PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

DIPLOMOVÁ PRÁCE

Snižování doby konvergence protokolu BGP

2011 Bc. Adam Hanko

Anotace

Práce seznamuje s vlastnostmi směrovacího protokolu BGP-4. Následně je zde popsán problém konvergence a způsoby jeho řešení. Pro urychlení konvergence BGP protokolu (snížení doby konvergence) bylo navrženo několik metod, které jsme implementovali do simulačního prostředí SSFNet. Nakonec je provedeno srovnání časové a komunikační složitosti těchto metod navzájem a s protokolem BGP-4 na základě získaných výsledků ze simulací v různě hustých sítích.

Rád bych na tomto místě poděkoval Doc. Ing. Lence Carr-Motyčkové, CSc. - své vedoucí diplomové práce za obětavou spolupráci i za čas, který mi věnovala při konzultacích, a svým rodičům za podporu po celou dobu mého studia.

Obsah

Seznam obrázků

Seznam tabulek

1. Úvod

1.1. Motivace a cíle

Celý Internet je hierarchicky rozdělen na větší celky, které zahrnují určitou část celosvětové sítě. Tyto části jsou označovány jako autonomní systémy. Každý autonomní systém je spravován samostatně nezávisle na ostatních autonomních systémech. Pro směrování mezi nimi se v dnešní době používá výhradně směrovací protokol BGP-4 (Border Gateway Protocol) [4].

V několika studiích [9] a [10] bylo prokázáno, že výkon směrovacího protokolu BGP-4 se mění v závislosti na typu topologie a velikosti sítě. Pokud se stane nějaká část sítě najednou nedostupná, trvá určitý čas než se zbytek celosvětové sítě přizpůsobí k této situaci. Z důvodu nestability jednoho síťového připojení dochází ke střídavému přepínání se (oscilaci) z jednoho síťového připojení na jiné, které má v konečném důsledku velký vliv na stabilitu připojení k Internetu. Z těchto nestabilních stavů se směrovače snaží co nejrychleji stabilizovat (zkonvergovat) své směrovací tabulky do konzistentního stavu, který počítá s těmito topologickými změnami.

Během stabilizace sítě je silně omezena dostupnost k sítím těch autonomních systémům, ve kterých došlo ke změnám v topologii. Někdy může nastat situace, že protokol BGP z důvodů především nastavených směrovacích politik v jednotlivých autonomních systémech, nemusí konvergovat.

Bylo objeveno několik způsobů řešení, jak tento problém konvergence řešit - staticky a dynamicky. V Internetu z důvodu rozsáhlosti topologie sítě se používá dynamické řešení. Příkladem dynamického řešení je metoda Route Flap Damping [25], která měla za cíl eliminovat oscilace, a zabránit tak zbytečně častým změnám ve směrovacích tabulkách. V pozdějších studiích se ukázalo, tato metoda oscilace neeliminuje úplně, ale pouze je zpomalí.

Dále se hledaly metody, kterými by šlo stabilizaci sítě urychlit a snížit tak tzv. dobu konvergence, protože kromě nestability a nedostupnosti některých částí sítě v síti putuje větší množství zpráv s informacemi, kterými lze směrovací tabulky v jednotlivých uzlech stabilizovat a představují určitou zátěž na síťový provoz a systémové prostředky směrovačů.

Tyto metody pro snížení doby konvergence protokolu BGP mají za cíl za co nejkratší dobu stabilizovat síť (zkonvergovat) po změně v topologii sítě. Patří sem především metody Ghost-Flushing a Ghost-Buster, popsané v [19]. Další podobnou a spíše teoretickou metodou je metoda konzistentních pravidel (Consistency Assertions) [20] a metoda určující původ změny (Route Cause Notification) [21].

Podrobné popsání teoretických základů těchto metod, jejich implementace v simulačním prostředí SSFNet [29] a měření a vzájemné srovnání jejich výkonu z hlediska časové a komunikační složitosti je cílem této diplomové práce.

1.2. Rozdělení kapitol

Kapitola 2. se nejdříve zabývá obecně směrováním a následně specifickými vlastnostmi směrovacího protokolu BGP-4, které jsou důležité a podstatné pro popis problému konvergence protokolu BGP.

Kapitola 3. seznamuje čtenáře s problémem konvergence, o jejích příčinách a způsobech řešení. Je zde vypracován přehled studií, analýz a měření, které se touto problematikou přibližně od roku 1996 až po současnost zabývaly. Zároveň nechybí ani přehled událostí, které v novodobé historii Internetu měly vliv na stabilitu směrování a dostupnost některých známých sítí vzhledem k celosvětové síti. Dále jsou teoreticky popsány metody, které snižují dobu konvergence protokolu BGP jako je Ghost-Flushing, Ghost-Buster, metoda konzistentních pravidel (Consistency Assertions) nebo metoda určující původ změny (Route Cause Notification).

Kapitola 4. poskytuje základní popis simulačního nástroje SSFNet [29] a jak jej spustit. Dále je zde popsána implementace všech metod pro snížení doby konvergence a způsob, jakým je aktivovat a použít v tomto simulačním prostředí.

V poslední kapitole 5. jsou popsány provedené simulace výše uvedených metod pro snížení doby konvergence v různých topologiích sítí, které se proměnlivě liší hustotou hran, v tomto simulačním prostředí. Nakonec jsou dle dosažených výsledků srovnány jednotlivé metody mezi sebou a protokolem BGP.

2. Směrovací protokol BGP-4

2.1. Směrování

Směrování je proces, který má za cíl nalézt v síti uzlů cestu z jednoho uzlu do jiného uzlu tak, že se bude snažit najít optimální cestu podle předem zvolené metriky např. vzdálenosti, časovému zpoždění atd. Zpravidla se směrování nezabývá celou cestou paketu od zdroje k cíli, ale řeší vždy jen jeden krok - tzn. kterému dalšímu sousednímu uzlu (označuje se pojmem next-hop) předat data dál tak, aby se data nakonec postupně dostaly do cíle. Sousední uzel se sám rozhodne jak s daty naloží a kam je dál pošle.

Tuto činnost provádějí směrovače (routery), které si sestavují a používají směrovací (routovací) tabulky k uchování vypočtených cest a k výpočtu nových cest. Směrovací tabulka se skládá ze dvou části:

- z cílové adresy - jde o adresu cíle (destinace), kam se mají data posílat. Může se jednat o adresu individuálního počítače, častěji však je cíl definován prefixem (začátkem adresy). Prefix mívá podobu 158.194.0.0/16.
- z informace, zda předat data přímo adresátovi (přímé směrování) nebo zda je předat některému ze sousedních uzlů (nepřímé směrování).

Směrovače vytvářejí a udržují směrovací tabulku pomocí směrovacího algoritmu. Pokud pro určitý směrovací algoritmus jsou definována přesná pravidla komunikace mezi směrovači a formáty zpráv obsahující směrovací (a případně i stavové) informace, vznikne směrovací protokol.

Směrování se rozděluje na statické a dynamické. Při statickém směrování se směrovací tabulky nemění (jsou nastaveny napevno), zatímco dynamické směrování průběžně reaguje na změny v síťové topologii a přizpůsobuje jim směrovací tabulky. Nejvýznamnějšími představiteli dynamických směrovacích protokolů jsou RIP [47], OSPF [46] a BGP [4].

Směrovací protokoly se rozdělují na protokoly pracující s vektory vzdáleností (distance vector) a na linkově stavové (link-state).

Protokoly pracující s vektory vzdáleností jsou založeny na Bellman-Fordově algoritmu pro vyhledávání nejkratších cest [51], který hledá cestu s nejkratší vzdáleností. Směrovače neznají celou topologii sítě, znají pouze své sousedy, přes které se dostanou k dalším uzlům. Sousedé si mezi sebou navzájem vyměňují zprávy s potřebnými informacemi. Zpráva obsahuje tzv. distanční vektor, který se skládá z destinace a příslušné vzdáleností k ní od uzlu, který zprávu odeslal. Tato vzdálenost k destinaci je definovaná metrikou, kterou může být například počet přeskoků mezi směrovači, přenosová šířka pásma nebo třeba cena za použití linky. Na počátku obsahuje směrovací tabulka každého směrovače jen destinace, které jsou staticky nakonfigurovány administrátorem. Směrovací tabulka je periodicky

rozesílaná všem svým sousedům. Každý směrovač si pomocí z těchto přijatých dat doplňuje a upravuje svou směrovací tabulku. Pokud soused nabízí cestu, kterou směrovač ještě nemá, přidá si ji do své směrovací tabulky. Pokud soused nabízí cestu, kterou směrovač již má, ale má ji s horší metrikou, do směrovací tabulky se místo ní zaznamená lepší cesta od tohoto souseda. Ostatní nabízené cesty jsou ignorovány případně jsou uschovány v nějaké jiné datové struktuře jako alternativní cesty pro případ, že ve stávající cestě se objeví chyba.

Odstraňování již neaktuálních cest se děje tak, že informace o každé cestě musí být sousedem periodicky aktualizovaná, takže pokud cesta nebyla delší dobu sousedem inzerována, ze směrovací tabulky se odstraní.

U linkově stavových protokolů probíhá směrování na základě znalosti „stavu“ jednotlivých linek sítě a také díky tomu, že směrovače znají topologii celé sítě, kterou si udržují ve své topologické databázi [49]. Každý směrovač sleduje stav svých přilehlých linek. Pokud se stav a funkčnost přilehlých linek směrovače změní, rozešle informaci o aktuálním stavu do svého okolí ostatním směrovačům. Všechny směrovače tak mají v topologické databázi stejné záznamy. Každý směrovač z těchto dat počítá pomocí Dijkstrova algoritmu strom nejkratších cest [52] a pomocí něj si následně sestavuje a udržuje svou směrovací tabulku. Hlavní výhodou linkově stavových protokolů je to, že se v síti šíří pouze informace o změnách stavu místo periodického rozesílání směrovacích tabulek u předchozí typů směrovacích protokolů.

Mezi protokoly pracující s vektory vzdáleností patří směrovací protokoly RIP (Routing Information Protocol) [47] a BGP (Border Gateway Protocol)[1] [4] zatímco mezi linkově stavové protokoly patří směrovací protokol OSPF (Open Shortest Path First) [46]. Všechny zmíněné protokoly jsou z principu dynamickými směrovacími protokoly.

2.1.1. Vnitřní a vnější směrování

Celý Internet je tak rozsáhlý a proměnlivý, že je nemožné udržovat ve směrovačích úplnou informaci o celé síťové topologii. Proto byl Internet hierarchicky rozdělen na části na tzv. autonomní systémy. Autonomní systém (AS) představuje část sítě s vlastním nastavením směrování oddělenou vůči zbytku Internetu. Typickým příkladem je síť jednoho poskytovatele Internetu (operátora) a jeho připojených zákazníků.

Autonomní systémy se mohou vnořovat. Vnořené (downstream) AS může být síť velké korporace nebo síť regionálního poskytovatele Internetu, kteří využívají konektivitu hierarchicky o úroveň vyššího (upstream) národního nebo nadnárodního operátora. Jeden poskytovatel připojení nebo velká korporace může mít přípojení k Internetu tvořeno více linkami k jednomu upstream operátoru nebo linkami přes více upstream operátorů. Systém se pak nazývá multi-homed systém.

[1] BGP patří svými vlastnostmi zároveň do obou skupin protokolů, někdy je označována tato skupina jako protokoly pracující s vektorem cest (path-vector).

Správce AS rozhoduje jaká linka bude dle různých kriterií použita při připojení k Internetu. Těmito kriterií bývá často např. kvalita linky, přenosová rychlost, poplatek za přenášená data. Pokud máme jen jednu linku k upstream operátoru, máme single-homed systém.

Při směrování mezi AS jsou autonomní systémy chápány jako základní jednotky, jejichž struktura není mimo hranice autonomního systému známa. Z každého autonomního systému se do okolí sdělují adresy sítí, které autonomní systém obsahuje a také adresy sítí dalších AS, ke kterým má cestu. Autonomní systémy se donedávna číslovaly celosvětově jednoznačnými šestnáctibitovými čísly. V souvislosti s rostoucí velikostí Internetu a úbytkem volných šestnáctibitových čísel jsou novým autonomním systémům přidělována třicetidvoubitová čísla [44].

