(config-router)#neighbor 10.10.10.2 remote-as 20
R5(config-router)#%BGP-5-ADJCHANGE: neighbor 10.10.10.2 Up
R5(config-router)#exit
R5(config)#exit
```