Uvnitř autonomního systému se používá vnitřní směrovací protokol (Interior Gateway Protocol, IGP). Typickými představiteli jsou protokoly RIP [47] a OSPF [46]. Předávání směrovacích informací mezi autonomními systémy probíhá pomocí vnějšího směrovacího protokolu (Exterior Gateway Protocol, EGP). Zde je hlavní prioritou stabilita směrovacího protokolu, proto je v porovnání s vnitřním směrovacím protokolem pomalejší na rychlost změn. Prakticky jediným představitelem této skupiny je zatím protokol BGP [4]. Tento protokol není jen výhradně vnější směrovací protokol, kdy může být použit i jako vnitřní směrovací protokol uvnitř AS (interní BGP) v případech, kdy se jeden AS skládá z více vnořených AS.

2.1.2. Směrování v BGP

Mezi autonomními systémy probíhá dnes směrování výhradně pomocí protokolu BGP. Směrovače si mezi sebou vyměňují zprávy se směrovacími informacemi, které obsahují cesty ke konkrétním destinacím (cílovým sítím, prefixům). Z těchto zpráv si směrovače vytvářejí své směrovací tabulky. Důležitým atributem každé inzerované cesty je atribut *ASpath*, který představuje posloupnost čísel AS, kterými vede cesta do destinace.

Narozdíl od jiných směrovacích protokolů nemá BGP jednoznačnou metriku podle které by se vždycky zvolila nejkratší cesta do jednotlivých destinací. Při směrování mezi autonomními systémy se směruje provoz přes cizí (tranzitní) autonomní systémy, jejichž provozovatelé (operátoři) mají nejrůznější provozní podmínky a obchodní zájmy. Pomocí těchto nejrůznějších faktorů určuje každý provozovatel autonomního systému svou tzv. směrovací politiku (routing policy) [49]. Směrovací politika zpravidla určuje

- do kterých AS necháme tranzitní provoz přes náš AS
- z kterých AS necháme tranzitní provoz přes náš AS
- kterým výstupním rozhraním (linkou) z našeho AS necháme odcházet provoz k daným destinacím

- kterým vstupním rozhraním (linkou) do našeho AS necháme přicházet provoz a ke kterým našim sítím

Do konfigurace BGP protokolu je nutné zahrnout všechny faktory směrovací politiky a tím je jeho konfigurace je více manuální než u ostatních směrovacích protokolů.

Jak už víme, směrovací protokoly se rozdělují na protokoly pracující s vektory vzdálenosti a na protokoly linkově stavové. Z pohledu úrovně znalosti topologie sítě, způsobu předávání i obsahu směrovací informace se protokol BGP řadí mezi ně, z každé třídy má něco. Někdy bývá označován jako protokol speciální třídy nazývané protokol pracující s vektorem cest (path-vector). Vektor cest je posloupnost autonomních systémů, přes které vede cesta do cílové destinace (sítě). Slouží také k výběru nejkratší cesty do jednotlivých destinací. Nejkratší bude považována ta cesta, která prochází nejmenším počtem autonomních systémů. Při výběru z více alternativních cest bude preferována kratší cesta. Nejvíce preferovaná cesta je v BGP protokolu určena nejprve podle směrovacích politik a následně podle délky cesty.

Směrovací informace se vyměňují vždy jen navzájem sousední směrovače. BGP směrovače se nacházejí vždy na hranici autonomních systému (kromě výjimky v případě použití interního BGP v tranzitních AS, kde se směrovače musí „dohodnout“, který z nich bude hraniční směrovač). Směrovač pošle zprávu svým sousedů, jen případě, že došlo ke změně ním preferované cesty k destinaci. Pracuje se zde se dvěma typy zpráv. První inzerují destinaci a cestu k ní a druhá oznamuje nedostupnost destinace.

2.2. CIDR a agregace IP adres

IP adresa je rozdělena v binárním zápise na dvě části. Levou část (prefix) IP adresy tvoří identifikátor sítě a zbylá pravá část IP adresy představuje identifikátor uzlu. Rozsahy pro adresování síti a uzlů byly nejprve rozděleny do několika tříd A, B, C.

Třída	**Počet bitů ID sítě**	**Počet bitů ID uzlu**	**Celkem uzlů v síti**
A	8	24	16777214
B	16	16	65534
C	24	8	254

Tabulka 1.: Rozdělení IP adres do tříd

Nejvíce se ujaly sítě třídy B, protože představovala určitý kompromis mezi počtem a velikostí sítě, síť třídy A byla příliš velká a síť tříd C zase příliš malá.

Internet se rozšiřoval a brzy počátkem 90. let se začaly objevovat problémy. Jak rostl počet připojovaných počítačů do nejrůznějších sítí, došlo k vyčerpání adresního prostoru pro třídu B a rychle přibývaly záznamy ve směrovacích tabulkách. Směrovací tabulky obsahovaly příliš mnoho záznamů, začaly být náročné na paměť routerů a také byla složitější jejich údržba a vyhledávání v nich. Proto vznikla nová pravidla pro tvoření podsítí, IP adres a manipulaci s nimi v rámci celého Internetu, které dostaly podobu konvence, pojmenované Classless Inter-Domain Routing (CIDR). Tato konvence nahrazuje původní „třídní“ charakter IP adres (jejich rozdělení na třídy A, B a C - proto také přívlastek „classless“).

Základní charakteristikou beztřídního rozdělení IP adres je existence síťové masky. Síťová maska označuje počet bitů zleva (prefix), které v IP adrese identifikují (maskují) síť. Zbylé bity představují identifikátor uzlu. Nyní se IP adresy přidělují po tzv. CIDR blocích, s velikostí danou příslušnou maskou, takže jemnost, s jakou jsou adresy čerpány z prostoru všech IP adres, může být velmi pružně přizpůsobována skutečným potřebám koncových zákazníků, což vedlo ke snížení rychlosti vyčerpávání celého adresového prostoru. Menší sítě se začaly hierarchicky spojovat (agregovat) na větší sítě a to umožnilo redukovat záznamy ve směrovacích tabulkách a zrychlit směrování v sítích. Ve směrovacích tabulkách routerů se pak uchovávají jen prefixy. Čím je velikost prefixu kratší (obecnější), tím více sítí nebo uzlů může zahrnovat.

Před vznikem CIDR nebyly IP adresy konkrétních sítí závislé na způsobu jejich připojení - pokud se provozovatel nějaké sítě rozhodl změnit svého poskytovatele připojení k Internetu, mohl si vybrat jiného a ponechat si své dosavadní IP adresy (neboť stačilo pouze změnit položky ve směrovacích tabulkách, odpovídající jeho síťovým adresám). V roce 1993 byl zaveden nový způsob přidělování IP adres [48], které závisí na poskytovateli sítě, který je správcem autonomního systému. Svým zákazníkům pak může přidělovat IP adresy jen ze svého adresního rozsahu, vytváří si tak vlastní další podsítě. Při změně poskytovatele musí zákazník překonfigurovat svoji síť na jiné IP adresy. Předchozí verze protokolu BGP-3 neumožňovala CIDR, proto byl nahrazen novějším BGP-4.

Od roku 1994 do roku 2001 rostla exponenciálně velikost globální směrovací tabulky BGP [45]. Díky vytváření nových podsítí v rámci již existujících sítí pomocí CIDR se na chvíli její růst zpomalil. Avšak od roku 2004 (díky zvýšené poptávce po více připojeních na 1 zákazníka) roste velikost směrovací tabulky znovu exponenciálně. V prvním čtvrtletí roku 2011 obsahuje globální BGP tabulka kolem 350 tisíc záznamů.

2.3. Popis BGP protokolu

Protokol BGP-4 je jeden de-facto jediný používaný externí směrovací protokol, který se používá pro směrování v Internetu. Byl definován nejprve v RFC 1654 [3], a dále v RFC 1771 [2] v březnu roku 1995. V dnešní době se čím dál častěji používá protokol BGP-4 dle RFC 4271 [4] z roku 2006 nebo Multiprotocol BGP-4

[5], který obsahuje podporu IPv6. Popisován bude protokol BGP-4 dle RFC 1771 [2], který je zatím nejvíce rozšířený (a také v důsledku užití simulátoru SSFNet, který obsahuje implementaci BGP protokolu podle tohoto RFC), některé důležité změny v protokolu BGP-4 popsané v RFC 4271 budou stručně shrnuty na konci kapitoly.

BGP-4 se řadí mezi protokoly pracující s vektory cest. Sousední směrovače si mezi sebou vyměňují směrovací informace o dostupných cestách k různým cílům. Destinace je vždy inzerována s cestou, jejímž důležitým atributem je ASpath, který představuje posloupnost čísel AS, kterými cesta prochází do destinace.

Každý směrovač před oznámením cesty dál, vloží číslo svého autonomního systému, ve kterém se nachází, před tuto ASpath. Díky tomuto mechanismu je zajištěna detekce jednoduchých smyček. Pokud ASpath v příchozí zprávě již obsahuje číslo autonomního systému příjemce, směrovač zprávu zahodí.

Výměna zpráv probíhá jen tehdy, jestliže došlo k nějaké změně v síti jako je nedostupnost některého uzlu nebo ztráta spojení na některé lince mezi dvěma uzly.

Pro výměnu těchto zpráv mezi dvěma autonomními systémy musí navázat dva jejich směrovače spojení, ustanovit tzv. BGP relaci. Dvěma sousedním routerům, kteří navzájem komunikují pomocí protokolu BGP, se někdy říká BGP sousedé (peery). Komunikace mezi BGP sousedy probíhá přes transportní vrstvu s použitím protokolu TCP na portu 179.

2.3.1. Úschova a propagace cest

Cesta je základní jednotkou informace v protokolu BGP. Je definovaná jako dvojice $(destinace, atributycesty)$. Cesta je mezi dvěma sousedy posílána v Update zprávě. Jedním z atributů cesty je ASpath, která představuje posloupnost čísel autonomních systémů, kterými cesta do destinace prochází.

Pro ukládání a zpracovávání přijatých cest používá BGP uzel tří datových struktur, které se celkově označují pojmem Routing Information Bases. Jsou to tyto:

- Adj-RIBs-In - uchovává cesty, které mu oznámili jeho sousedé. Pro každého souseda existuje jedna Adj-RIB-In tabulka. Jsou v nich obsažená nezpracovaná data, která jsou vstupem do algoritmu výpočtu nových cest a aktualizace stávajících cest ve směrovací tabulce.

- Loc-RIB - uchovává cesty pro každou dosažitelnou destinaci, které vybral BGP uzel z tabulek Adj-RIB-In jako nejlepší na základě metriky vzdáleností nebo vlastních směrovacích politik . Je to vlastní směrovací tabulka, pomocí níž následně BGP uzel odesílá příchozí pakety do správných cílových destinací.

- Adj-RIbs-Out - uchovává cesty, které se nově objevily nebo byly změněny v tabulce Loc-RIB a které BGP uzel oznámí v nejbližší době jeho sousedům. Pro každého souseda existuje jedna Adj-RIB-Out tabulka.

Do těchto 3 typů směrovacích tabulek se ukládají cesty z příchozích z Update zpráv tak, že každý řádek tabulky obsahuje destinaci a cestu k ní s příslušnýmu atributy, které jsou detailněji popsány v následující kapitole 2.3.2.. Atributy cesty slouží k výpočtu vhodné cesty pro směrování. Mezi základní atributy cesty patří Next-hop a ASpath.

2.3.2. Typy zpráv

Protokol BGP-4 používá při komunikaci 4 typy zpráv - *Open*, *Update*, *Notification* a *Keepalive*.

Zprávou **Open** se navazuje spojení mezi dvěma BGP uzly a otevírá se nová BGP relace. Je to první zpráva, kterou se zahajuje komunikace a obsahuje tyto informace:

- Version - používaná verze BGP protokolu
- My Autonomous System - číslo autonomního sytému odesílatele
- Hold Time - časový interval, který odesílatel navrhuje pro časovač Hold Timer, během kterého musí odesílatel poslat buď zprávu KeepAlive nebo zprávu Update, jinak se BGP relace uzavře.
- BGP Identifier - IP adresa jednoho z rozhraní odesílajícího (hraničního) routeru, které má spojení se sousedním BGP uzlem.
- Optional Parameters - dodatečné další parametry, které byly definovány později než vznikl původní návrh BGP protokolu. Dle nejnovějšího RFC 4271 [4] sem patří parametr Authentication Information, který obsahuje autentifikační informace pro šifrovanou komunikaci, v případě, že BGP uzly používají autentifikační mechanismus.
- Optional Parameters Length - celková délka Optional Parameters v oktetech

Update zprávy přenášejí informace o dostupnosti uzlů v síti a jejích vlastních destinací mezi BGP sousedy. Oznamují buď dostupnost nějaké destinace pomocí nové cesty nebo oznamují nedostupnost destinace (nebo celého uzlu v případě nedostupnosti všech jeho vlastních destinací).

Jedna zpráva obsahuje informace pouze o jedné cestě. Jsou to tyto informace:

- Withdrawn Routes - obsahuje IP prefixy destinací, jejichž cesty jsou neplatné a mají být odstraněny ze směrovacích tabulek

- Unfeasible Routes Length - délka pole Withdrawn Routes v oktetech
- Path Attributes - obsahuje informace o nové cestě, které jsou uloženy v následujících atributech:
  - Origin - původce této zprávy, vnitřní nebo vnější autonomní systém.
  - AS_path - kompletní cesta složená z čísel autonomních systémů, jimiž cesta prochází.
  - Next-Hop - obsahuje IP adresu interního nebo hraničního routeru (závisí na tom, zda další AS v cestě do destinace je vnitřní nebo vnější AS), který se musí nacházet ve stejné síti (podsíti), ve které se nachází i jedno ze síťových rozhraní odesílatele. Přes zvolený směrovač se dostaneme do dalšího AS v cestě do destinace.
  - Multi-Exit-Disc - číselné ohodnocení Next-Hopu. Pokud je možný výběr uzlu jako next-hopu ke stejnému sousednímu AS z více možností, při výběru vhodné cesty se bere v úvahu ten s nejnižším ohodnocením.
  - Local-Pref - další lokální metrika uvnitř AS pomocí nichž se probíhá výběr nejlepší cesty uvnitř AS.
  - Atomic_aggregate - informuje sousední AS o tom, že byla vybrána méně specifikovaná (obecnější, agregovaná) cesta, která zahrnuje původní více specifikovanou cestu (jde o agregaci cest)
  - Aggregator - obsahuje číslo posledního AS a IP adresu jeho routeru, ve kterém vznikla agregovaná cesta.
- Total Path Attribute Length - celková délka všech předchozích atributů v poli Path Atributes v octetech.
- Network Layer Reachability Information (NLRI) - obsahuje IP prefix cílové destinace.

Update zprávy se dělí na 2 typy podle toho, jakou informaci obsahují a jak se příjemce zprávy zachová.

- Announcement (oznámení) - zpráva inzerující cestu, která oznamuje novou cestu k nějaké destinaci, příjemce vloží NLRI a atributy cesty do odpovídající tabulky Adj-Rib-In, ve které se nacházejí všechny inzerované cesty odesílatele.
- Withdrawal (stažení) - zpráva, která oznamuje nedostupnost destinace. V Update zprávě se používá jen pole Withdrawn Routes a Unfeasible Routes Length. Tato zpráva informuje příjemce o tom, že neexistuje žádná alternativní cesta od odesílatele k destinaci. Dojde tak ke stažení cesty k destinaci, kterou před tím oznámil odesílatel zprávou inzerující cestu. Příjemce odstraní příslušné NLRI spolu s atributy cesty z odpovídající tabulky Adj-Rib-In vztahující se k odesílateli.

Jedna Update zpráva ale může kombinovat oba typy, protože může oznamovat nedostupnost nějakých destinací a zároveň informovat o nové cestě k nějaké jiné destinaci.

Zprávou **Notification** upozorňuje BGP uzel na chybu, která u něho nastala. BGP relace je okamžitě uzavřena po odeslání této zprávy.

- Error Code - obsahuje číslo chyby, která nastala
- Error Subcode - více upřesňuje typ chyby, která je uvedena v poli Error Code
- Data - obsahuje konkrétnější informace o chybě určené k diagnostice a vyřešení vzniklé chyby

Pomocí **Keepalive** se udržuje spojení mezi BGP uzly, které si mezi sebou periodicky posílají. Informují se tak navzájem, že je soused dostupný a nenastaly žádné potíže. Pokud jeden z nich přestane být dostupný, musí směrovač odstranit všechny cesty vedoucí přes něj a informovat o nedostupnosti cest ostatní sousedy, se kterými má otevřenou BGP relaci. Doporučený časový interval mezi dvěma KeepAlive zprávami je třetina Hold Time časového intervalu (Hold Time je jeden z atributů v Open zprávě).

2.3.3. Proces aktualizace směrovací tabulky

Po přijetí se Update zpráva zpracuje následujícím způsobem. Pokud jde o zprávy inzerující cestu, tak se nejdříve ověří, zda cesta neobsahuje smyčku, tj. zda v atributu cesty ASpath se již nenachází číslo autonomního systému příjemce, pokud ano, její zpracování končí. Jinak se přijatá cesta uloží do tabulky Adj-RIB-In pro daného souseda. Pokud už byla v tabulce Adj-RIB-In uložena z nějaké předchozí Update zprávy jiná cesta ke stejné destinaci, přepíše se na novou cestu.

Pokud dorazí zpráva oznamující nedostupnost destinace, která byla dříve oznámená, odstraní se z tabulky Adj-RIB-In (včetně její cesty), která obsahuje cesty od odesílatele.

Výpočetní proces pro výběr nejlepší cesty

Následně se provede výběr optimálních cest do směrovací tabulky Loc-RIB. Tento výpočetní proces se skládá z následujích po sobě jdoucích tří fází:

1. **fáze - Ohodnocení cest dle preferovanosti**
   Před zahájením této fáze se uzamknou všechny tabulky Adj-RIB-In (všechny další příchozí Update zprávy jsou ukládány mezitím do pomocného fronty pro pozdější zpracování). Pro každou novou nebo nahrazenou cestu spočte lokální BGP uzel pomocí ohodnocovací funkce ohodnocení

preferovanosti na základě metriky směrovacích politik a délky cesty (počtu AS, kterými cesta do destinace prochází). Nakonec se opět otevřou tabulky Adj-RIB-In pro zápis.

2. fáze - Výběr cest

V této fázi se pracuje výsledky ohodnocení preferovanosti cest, které byly získány v 1. fázi. Začne se upravovat směrovací tabulka Loc-RIB dle tabulek Adj-RIB-In.

Cesty, které byly staženy pomocí zprávy oznamující nedostupnost destinace a nacházejí se v tabulce Loc-RIB, z ní budou odstraněny.

Pro každou existující destinaci nalezne nejvíce preferovanou cestu podle následujících pravidel:

- pokud existuje více cest k jedné destinaci, vybere se ta s nejvyšším ohodnocením.
- pokud existuje více cest k jedné destinaci se stejným ohodnocením, zkoumají se u nich atributy cesty. Nejdříve se vybere cesta s menším Multi-Exit-Disc ohodnocením, pokud jich je stále více, tak dále s menším Local-Pref ohodnocením a nakonec, pokud jich je ještě stále více, vybere se cesta s nejmenším BGP Identifier.
- pokud existuje jen jediná cesta k destinaci, vybere se ta.

Nalezené cesty se vloží (v případě stejné destinace se nahradí) do směrovací tabulky Loc-RIB.

3. fáze - Šíření změněných cest

Pokud byl obsah tabulky Loc-RIB v 2. fázi změněn, mohou být inzerovány tyto změny okolním sousedům. Přidané destinace a jejich cesty nebo nové cesty ke stávajícím destinacím se uloží do tabulek Adj-RIBs-Out. Zde je možné použít agregaci cest. Nakonec se rozešle obsah tabulek Adj-RIBs-Out okolním sousedům v zprávách inzerující cestu a zároveň s nimi i zprávy oznamující nedostupnost destinace k již nedostupným destinacím, které byly předtím v Loc-RIB.

2.3.4. Další poznámky k protokolu BGP

Rychlost přenosu Update zpráv mezi sousedy bývá ovlivněno zatížením (latencí) jejich linky (delay link time) a časem vlastního výpočetního procesu pro výběr nejlepší cesty(tzv. process time), který zpracovává příchozí Update zprávy.

Mezi přijatými Update zprávami musí uplynout minimálně čas stanovený časovým intervalem Minimum Route Advertisement Interval (MRAI), který by měl být dle [2] nastaven na 30 sekund.

Časovým intervalem Minimum AS Origination Interval (MASOI) se omezuje, jak nejvíce často mohou být oznamovány změny, které proběhly uvnitř AS, okolním sousedům. Měl by být dle [2] nastaven na 15 sekund.

Časový interval KeepAlive stanovuje čas, během kterého musí být odeslána Keepalive zpráva, která udržuje otevřenou BGP relaci v případě, že neprobíhá výměna Update zpráv se sousedem. Dle [2] by měl být nastaven na 30 sekund.

Příjímání zpráv by mohlo způsobit náhlé přetížení BGP uzlu, pokud by je přijal od všech jeho sousedů v jeden časový okamžik. Proto se k výše uvedeným časovým intervalům MRAI, MSAOI a KeepAlive připočítává další čas pro pozdržení odeslání zprávy, tzv. Jitter. V BGP uzlu se před započtením odpočítávání času odpovídajícími časovači, upraví počáteční hodnota startu prvního odpočítávání daného časovače, která se vynásobí náhodným faktorem z intervalu 0,75 - 1 (všechny časovače v jednom uzlu stejnou hodnotou). Dojde tak ke zmírnění špiček při přenosu a zrovnoměrnění distribuce nových informací mezi jednotlivými uzly.

V roce 2006 se objevil nový RFC dokument popisující směrovací protokol BGP-4 - RFC 4271 [4]. Oproti původnímu RFC 1771 bylo tímto dokumentem upřesněn význam a použití některých atributů (NEXT-HOP, ATOMIC-AGGREGATE) ve zprávách používaných v protokolu BGP, práce protokolu při vnitřním směrování v AS a přidalo doporučení, že každá implementace BGP protokolu by měla obsahovat podporu šifrování s tajným klíčem k ochraně BGP relace pomocí TCP MD5 (RFC 2385 [6]).

3. Problém konvergence

BGP uzel si vyměňuje informace o dosažitelností destinací se svými sousedy, pokud nastala nějaká změna v síti, která má vliv na platnost cest ve směrovacích tabulkách.

Pokud nastane v síti jakákoliv změna, která ovlivňuje cesty ve směrovacích tabulkách, dojde tak dočasně k porušení **stabilního stavu** a k běhu procesu konvergence, který trvá do doby, než uzly všechny tyto změny zohlední ve svých směrovacích tabulkách a dostanou se tak znovu do stabilního stavu. Tento časový interval je někdy označován **dobou konvergence**. V tomto období mají některé uzly nesprávné údaje o cestách k ostatním uzlům a uzel směruje pakety podle nesprávných údajů a zároveň dochází k výpadkům připojení k těmto uzlům a jejich destinacím. Proto se mnoho studií zabývalo problémem, jak dobu konvergence snížit.

3.1. Měření a studie problému konvergence

Základní vlastností vektorových protokolů je dle [9] a [17] **pomalá konvergence** jejich směrovacích algoritmů, která závisí na rychlosti šíření změn v topologii sítě a na velikosti sítě. Čím více je síť rozsáhlejší, tím pomaleji se informace o nějaké změně v síti rozšíří mezi všechny uzly.

V protokolu BGP může dojít k porušení stabilního stavu ze dvou důvodů. Může to být z důvodu jakékoliv změny topologie sítě nebo některé z její vlastnosti (např. zpoždění na lince) nebo změny směrovacích politik v některém autonomním systému. Změnu topologie sítě zjistí BGP uzel tak, že některý jeho soused mu poslal Update zprávu, ve které oznamuje novou cestu k nové nebo stávající destinaci nebo naopak se jedná o zprávu, informující o nedostupnosti destinace.

Po ztrátě cesty k nějaké destinaci následně začne BGP uzel vyhledávat novou cestu přes jiného souseda ve svých Adj-RIBs-In tabulkách [4]. Po vybrání nové cesty výpočetním procesem o ni informuje pomocí Update zprávy sousední BGP uzly. Pokud žádná cesta nebyla nalezena, pošle jim zprávu o nedostupnosti destinace.

BGP uzel nemusí nalézt novou stabilní cestu k destinaci hned napoprvé. V případě, že od některého z jeho sousedů, kterým tuto cestu oznámil, přijde zpráva o nedostupnosti destinace, pokusí se najít novou cestu přes zbylé sousedy a informuje o tom všechny své sousedy. K nalezení konečné stabilní cesty tak může BGP uzel prozkoumat všechny možné kombinace cest. Tento problém přeskakování způsobuje prodlužování doby konvergence. Podle měření provedených Labovitzem at al. v [9] trvá dosažení konvergence v internetové topologii v průměru 3 minut, ale v některých případech však trvá i 15 minut.

Jedním z dalších problémů, který prodlužuje dobu konvergence, je neustálé střídavé přepínání dvou cest k jedné destinaci (route flapping). Při selhání rou-

teru ať už z důvodu poruchy hardwaru nebo softwaru nebo kvůli fyzické závadě na lince, nejbližší soused najde a oznámí svému okolí jinou cestu k „postižené“ destinaci. Někdy však může nastat jen velmi malé zdržení výpočetního procesu nebo malé zpoždění na lince při zvětšené zátěži na výpočetní prostředky routeru (jako je procesor a paměť), které může trvat zlomky sekundy a dojde k obnovení předchozí cesty. Po obnovení původní cesty tak inzeruje uzel znovu cestu, kterou inzeroval v nedávné krátké době.

V letech 1996 - 1998 byl prováděn výzkum a měření nestability při směrování mezi autonomními systémy amerických poskytovatelů, který prováděli Labovitz et al. a popsali v článcích [7] a [8]. Zjišťovali, jaký vliv mají příchozí Update zprávy na směrovací tabulky, jejich celkový počet při komunikaci mezi páteřními směrovači a vliv na výkon koncových sítí zákazníků. Zjistili, že s velikostí směrovací tabulky lineárně roste množství střídavě přepínaných cest k destinacím. K velkému problému při směrování může dojít, když je mnoho takto střídavě přepínaných cest. Směrovače rozesílají znovu zprávy, které rozesílaly nedávno. To může způsobit zahlcování sítě (a k prodlužování doby doručení zpráv nebo jejich zahození) a zvýšený nápor na výpočetní prostředky. V krajním případě to může vést k výpadku rozsáhlého množství routerů a nedostupnosti destinací a například i e-mailových a webových služeb. V roce 1997 nastal velký výpadek konektivity, který postihl několik miliónů amerických uživatelů.

Největší slabina se ukázala v implementacích BGP protokolu, protože tento problém nestability neobjevil jen na malém množství routerů (byla vyloučena hardwarová závada), ale vznikl ve velkém množství autonomních systémů. Proto hlavní výrobci routerů implementovali do svých zařízení lepší mechanismy k řízení přenosu při komunikaci protokolem BGP, ve kterém jsou například při zvýšené zátěži prioritně odesílány Keepalive zprávy, které udržují spojení.

Z předešlých měření Labovitz et al. zjistili nejmenší a největší časovou složitost konvergence po selhání nějaké cesty: doba konvergence roste lineárně s délkou alternativní cesty *ASpath* [9],[10]. Dále navrhli vylepšení, která lze použít pro snížení doby konvergence jako je detekce smyček v cestě *ASpath* na straně odesílatele (Sender-Side Loop Detection).

Griffin et al. se zabývali konvergencí protokolu BGP a zkoumali vliv MRAI, vlastností topologie a směrovacích politik [11]. Zjistili, že doba konvergence protokolu BGP s implicitním MRAI časem 30 sekund závisí na délce nejdelší *ASpath* v síti, kterou její některá cesta k destinaci obsahuje. Zároveň provedli srovnání s proměnlivými časy MRAI a zjistili, že rychlost výběru alternativní cesty ke stejné destinaci závisí nejen na vlastních časovačích jednoho uzlu, ale i na hodnotách časovačů ostatních uzlů v jednom časovém okamžiku. Jako vylepšení snížení doby konvergence protokolu BGP navrhli adaptivní MRAI časovač a přidání dalších informací do zpráv, které oznamují nedostupnost destinace.

Griffin et al. vydali následně několik článků [12] a [16], ve kterém provádějí analýzu protokolu BGP jako distribuovaného grafového algoritmu. Protokol BGP se liší od jiných směrovacích protokolů pracujících s vektory cest možností expli-

citně určit preferovanou cestu k destinaci nezávisle na metrice nejkratších cest. Tuto metriku, zvanou směrovací politika, ručně nastavuje správce AS. Vlivem nastavených směrovacích politik může dojít k tzv. trvalé oscilaci, kdy dochází v pravidelných časových intervalech ke cyklickému přepínání z jedné cesty na druhou.

Varadhan at al. provedli důkaz divergence protokolu BGP [13], ve kterém ukázali, že u směrovacích protokolů pracujících s vektory cest jako je BGP mohou vlivem směrovacích politik nastat trvalé oscilace a protokoly nemusejí vždy konvergovat.

Protokol BGP tedy může i divergovat (nikdy nedojde ke konvergenci), protože administrátoři v jednotlivých AS mohou nastavit takové vzájemně vylučující politiky, kvůli kterým může dojít k trvalé oscilaci cest. Metrika nejkratších cest splňuje podmínku monotónnosti a izotónnosti [14]. U metriky směrovacích politik nelze zajistit, aby tato metrika byla monotonní a izotonní, protože směrovací politiky obsahují napevno zvolené preference cest. V článku [15] byla popsána specifikace politik, které garantují, že protokol BGP nebude nikdy konvergovat. Jiné protokoly pracující s vektory cest jako je RIP [47] garantují konvergenci, protože nejlepší cestu hledají jen podle metriky nejkratších cest.

Autoři článků [12] a [16] popisují, že problém konvergence lze řešit dynamicky nebo staticky. Dynamické řešení je založeno na tom, že při běhu protokolu BGP jsou eliminovány nebo potlačovány ty oscilace, které vznikly z důvodu konfliktu politik v několika AS. Statické řešení problému konvergence je založeno na analyzování směrovacích politik, při kterém dochází k ověřování, zda některá ze směrovacích politik nemůže způsobit divergenci.

V článku [12] je teoreticky popsáno hledání statického řešení v protokolu SPVP (Simple Path Vector Protocol), který je zjednodušeným modelem protokolu BGP. K tomu využívá tzv. evaluační graf. Je to stavový diagram, ve kterém stavy obsahují cesty, které jsou ve směrovacích tabulkách všech uzlů sítě v nějakém časovém okamžiku. Přechod z jednoho stavu do druhého probíhá na základě přijatých zpráv s inzerovanými cestami.

BGP systém může postupně dosáhnout stabilního stavu. Ve stabilním stavu mají všechny uzly sítě nastaveny svoji nejvíce preferovanou cestu a uzly nemají důvod ji vlivem příchozích Update zpráv měnit. Protokol BGP v tomto případě konverguje. Pokud z některého stavu dojdeme několika přechody do stejného stavu, znamená to, že při dané konfiguraci (konkrétních vybraných cest a konkrétních příchozích Update zpráv) BGP systému dochází k oscilaci cest, při které protokol BGP nebude nikdy konvergovat. Je ale možné, že existuje nějaká další konfigurace, která z tohoto výchozího stavu umožní přejít v dalších přechodech do stabilního stavu. Griffin et al. rozdělili BGP systémy do několika skupin podle řešitelnosti konfliktu směrovacích politik.

- systém je vždy řešitelný a protokol BGP konverguje
- systém je neřešitelný a protokol BGP diverguje

- systém je řešitelný jen v některých konfiguracích systému

Způsob zjištění toho, v jakých případech je BGP systém se směrovacími politikami řešitelný, se hodí se jen pro malé sítě do pěti uzlů, protože při větším počtu uzlů je to časově i paměťově náročné. Možnost použití statické analýzy pro řešení problému konvergence BGP v Internetu je silně omezeno, protože autonomní systémy nesdílejí mezi sebou směrovací politiky. Pokud by byly tyto směrovací politiky přístupné, je i tak nemožné nalézt v přijatelném čase uspokojivý výsledek, který by pokrýval všechny dostupné autonomní systémy. Navíc samotné ověřovací algoritmy, ze kterých se tato statická analýza konfliktnosti směrovacích politik skládá, jsou převážně NP těžké úlohy.

Z těchto důvodů musí být v praxi použito dynamické řešení problému konvergence. Sem patří například Route Flap Damping (RFD) [25], o kterém bude pojednáno v kapitole 3.5., který po určitou chvíli potlačuje cestu v inzerování dál, kvůli které dochází k oscilaci.

Později bylo zjištěno a popsáno v [12] a [26], že dochází jen ke zpomalení oscilací a ne jejich úplné eliminaci. Metoda ani nezjistí, že se tak děje z důvodu konfliktu politik, protože není možnost, jak to zjistit. Jedním z navrhovaných řešení by bylo vkládání informací souvisejícími s politikami do Update zpráv [12]. Pokud by byl v BGP uzlech implementován algoritmus, který by pracoval s těmito informacemi a řešil by oscilace způsobené politikami, dalo by se garantovat, že by oscilace nebyly trvalé. Dle nejnovějších doporučení [28] by Route Flap Damping neměl být poskytovateli připojení k Internetu vůbec používán. Zvýšení stability cest bez použití RFD je předmětem nynějšího výzkumu [35].

Výzkum protokolu BGP se rovněž zabývá zkoumánín, jak výrazněji zkrátit dobu konvergence. Pro snížení doby konvergence byly postupně navrženy metody Ghost-Flushing a Ghost-Buster [19], metoda konzistentních pravidel (Consistency Assertions) [20] a pak metoda určující původ změny (Root Cause Notification) [21], které jsou popsány v následujících kapitolách. Při výpadku nějaké cesty nebo destinace se v síti rychleji eliminují ty alternativní cesty, které jsou po změně topologie sítě rovněž neplatné a tím se urychluje stabilizace směrovacích tabulek.

Při uzavření BGP relace se všechny cesty k destinacím, které byly oznámeny přes tuto BGP relace stáhnou a nahradí jinou alternativní cestou, a pokud k tomu dochází často, narušuje to stabilní stav směrovacích tabulek. Nedávno byl popsán návrh BGP graceful shutdown [36], který umožňuje najít stabilní alternativní cesty zprávou Notification ještě dřív než se uzavře spojení a to umožňuje rovněž snížit dobu konvergence.

3.2. Příklady problémového chování protokolu BGP

Poprvé v roce 1997 došlo k velkému výpadku cest a uzlů, který měl velký dopad na stabilitu cest a připojení k Internetu, z důvodu špatných a nevhodných implementací BGP protokolu na tehdejších routerech.

Dalším nečekaným problémem, který může ovlivnit stabilitu cest a narušit stabilní stavy uzlů je maximální délka atributu *ASpath*, která bývá ve většině implementacích BGP protokolu omezena maximálně na 255 AS v jedné cestě. Pokud délka *ASpath* překročí tuto hodnotu, je tato cesta brána za špatnou, ačkoliv podle RFC 4271 je cesta v pořádku. Pokud chce administrátor AS znevýhodnit jednu cestu před druhou k vlastnímu prefixu (destinaci) bez použití směrovacích politik, vloží několikrát své číslo AS před stávající *ASpath* a tak ji prodlouží. Tuto vlastnost implementovali do svých novějších zařízení někteří výrobci routerů.

Podle [37] se to stalo osudným českému poskytovateli Internetu SUPRONET (z Uherského Brodu) v únoru 2009. Jeho administrátoři konfigurovali záložní cestu k nadřazenému poskytovateli připojení. Při zadávání parametru počtu opakování čísla AS zadali místo počtu opakování číslo svého AS (47868). Navíc software v routeru, který byl od MikroTik (výrobce routerů z Lotyšska), nehlídal zda hodnota padne do nějakého rozsahu (např. 0 - 255).

Routery se starší verzi softwaru Cisco IOS na přijetí takové dlouhé cesty reagují tak, že zruší BGP relaci a uzel je nedostupný. Následně se zase BGP relace obnoví, jenže po přijetí téže cesty ji zase zruší a tak dokola. U sousedních uzlů dochází k neustálému střídavému přepínání cest a to má za následek, že vlastník takového uzlu je bez konektivity. Cesta k této destinaci se postupně rozšířila postupně mezi nadnárodní operátory a ti, kteří měli směrovače Cisco se rázem ocitli v potížích s připojením se k Internetu.

Nebezpečným problémem, který má vliv na stabilitu cest a uzlů, je možnost inzerovat destinace se záměrně podvrženou *ASpath* nebo cesty rušit. To může narušit internetovou komunikaci napříč světadíly, omezit některé služby ke kterým přistupují stovky miliónů lidí. Poslední dobou se to děje v zemích Blízkého Východu až po jihovýchodní Asii.

V únoru 2008 pákistánský operátor Pakistan Telecom na několik hodin znemožnil přístup téměř celému světu službu Youtube tak, že prefix, na kterém běží tato služba, nasměroval na svůj AS. Routery nadřazeného poskytovatele začaly tuto cestu inzerovat dál, protože byla kratší než cesta, která pocházela od AS, v jehož síti běžela služba Youtube [38].

V dubnu 2010 směrovače největšího čínského operátora China Telecom inzerovaly během 15ti minut kolem 37000 destinací, které mu nepatřily (normálně vlastní kolem 40ti destinací) [39]. Jeho nadřazeni operátoři rozšířili asi 10 procent těchto prefixů buď z důvodu kratší délky *ASpath* nebo nastavených politik. Jednalo se o BGP hijacking velkého rozsahu, který velmi krátce postihl některé operátory v západní Evropě, Rusku, USA, Japonsku a Brazílii. Ze známých služeb byly omezeny např. CNN, Amazon, Rapidshare.

27.ledna 2011 se objevilo velké množství zpráv o nedostupnosti destinace, které měly svůj původ v Egyptě[40]. Tou dobou tam probíhaly nepokoje a tamní vláda nařídila místním operátorům, aby zablokovali připojení do Internetu a směrovače nadřazených operátorů to vyhodnotily jako nedostupnost destinace. Téměř všech 3000 egyptských prefixů bylo odstraněno z BGP tabulek světových operá-

torů. Až 2. února došlo k novému inzerování těchto prefixů [41]. Tato situace vedla v tyto dny k celosvětovému nárůstu zátěže na BGP směrovače a také poprvé v historii existence Internetu byl nějaký stát na několik dní ostřihnout od Internetu.

Z těchto případů je vidět, že nevhodným nebo záměrným zásahem v nastaveních BGP routerů, lze výrazným způsobem omezit funkci Internetu jako celosvětového informačního a komunikačního média.

Jedním z řešení je dle [42] omezit podřízeného operátora tak, že může inzerovat jen destinace, které sám vlastní, ale v praxi je to nepoužitelné, protože neexistuje žádná vyšší ověřovací autorita, která by to ověřovala. Řešení tohoto problému hledá několik organizací. Nejprve výrobci routerů implementovali vlastní řešení jako je nový typ zprávy pro digitální podpis pro Update zprávy, pomocí níž by se hlídala platnost prefixů, ale tato řešení se zatím neujala. Skupina SIDR (Secure Inter-Domain Routing) [43] při IETF navrhla do BGP protokolu funkcionalitu, která by umožnila ověření zdroje přijímaných dat a přitom nezatížila páteřní směrovače. O něco podobného se pokouší i jedna ze skupin v organizaci RIPE, která sdružuje evropské poskytovatele připojení.

3.3. Přehled stavů BGP uzlu při změně vlastností sítě

V průběhu komunikace BGP protokolu vlivem topologických změn nebo směrovacích politik se může směrovač (BGP uzel) dostat do některého z následujících stavů (někdy i do více zároveň), od kterých měříme čas (dobu konvergence), který uplyne do stabilizace směrovacích tabulek. Byly definovány v článcích [9] a [19]. Jsou to stavy:

- E_{up} - nová destinace je dostupná skrze nějakou cestu.
- E_{down} (fail-down) - destinace přestala existovat a neexistuje k ní nyní žádná cesta (výpadek BGP uzlu).
- $E_{shorter}$ - preferovaná cesta k destinaci je nahrazena kratší cestou než jaká byla dosud.
- E_{longer} (fail-over) - preferovaná cesta k destinaci je nahrazena delší cestou než jaká byla doposud (z důvodu směrovacích politik nebo výpadku tranzitního uzlu).

3.4. Vliv MRAI na rychlost konvergence

Při hledání nové cesty trvá určitý čas než BGP uzel nalezne stabilní cestu, která se dále již měnit nebude, pokud se stav topologie nezmění. V úplném grafu sítě s k uzly může každý BGP uzel během stabilizace v nehorším případě prozkoumat $k!$ všech možných cest pro jednu destinaci. Po nalezení nové cesty ji BGP

uzel ji oznámí svým sousedům pomocí Update zpráv. Může se stát, že než se cesta k destinaci na delší čas ustálí, BGP uzel ji bude vždy po každé změně cesty oznamovat sousedům. K tomu, aby nedošlo k šíření cest, které ve skutečnosti nemusejí být správné a zabránilo se zahlcení sítě Update zprávami, je omezeno jejich rozesílání po určených časových intervalech.

V BGP protokolu je tento časový interval uváděn jako Minimum Route Advertisement Interval (MRAI), který je standardně nastavený podle [2] na 30 sekund. Až do uplynutí tohoto časového intervalu nesmí být mezi dvěma sousedními uzly odeslána další Update zpráva s cestou ke stejné destinaci nebo nebo jakákoliv další Update zpráva v závislosti na tom, zda se MRAI časovač v konkrétní implementaci protokolu BGP vztahuje na destinaci nebo souseda.

Tento časový interval se vztahuje jen na Update zprávy, které inzerují cestu a nevztahuje se na zprávy o nedostupnosti destinace. Pokud je však v implementaci BGP protokolu povolen parametr WRATE (viz kapitola 3.7.), pak se rovněž tento interval vztahuje i na tyto zprávy.

MRAI nemá velký vliv na prodloužení časové konvergence v případě, že BGP uzel vybere po změně v topologii sítě napoprvé novou stabilní cestu. Má ale vliv na prodloužení doby konvergence v případě, že se tomu nestane napoprvé. Čím řidší je topologie sítě, tím rychleji dojde k nalezení správné cesty, protože se prohledává menší počet alternativních cest a zároveň je větší pravděpodobnost, že se strefí hned v několika prvních pokusech.

Z předchozího odstavce se dá odvodit funkce největší časová složitost konvergence BGP protokolu, která je

$$O(n.minRouteAdver) \tag{1}$$

kde n je délka nejdelší jednoduché cesty k destinaci a $minRouterAdver$ je časový interval MRAI.

Funkce největší komunikační složitosti se odvodí z toho, že každý uzel pošle svým sousedům cesty ke všem destinace, ke kterým má cestu.

$$O(nE) \tag{2}$$

kde n je délka nejdelší jednoduché cesty k destinaci a E je počet hran grafu sítě.

Jednoduchou metodou řešení problému s opakovanými pokusy hledání nové stabilní cesty by mohlo být zmenšit časový interval MRAI mezi vysíláním Update zpráv, kterou popisuje článek [19]. Pokud tuto hodnotu snížíme až na 0 sekund, časová složitost z $O(n.minRouteAdver)$ se sníží na $O(nh)$, ale zato velmi prudce stoupne komunikační složitost stabilizace z $O(nE)$ na $O(n!E)$. Došlo by k většímu zatížení sítě zvláště v topologiích, které se blíží úplnému grafu.

3.5. Route Flap Damping

MRAI časovač byl navržen k tomu aby na čas omezil změny cest během stabilizace. Pro rychlejší ustálení cest, kdy dochází ke střídavému přepínání cest na novou cestu a zpět na původní cestu, a tím i ke zkrácení doby konvergence, byla navržena metoda Route Flap Damping [25].

Pro každou destinaci d a pro každého souseda n si BGP směrovač udržuje trestné body (penalty) $p[d, n]$. Hodnota $p[d, n]$ se může změnit pomocí dvou pravidel:

1. Pokud soused n změní cestu k destinaci d, inkrementuje se penalta $p[d, n]$. Cesta k destinaci se může změnit z dostupné cesty na nedostupnou cestu a naopak, nebo se může přepnout na lepší cestu než byla doposud, nebo naopak na horší. V závislosti na typu změny je určeno, o kolik se zvětší stávající penalta $p[n, d]$.

2. Hodnota penalty $p[n, d]$ se roste exponenciálně v závislosti na čase podle rovnice:
$$p[d, n](t') = p[d, n](t)e^{\lambda(t'-1)} \tag{3}$$

Parametr λ je nejčastěji vyjádřen z half-life parametru H, který udává čas, kdy je možné snížit hodnotu penalty p o polovinu dle vztahu $e^{\lambda H} = 0, 5$

Na počátku jsou zvoleny 2 parametry. Mez potlačení - Suppression Treshold, která představuje limit, jejíž hodnotu menší nebo rovno je možné považovat za stabilní cestu. Dále je zvolen časový interval mez znovupoužitelnosti - Reuse Treshold, který představuje čas, po jejímž uplynutí lze s potlačenou cestu zase pracovat.

Jakmile BGP uzel přijme cestu k destinaci d od souseda n, spočítá novou hodnotu penalty $p[d, n]$ dle výše uvedených pravidel. Srovná tuto hodnotu $p[d, n]$ s mezí potlačení, a pokud je $p[d, n]$ větší nebo rovno než mez potlačení, pak tuto cestu v příslušné tabulce Adj-Rib-In označí jako potlačenou cestu. Při výpočtu cest do směrovací tabulky Loc-RIB se s takto označeními cestami nepracuje.

Když je cesta potlačena, je spuštěn zároveň k ní příslušný časovač Reuse Timer, jehož časový interval je nastaven na čas použitelnosti. Po vypršení času v tomto časovači, je možné tuto cestu použít při výpočtu cest do Loc-RIB. Pokud soused n změní cestu k destinaci d na jinou během spuštěného časovače Reuse Timer, je Reuse Timer zastaven a znovu spuštěn.

Základními parametry pro Route Flap Damping jsou tedy Suppression Treshold, Reuse Treshold a parametr λ. Některé implementace této metody v routerech mohou podporovat další speciální parametry jako rozdílné trestné body pro důvod změny. Například jiné hodnocení může být použito při přepnutí z horší cesty na lepší cestu a jiné ohodnocení, když k tomu dojde naopak.

Při nesprávném nastavení těchto parametrů může dojít k nechtěným efektům například, když poskytovatel pro své záložní cesty použije méně agresivní pravidla pro Route Flap Damping. Přepne-li se datový tok k (downstream) klientovi na

záložní cestu, může se stát, že nedojde ke zpětnému přepnutí na primární cestu, i když už bude dostupná.

Z těchto a podobných důvodů sdružení evropských poskytovatelů RIPE v [27] v roce 2001 vydalo nejprve doporučení, na jaké hodnoty tyto parametry nastavit. Z nejrůznějších pozorování bylo následně zjištěno a v [26] prokázáno, že mezi procesem rušení cest v BGP výpočetním procesu a mechanismem přepínání Route Flap Damping dochází k nežádoucí interakci v některých typech topologií, kdy nové oznámení jednou již stažené cesty, se může pozdržet i o hodinu. Dle článku [16] Route Flap Damping navíc neeliminuje trvalé oscilace v BGP protokolu z důvodu směrovacích politik, ale pouze jejich průběh zpomalí. Nakonec pak v roce 2007 sdružení RIPE vydalo doporučení [28], podle kterého by Route Flap Damping neměl být poskytovateli připojení k Internetu vůbec používán. A tak v nynější době probíhá výzkum problému stability cest bez použití Route Flap Damping [35].

3.6. Sender-side Loop Detection (SSLD)

Metoda Sender-side Loop Detection (SSLD) uvedená v článcích [11] a [24], umožňuje detekovat možný cyklus cesty v cestě v Update zprávě připravené k odeslání a znemožnit tak její odeslání. Pokud se v atributu cesty $ASPath$ objevuje číslo AS adresáta, tak odesílatel tuto zprávu neodešle a tím zmenšuje počet odeslaných/přijatých zpráv při konvergenci a velmi nepatrně zmenšuje dobu konvergence.

3.7. Withdrawal Rate Limiting (WRATE)

Při aktivaci parametru Withdrawal rate limiting (WRATE) v implementaci protokolu BGP, budou BGP uzly mezi sebou odesílat zprávy o nedostupnosti destinace stejným způsobem jako zprávy, které inzerují cestu, po MRAI časových intervalech.

Zpráva o nedostupnosti destinace může někdy narušit stabilní stav uzlu, když uzel ztratí současnou cestu k destinaci a nahradí ji jinou už rovněž neplatnou alternativní cestou. Avšak na druhou stranu WRATE může způsobit, že místo toho aby zpráva o nedostupnosti destinace odstranila už neaktuální cestu, tak dojde k tomu, že se prodlouží doba její existence.

V RFC 1771 [2] bylo doporučeno WRATE nepoužívat. Avšak někteří výrobci routerů i přesto WRATE standardně na svá zařízení implementovali a prosadili ji v připomínkách k dalšímu vývoji BGP protokolu [34], a podle nejnovějšího RFC 4271 [4] je možné WRATE používat.

Dle [11] a [24] je jisté, že WRATE vylepšuje dobu konvergence při stabilizaci z E_{down} v úplném grafu sítě, v ostatních případech je doba konvergence delší než bez WRATE. To potvrzuje i nejnovější studie [18], která kromě toho ukazuje

příznivý vliv WRATE na potlačení některých nežádoucích zpráv při optimálním nastavení MRAI časovače.

3.8. Metoda Ghost-Flushing

Jedna ze známých metod pro snížení doby konvergence je metoda Ghost-Flushing, která byla představena v článku [19].

BGP uzel rozesílá nové Update zprávy sousedním uzlům po uplynutí časového intervalu MRAI. Pokud dojde v průběhu plynutí MRAI časového intervalu k selhání nějaké cesty, sousedící BGP uzel se pokusí najít novou alternativní cestu, která je připravena k odeslání v nové Update zprávě až po uplynutí MRAI časového intervalu.

Pokud dojde v průběhu plynutí MRAI časového intervalu k selhání destinace, která není dostupná žádnou cestou, jeho bývalí přímí sousedé o tom informují své okolní sousedy zprávou oznamující nedostupnost destinace, v jejichž odeslání nic nebrání. Po přijetí těchto zpráv příjemci hledají alternativní cestu přes jiný uzel, než od kterého mu přišla zpráva o nedostupnosti destinace (neví totiž, že se k cíli nelze dostat žádnou cestou), kterou odešlou po uplynutí MRAI časovém intervalu. Avšak v tomto případě je každá taková alternativní cesta k již nedostupné destinaci neplatná. Tím se v síti šíří neplatné cesty, které se označují jako slepé cesty.

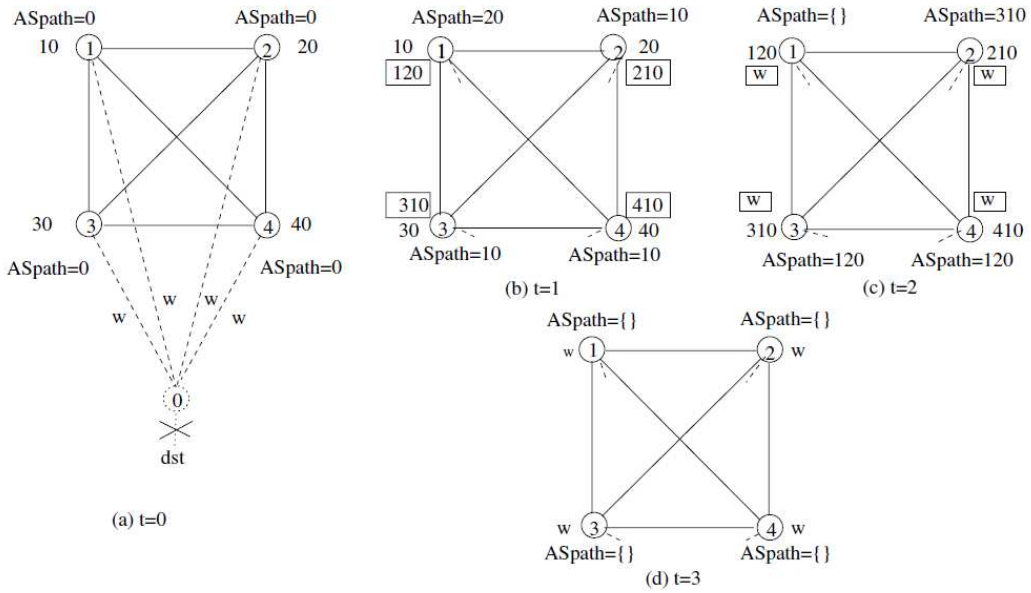
Z každým novým hledáním alternativní cesty k neexistující destinaci, buď cesta zůstane stejně dlouhá, nebo se prodlouží. Prodloužení cesty pokračuje až do té doby než vznikne v cestě smyčka (pozná se to podle atributu *ASpath*). Každá příchozí Update zpráva s touto cestou je dle [2] zahozena. Pokud příjemce zprávy v jakémkoliv okamžiku nenajde žádnou alternativní cestu přes žádného z jeho sousedů, odešle jim zprávu o nedostupnosti destinace.

Cílem metody Ghost-Flushing je co nejrychlejší odstranění těchto slepých cest a tím urychluje konvergence BGP protokolu. Děje se to pomocí dodatečných „čistících“ zpráv o nedostupnosti destinace. BGP uzel okamžitě odešle zprávu o nedostupnosti destinace, když si původní cestu ve své směrovací tabulce Loc-RIB nahradil tuto cestu novou cestou, která je ovšem delší, než byla původní cesta k dané destinaci. Rovněž se odesílají zprávy o nedostupnosti destinace, jestliže nebyla nalezena žádná alternativní cesta. Na obrázku 1. vidíme odpovídající scénář komunikace BGP protokolu s metodou Ghost-Flushing v síťové topologii úplného grafu se čtyřmi uzly.

Algoritmus metody Ghost-Flushing pracuje takto:

```
Když cesta k destinaci
  neexistuje nebo nahrazena delší cestou
  a minRouterAdver interval od poslední
  přijaté zprávy inzerující cestu dosud nevypršel
pak pošli zprávu o nedostupnosti destinace
```

obsahující tuto destinaci všem svým sousedům.



Obrázek 1.: Scénář komunikace BGP protokolu s metodou Ghost-Flushing v úplném grafu se čtyřmi uzly [19]

3.8.1. Výpočet časové a komunikační složitosti

Předpokládejme destinaci, která není nadále dostupná. Přímí sousedé uzlu, kterému destinace patřila, byli o tom informováni zprávou o nedostupnosti destinace. Délka alternativních cest, které mohou uzly „zdánlivě“ využít musí mít délku větší než 1 [19], protože byli informováni o nedostupnosti destinace uzlem s kterým mají otevřenou BGP relaci. Dalšími induktivními kroky se můžeme dostat až na nejdelší cestu v síti, která je možná pro danou destinaci, aniž by nevznikla smyčka, kterou označíme n. Zároveň v každé časové jednotce h, která představuje čas doručení zprávy mezi dvěma uzly, zmizí ze sítě slepé cesty s délkami menšími než jaká délka nejdelší cesty, která se zrovna v síti vyskytuje. Po nejvýše n krocích mají všechny uzly potlačenou cestu k nedostupnému uzlu. V [19] z této úvahy odvodili funkci největší časové složitosti konvergence, která pro metodu Ghost-Flushing vychází

$$O(nh) \tag{4}$$

kde n je délka nejdelší možné (jednoduché) cesty k destinaci a h je průměrný čas průchodu zprávy mezi dvěma sousedními BGP uzly.

V každém časovém intervalu MRAI může odeslat BGP uzel svému sousedovi nejvýše 2 zprávy - 1 inzerující Update zprávu a 1 zprávu o nedostupnosti

destinace. Navíc jsou zprávy posílány nejvýše do vzdálenosti n. V [19] odvodili funkce největší komunikační složitost (počtu odeslaných/přijatých Update zpráv) od stavu E_{down} do stabilizace směrovacích tabulek všech zbývajících uzlů sítě

$$O(\frac{2hnE}{minRouteAdver}) \tag{5}$$

kde n je délka nejdelší jednoduché cesty k destinaci, h je průměrný čas průchodu zprávy mezi dvěma sousedními BGP uzly a E je počet hran grafu sítě.

Metoda Ghost-Flushing se nejčastěji používá pro urychlení stabilizace sítě ze stavu E_{down}. Pro dobu konvergence po stavu E_{longer} nelze analyticky podle [19] ověřit, zda je vždy menší než v původním BGP protokolu, protože při výpadku jedné cesty k destinaci, je destinace stále dostupná jinou delší cestou, jejíž $ASpath$ zpravidla obsahuje část původní cesty.

Uzel, který používá metodu Ghost-Flushing, může podle výše uvedeného algoritmu odeslat svým sousedům čistící zprávu o nedostupnosti destinace, když to stihne před uplynutím času MRAI, kdy nelze odeslat novou cestu. V ostatních případech se protokol BGP s metodou Ghost-Flushing chová klasickým způsobem jako protokol BGP bez této metody, protože další zprávou po předchozí inzerující zprávě je znovu inzerující zpráva s novou cestou nebo zpráva o nedostupnosti destinace.

Při stavu E_{longer} výpadku cesty je nutné rozšířit mezi zbývající uzly obsažené v původní $ASpath$ delší cestu. Zde nastává problém s tím, že jako slepá cesta je brána každá delší cesta. Po jejím neoprávněném stažením je nutné tuto cestu znovu rozšířit mezi tyto uzly, což prodlužuje dobu konvergence a narůstá počet rozesílaných zpráv. Rychlost stabilizace ze stavu E_{longer} tedy závisí na tom, jak rychle se síť zbaví slepých cest a kolik času zabere oznámení alternativní cesty v celé síti.

Rychlost stabilizace po stavech E_{down} nebo $E_{shorter}$ závisí na rychlosti odstranění slepých cest ze sítě, ale už nezávisí na propagaci nové cesty, proto v těchto případech má význam metodu Ghost-Flushing používat.

3.9. Metoda Ghost-Buster

U předchozí metody Ghost-Flushing jsme dokázali, že doba konvergence závisí na délce nejdelší jednoduché cesty $ASpath$, která se ve směrovacích tabulkách uzlů sítě vyskytla. Následující metoda Ghost-Buster, která byla taktéž představena v článku [19], vylepšuje dobu konvergence tak, že doba konvergence závisí na délce nejdelší jednoduché cesty, která existovala v síti před selháním některé destinace, což odpovídá průměru sítě d.

Tato metoda redukuje ještě více dobu konvergence oproti metodě Ghost-Flushing tak, že v MRAI časových intervalech neprobíhá rozesílání nových cest. Neustále však probíhá odstraňování cest k neexistujícím destinacím pomocí čistících zpráv o nedostupnosti destinace.

Odesílání dané cesty je pozdrženo až do uplynutí času δ, který se začne běžet od přijetí dané cesty a jejího vložení do odpovídající tabulky Adj-RIB-In. Po vypršení tohoto času se tato cesta odešle jen v případě, jestliže tuto cestu má BGP uzel ve své směrovací tabulce Loc-RIB. To eliminuje možnost odeslání již neplatné cesty a šíření tak slepých cest.

Před algoritmus metody Ghost-Flushing přidáme ještě navíc tento algoritmus:

```
BGP uzel oznámí novou cestu jeho sousedům,
 když sám přijal Update zprávu s touto cestou
  nejméně před delta sekund,
 jinak oznámení pozdrží až do uplynutí delta sekund.
```

3.9.1. Výpočet časové a komunikační složitosti

Vzhledem k tomu, že metoda Ghost-Flushing nejvíce vylepšuje dobu konvergence 3.8., jen když některá destinace přestala být v síti dostupná, se budeme zabývat výpočtem časové a komunikační složitostí konvergence ze stavu E_{down}.

Nejprve si určíme číslo $K = \frac{\delta+h}{h}$, který vyjadřuje poměr mezi časem doručení zprávy inzerující cestu a zprávy oznamující nedostupnost destinace mezi 2 sousedními BGP uzly. Parametr h představuje průměrný čas doručení zprávy mezi 2 sousedními BGP uzly a δ představuje časové zpoždění při odeslání zprávy inzerující cestu. Parametr δ je výhodné dle [19] zvolit $\delta \simeq MinAdverTime$.

Díky metodě Ghost-Buster naroste dosud nejdelší délka jednoduché cesty (která je slepou cestou) pouze jednou v čase $\delta+h$, což odpovídá času Kh. V každé časové jednotce h zmizí ze sítě slepé cesty s délkami menšími než jaká je délka nejdelší cesty, která se zrovna v síti vyskytuje.

V článku [19] odvodili funkci největší časové složitosti konvergence ze stavu E_{down}

$$O(hd\frac{K}{K-1}) \tag{6}$$

kde d je délka nejdelší jednoduché cesty v síti před selháním některé destinace, h je průměrný čas průchodu zprávy mezi dvěma sousedními BGP uzly a K je poměr rychlostí doručení zprávy inzerující cestu a zprávy oznamující nedostupnost destinace.

Negativem této metody je to, že při nalezení nové destinace v síti (stavu E_{up}) bude trvat delší dobu (kvůli dodatečnému zpoždění) než se všechny BGP uzly v síti dozví, že existuje. Platí to i v případě stavu $E_{shorter}$, když by některý uzel začal směrovat kratší cestou než doposud. Pro konvergenci ze stavu E_{longer} platí totéž, co u metody Ghost-Flushing.

Výpočet komunikační složitosti byl v článku [19] proveden obdobně jako u metody Ghost-Flushing, protože i tady platí, že během jednoho časového intervalu MRAI mohou být odeslány nejvýše 2 zprávy. Komunikační složitost pro dosažení

stabilního stavu ze stavu E_{down} se spočítá pomocí

$$O(\frac{2hdhK}{minRouteAdver(K-1)}) \tag{7}$$

kde d je délka nejdelší jednoduché cesty v síti před selháním některé destinace, h je průměrný čas průchodu zprávy mezi dvěma sousedními BGP uzly, K je poměr rychlostí doručení zprávy inzerující cestu a zprávy oznamující nedostupnost destinace, a E je počet hran grafu sítě.

3.10. Metoda konzistentních pravidel (Consistency Assertions)

Při změně topologie sítě, kdy dojde k výpadku některé destinace, sousední uzly, které s uzlem vlastnící tuto destinaci mají otevřenou BGP relaci, začnou hledat cestu přes jiné sousedy. Z předchozí metody Ghost-Flushing víme, že tyto nalezené cesty jsou rovněž neplatné, protože se jedná o slepé cesty, které by měly být co nejdříve odstraněny.

Metoda konzistentních pravidel (Consistency Assertions), která byla představena v článku [20] umožňuje BGP uzlu detekovat a ignorovat cestu inzerovanou v příchozí zprávě, pokud nesplní dané podmínky. Tak zabraňuje směrování pomocí této cesty a následně i jejímu inzerování dál.

Podmínky, podle kterých se určuje, která cesta bude povolena nebo potlačena pro výběr nejlepší cesty byly definovány pro protokol Simple Path Vector Protocol (SPVP), který je teoretickou zjednodušenou verzí protokolu BGP, v [20].

Tyto podmínky jsou postaveny na vlastnosti konzistentnosti dvou cest. Dvě cesty ke stejné destinaci jsou konzistentní, jestliže je splněna jedna z následujících podmínek:

- Obě cesty jsou prázdné.
- Obě cesty, které jsou neprázdné, nemají žádný společný uzel (kromě destinace).
- Obě cesty, které jsou neprázdné, se protínají v některém společném uzlu, vedou obě od tohoto uzlu stejnou cestou do destinace.
- První cesta je prázdná a druhá cesta je neprázdná, a v neprázdné cestě neexistuje uzel, který oznámil nedostupnost destinace (prázdnou cestu).

Dále se používá vztah platnosti (validity) cesty. Cesta je platná (validní), jestliže žádný uzel v této cestě dosud nezměnil svou nejvíce preferovanou cestu k destinaci na jinou cestu. V případě, že dvě cesty nejsou konzistentní, není jednoznačně určeno, jestli nejsou platné obě cesty nebo jenom jedna. Podle věty

platí, že když nejsou dvě cesty konzistentní a odesílatel první cesty je obsažen v druhé cestě, pak je druhá cesta nežádoucí [20].

Požaduje se, aby z nežádoucích cest nebyla vybrána nejlepší cesta pro danou destinaci do směrovací tabulky. Neplatné cesty jsou při výpočtu nejlepších cest úplně ignorovány a tyto cesty se mezi uzly nešíří. Jako nežádoucí cesta může být označena i platná cesta, která je delší než jiná cesta k destinaci, protože by stejně nebyla vybrána jako optimální cesta do směrovací tabulky během výpočtu nejlepší cesty k destinaci. Ignorování platných cest, které jsou označeny jako nežádoucí, má pozitivní vliv na zkrácení doby konvergence, pokud všechny uzly vybírají nejlepší cesty jen podle délky $ASpath$, protože v tomto případě jsou jako nežádoucí cesty označovány cesty s delší $ASpath$.

3.10.1. Konzistentní pravidla

Tyto pravidla, která tvoří základ metody konzistentních pravidel byla definována teoreticky a popsána v [20].

Podmínka pro zprávy o nedostupnosti destinace
Pokud byla přijata zpráva o nedostupnosti destinace od uzlu N_{lost}, pak každá cesta do destinace přes jiné sousedy je označena za nežádoucí, jestliže se N_{Lost} vyskytuje v této cestě.

Podmínka pro zprávy inzerující cestu
Pokud byla přijata cesta k destinaci v Update zprávě, kterou inzeruje uzel N_{change}, pak se provede se následující:

- Pokud se uzel N_{change} vyskytuje v nějaké cestě ke stejné destinaci přes jiného souseda, označ ji jako nežádoucí.
- Pokud se nějaký jiný soused než N_{change} vyskytuje v cestě, kterou inzeroval N_{change}, pak označ tuto cestu jako nežádoucí.

Po přijetí zprávy o nedostupnosti destinace si příjemce ověří, zda některé alternativní cesty do destinace přes ostatní sousedy neobsahují uzel, který poslal tuto zprávu, a pokud ano, označí se tyto cesty jako nežádoucí.

Po přijetí zprávy s inzerovanou cestou si příjemce ověří, zda existující alternativní cesty k destinaci neobsahují uzel, který poslal tuto zprávu a následně zda některý jiný sousední uzel není obsažen v inzerované cestě, a pokud ano, označí se tyto cesty jako nežádoucí.

3.10.2. Přizpůsobení BGP protokolu pro metodu konzistentních pravidel

Konzistenční pravidla vylepšují dobu konvergence protokolu SPVP. V případě protokolu BGP je nutné si dát pozor při jeho úpravě pro vnitřní směrování. AS

je reprezentován v SPVP jedním uzlem, zatímco v BGP je jeden AS zpravidla představován více BGP směrovači.

BGP uzel se liší od uzlu v SPVP modelu v několika pohledech - v možnosti rozdělení jednoho AS do několika logických a vnořených AS a ve způsobu doručování zpráv uvnitř jednoho AS. Při práci se směrovacími politikami je nutné přidat do zprávy o nedostupnosti destinace další atribut, který rozlišuje jestli stažení cesty probíhá z důvodu změny topologie nebo politik, protože při stažení cesty z důvodu politik není taková cesta označena jako nežádoucí.

Samotný algoritmus této metody se skládá z aplikací konzistentních pravidel na příchozí inzerující zprávy. Podle konzistentních pravidel se označují některé přijaté cesty v tabulkách Adj-RIBs-In za nežádoucí, a tím se potlačí možnost jejího výběru do směrovací tabulky Loc-RIB.

Časová a komunikační složitost této metody nebyla dosud stanovena.

3.11. Metoda určující původ změny (Root Cause Notification)

Novějším řešením pro snížení doby konvergence protokolu BGP je metoda určující původ změny (Root Cause Notification) [21], ve které doba konvergence závisí podobně jako metoda Ghost-Buster na velikosti průměru sítě. Kromě toho má dobré uplatnění při konvergenci ze stavu E_{long}.

Dva BGP uzly, které původně měly mezi sebou otevřenou BGP relaci, najdou alternativní cestu k destinacím svého bývalého souseda přes jiné své sousedy. Každý z nich je označen jako uzel s výskytem změny (root cause node). Do Update zpráv, kterými se následkem změny inzerují nové cesty, vloží svoje číslo AS do speciálního atributu. Přijemce podle toho zjistí, že tuto zpráva začal prvně rozesílat uzel s výskytem změny a odstraní alternativní cesty do dané destinace z tabulek Adj-RIB-In procházející tímto uzlem.

Vzhledem k tomu, že propagace nových cest probíhá v síti různými linkami s odlišnými vlastnostmi, je nutné zajistit jejich synchronizaci pomocí sekvenčních čísel, protože by docházet k odstraňování cest, které s touto linkou počítají, ačkoliv už byla obnovena. Každý uzel sítě si vede tabulku sekvenčních čísel, do které z každé přijaté Update zprávy s novou cestou vloží sekvenční číslo pro danou destinaci jen v případě, že je větší než jaké bylo doposud. Při každé změně sekvenčního čísla příjemce ověří cesty k dané destinaci v tabulkách Adj-RIBs-In (které rovněž jako Update zprávy obsahují další pomocný atribut pro uložení sekvenčního čísla). Pokud je sekvenční číslo některého uzlu v nějaké cestě v tabulce Adj-RIB-In menší než je aktuální sekvenční číslo příslušné k danému uzlu, pak je tato cesta odstraněna z Adj-RIB-In. Tak je zaručeno, že neplatné zprávy rychle zmizí ze sítě a sníží se doba konvergence protokolu BGP.

Nevýhoda metody detekující výskyt spočívá v náročnosti přizpůsobení této metody do protokolu BGP (tak jako u metody konzistentních pravidel byla navr-

hována pro protokol SPVP) a také rozšíření Update zpráv a tabulek Adj-RIBs-In o atributy pro uchování čísla AS uzlu indikujícího změnu a aktuální sekvenční číslo odesílatele. Rovněž nevýhodou je větší zátěž na výpočetní prostředky z důvodu udržování tabulky se sekvenčními čísly.

I přes tyto nevýhody zkoumání metody určující původ změny a její modifikací [22] stále probíhá [23].

3.11.1. Výpočet časové a komunikační složitosti

Výpočet časové a komunikační složitostí byl teoreticky proveden pomocí vět a důkazů z článku [21] na protokolu SPVP.

Pro výpočet časové složitosti bylo bráno v úvahu, že Update zpráva s největším sekvenční číslem dojde nejrychleji po existující nejkratší cestě, a tak pro konvergenci ze stavu E_{down} platí funkce největší časové složitosti

$$O(dh) \tag{8}$$

kde d je délka nejdelší jednoduché cesty v síti před selháním, h je průměrný čas průchodu zprávy mezi dvěma sousedními BGP uzly.

Pro výpočet komunikační složitosti platí, že každý uzel sítě při stabilizaci své směrovací tabulky svým sousedům odešle nejvýše jednu zprávu o nedostupnosti destinace. Komunikační složitost pro dosažení stabilního stavu ze stavu E_{down} se je tedy

$$O(E) \tag{9}$$

kde E je počet hran.

Autoři článku [21] dokonce odvodili i funkce největší časové a komunikační složitosti ze stavu E_{long}, které jsou následující:

Časová složitost:

$$O(d(2h + MinAdverRoute)) \tag{10}$$

Komunikační složitost:

$$O(E(d + \frac{2h + MinAdverRoute}{MinAdverRoute})) \tag{11}$$

kde d je délka nejdelší jednoduché cesty v síti před selháním, h je průměrný čas průchodu zprávy mezi dvěma sousedními BGP uzly a E je počet hran.

3.12. Závěrečné srovnání metod

Všechny předcházející metody pro snižování doby konvergence byly testovány svými autory v simulátorech sítí a srovnávány s protokolem BGP v síti topologie úplného grafu (ve kterém každý uzel sítě má BGP relaci s každým jiným uzlem sítě), ve kterém v jednom z uzlů došlo k výpadku destinace (stavu E_{down}) nebo výpadku jedné z cest k destinaci (stavu E_{longer}).

Po výpadku destinace v takové síti dosahuje protokol BGP nejvyšší doby konvergence a nejvyššího počtu zpráv, protože se postupně vyměňují cesty ve směrovacích tabulkách až do doby, než je nalezena nejdelší možná cesta, která prochází přes všechny uzly sítě a je nakonec stažena zprávou o nedostupnosti destinace. Metody pro snížení konvergence mají menší časovou a komunikační složitost, protože výrazně rychleji stáhnou všechny cesty, které jsou označeny jako slepé nebo nejsou v síti vůbec inzerovány, protože jsou označeny jako nežádoucí a tím významně urychlují stabilizaci celé sítě a redukují počet zpráv.

V článku [21] změřili a srovnali vzájemně metody Ghost-Flushing, metody konzistentních pravidel a metody určující původ změny v síti topologie úplného grafu a internetové topologie pomocí simulací v SSFnet [29]. V topologii úplného grafu se doba konvergence snížila v porovnání s metodou Ghost-Flushing až dvakrát užitím metody konzistentních pravidel a metody určující původ změny. Dobu konvergence v topologiích odvozených z reálné internetové topologie nejvíce snižuje metoda určující původ změny. Ze všech uvedených metod má největší dobu konvergence metoda konzistentních pravidel.

Analyticky byly odvozeny časové a komunikační složitostí stabilizace směrovacích tabulek ze stavu E_{down} kromě metody konzistentních pravidel, u které složitost zatím nebyla stanovena. Jejich přehled ukazuje tabulka 2.

Modifikace	**Časová složitost**	**Komunikační složitost**
BGP s $MRAI = 30$	$30n$	nE
BGP s $MRAI = 0$	nh	$n!E$
Ghost-Flushing	nh	$\frac{2Ehn}{30}$
Ghost-Buster	$\frac{kdh}{K-1}$	$\frac{2EKdh}{30(K-1)}$
Route Cause Notification	dh	E

n - nejdelší cesta, kterou má směrovač k destinaci, která selhala
d - nejdelší cesta, kterou měl směrovač k destinaci před jejím selháním
K - poměr mezi rychlostí doručení zprávy inzerující cestu a zprávy informující o nedostupnosti destinace ($K = \frac{\delta+h}{h}$), kde *delta* je čas zpoždění odesílání zpráv s inzerovanou cestou

Tabulka 2.: Časová a komunikační složitost stabilizace ze stavu E_{down}

4. Simulační prostředí SSFNet

4.1. Přehled simulačních prostředí a softwaru

Pro vědecké zkoumání a hledání nových poznatků o síťových protokolech se používají nejrůznější simulátory, které umožňují nasimulovat nejrůznější skutečnosti, jaké v síti mohou nastat. Mezi nejznámější simulátory sítí a BGP protokolu patří:

- GNU Zebra - opensource implementace BGP protokolu
- GTNetS - simulátor sítí obsahující implementaci BGP protokolu frameworkem BGP++ založený na GNU Zebra
- SSFNet - simulátor sítí, obsahující BGP protokol
- NS-2 s ns-BGP - simulátor založený na SSFNet implementaci obsahující rozšíření BGP pro NS-2

Pro implementaci metody pro snížení doby konvergence byl dle zadání vybrán simulátor SSFNet [29], jehož poslední verze 2.0 byla vydána 15. ledna 2004.

4.2. Popis simulačního prostředí SSFNet 2.0

SSFNet je kolekce komponent v Javě pro modelování a simulaci různých síťových protokolů a sítí od úrovně síťové vrstvy (IP paketů). Linková a fyzická vrstva může být doplněna pomocí jiných dodatečných komponent, pokud bychom jejich vlastnosti chtěli blíže zkoumat. Vzhledem k tomu, že jazyk Java je rozšířen na mnoha platformách a používá se často i na výkonných víceprocesorových počítačích, je možné, aby výpočetní procesy simulátoru SSFNet pro reálnější přiblížení, probíhaly paralelně.

SSFNet modely sítí jsou samostatně konfigurovatelné, tzn. že každá instance třídy SSFNet (kompletního modelu sítě) se konfiguruje pomocí jednoho konfiguračního souboru (může být jich i více), který může být dostupný jak lokálně, tak i na libovolném místě v Internetu. Tyto konfigurační soubory jsou textové soubory ve formátu DML (Domain Model Language), který má jednoduchou syntaxi. Dokumentace [30], jak lze napsat DML soubor a správně nakonfigurovat model sítě, je součástí balíčku s originálním zdrojovým kódem a knihovnami SSFNet, který je možné stáhnout z [29]. Ukázka toho, jak takový soubor vypadá, se nachází v příloze F. na konci této práce. Je možné zautomatizovat výrobu modelů sítí pomocí nejrůznějších skriptů, protože se jedná o textové soubory s danou syntaxí. Pro potřeby této práce jsou k dispozici několik takových skriptů, které jsme napsali v jazyce Python.

Komponenty SSFNet, pomocí nichž můžeme modelovat a simulovat sítě jsou organizovány do dvou odvozených frameworků, do SSF.OS (pro modelování uzlů sítě a vlastností jejich operačního systémů a především síťových protokolů) a do SSF.Net (pro modelování síťového spojení, vytváření uzlů a jejich propojení).

Frameworky SSF.OS a SSF.Net jsou zapouzdřené do SSF API, jejíž detaily lze nalézt v [29]. Implementace BGP protokolu se nachází v SSF.OS.BGP4 a funkčně odpovídá definici protokolu BGP podle RFC 1771 [2].

4.3. Spuštění simulátoru SSFNet 2.0

Pro spuštění simulátoru SSFNet je třeba mít v operačním systému nainstalován Java Virtual Machine verze 1.2 nebo novější a je nutné mít v systémové proměnné `CLASSPATH` cesty na knihovny simulátoru SSFNet. Simulaci v simulátoru SSFNet pak spouštíme podle [30] následovně:

```
java SSF.Net.Net <runtime> <dmlfile1> [<dmlfile2> ...]
```

kde runtime označuje dobu běhu simulace a dmlfile1, dmlfile2... DML soubory s konfiguracemi modelu sítě a jejími vlastnostmi.

4.4. Implementace metody Ghost-Flushing

Implementace metody Ghost-Flushing byla prováděna ve třídě BGPSession a z větší části odpovídá implementaci této metody v [32].

V 2. fázi výpočetního procesu jsme zavedli detekci slepých cest, protože se zde pracuje s tabulkou Loc-RIB. Jedním ze vstupních parametrů této fáze je tabulka změn cest, která obsahuje nově nalezené cesty (buď úplně nové nebo náhrady za již existující cesty) a také nová datová struktura GhostData, která obsahuje destinace stažených cest a délky jejich *ASpath*. V cyklu této fáze je v tabulce změn považována za slepou cestu cesta, která je delší než stávající cesta k sstejné destinaci v Loc-RIB. Informace o slepých cestách jsou zaznamenávány do pole GhostData typu třídy Ghost, do které jsou ukládány NLRI a k nim délky *ASpath* stávajících a nových cest.

V následující 3. fázi výpočetního procesu, ve které se sestavují připravují a inzerují změny provedené v tabulce Loc-RIB. Destinace z pole GhostData, ke kterým se zvětšila délka *ASpath*, jsou okamžitě rozeslány ve zprávách oznamující nedostupnost destinace okolním sousedům (protože na tyto zprávy se MRAI časovač nevztahuje).

4.4.1. Třída Ghost

Pro implementaci metody Ghost-Flushing jsme přidali novou třídu Ghost, která se používá pro uchování informací o cílových destinacích a jejich délek ASpath k nim z Update zpráv. Obsahuje funkci pro přidání nové destinace a

délek Aspath nové a původní cesty a funkci pro zjištění, zda cesta k destinaci je slepá.

4.4.2. Modifikace ve třídě BGPSession

Detailnější přehled implementace metody Ghost-Flushing v systému SSFNet ukazuje následující pseudokód.

```
ArrayList decision_process_2(ArrayList changedroutes,
                             ArrayList GhostData) {
   ArrayList locribchanges = seznam změn v Loc-RIB pro 3. fázi VP
   foreach (changedroutes[i]) {
      detekce smyček v changedroutes[i]
      if (je smyčka) {
         odstraň cestu changedroutes[i] z Loc-RIB
         najdi nejlepší cestu přes jiného souseda
         z tabulek Adj-RIBs-In a vlož ji do Loc-RIB

         //Ghost-Flushing
         if (ASpath nové cesty je delší než původní cesty)
           vlož do GhostData destinaci a novou i starou délku ASpath
         //konec Ghost-Flushing

      } else (neni smyčka) {
         zjištění zda cesta odpovídá nastaveným politikám
         if (cesta changedroutes[i] odpovídá politikám)
            proběhne route flap damping, pokud je povolen
            if (cesta changedroutes[i] je lepší
                než aktuální cesta v Loc-RIB (curinfo)) {
               nahraď ji v Loc-RIB (ozn. info)

               //Ghost-Flushing
               if (ASpath nové cesty je delší než původní cesty) {
                 najdi zda destinace již je v GhostData
                 if (destinace existuje){
                   aktualizuj jen délku ASpath
                 }else{
                   vlož do GhostData destinaci a novou
                   i starou délku ASpath
                 }
               }
              //konec Ghost-Flushing
          }
```

```
      }
      return locribchanges;
}

void decision_process_3(ArrayList locribchanges,
                        ArrayList GhostData) {
   HashMap wds_tbl = seznam destinací ke stažení
   HashMap ads_tbl = seznam cest k propagaci
   vlož do wds_table destinace, které byly odstraněny z Loc-RIB
   vlož do ads_table cesty, které mohou být oznámeny sousedům

   //Ghost-Flushing
   foreach (všechny destinace dest z GhostData)
    if (posledně vložená cesta k dest do Loc-RIB
        má delší ASpath než byla původní ASpath)
      vlož dest do wds_table
     rozešli okamžitě svým sousedům zprávy
     oznamující nedostupnost destinace
   //konec Ghost-Flushing

   rozešli sousedům Update zprávami obsah tabulek
   wds_table a adv_table
}
```

4.4.3. Aktivace metody Ghost-Flushing v DML souboru

Do metody config() ve třídě BGPSession jsme přidali podporu metody Ghost-Flushing pro její aktivaci v DML souboru. Je možné ji aktivovat dvěma způsoby:

- v sekci pro globální konfiguraci BGP protokolu (`bgpoptions`)
- v sekci pro lokální konfiguraci uzlu (`ProtocolSession use SSF.OS.BGP4 .BGPSession`), pak tato metoda poběží jen při běhu výpočetního procesu v daném uzlu.

V obou případech lze metodu aktivovat nebo deaktivovat pomocí

```
ghostflushing true | false
```

Pro vypisování dodatečných informací během užití metody Ghost-Flushing je někdy výhodné použít v sekci pro globální konfiguraci BGP protokolu (`bgpoptions`)

```
ghostdebug true | false
```

Výchozí hodnota obou proměnných je v SSFNet nastavena na false.

4.5. Implementace metody Ghost-Buster

Implementace metody Ghost-Buster byla rovněž jako v předešlém případě prováděna ve třídě BGPSession a z větší části odpovídá implementaci této metody v [32].

Implementovali jsme nové datové struktury. Nejprve při vložení nové cesty nebo nahrazení stávající cesty v tabulce Adj-RIB-In se aktivuje tzv. delta časovač, který pozdrží odeslání této cesty ostatním sousedům až do uplynutí času δ, pokud by se cesta objevila následně ve směrovací tabulce Loc-RIB. Přidružení časovače k cestě a jeho aktivaci má na starosti nově implementovaná třída GhostBuster.

Samotný delta časovač je tvořen novou třídou GhostBusterTimer. Jejím hlavním úkolem je odeslat zprávu systému SSFNet informující o vypršení času nutného k pozdržení propagace této cesty. Tato zpráva je implementovaná třídou GhostBusterTimeoutMessage. Seznam cest spolu s příslušnými časovači jsou vedeny v hašovací tabulce ve třídě PeerEntry.

Při pokusu o propagaci cesty, jejíž přidružený časovač ještě běží (potřebný čas k pozdržení ještě nevypršel), je znemožněno odeslání zprávy s touto cestou. Po příchodu zprávy informující o vypršení časovače přidruženého k cestě se zjistí, zda tato cesta se nachází v tabulce Loc-RIB, a pokud ano, odešle se tato cesta v inzerující zprávě všem sousedům.

4.5.1. Třída GhostBuster

Tato třída představuje datovou strukturu, která sdružuje cestu k destinaci a delta časovač, který je instancí třídy GhostBusterTimer.

4.5.2. Třída GhostBusterTimer

Tato třída rozšiřuje třídu Timer přidáním datové položky pro cestu, která se vztahuje k delta časovači, který pozdrží propagaci Update zprávy. Obsahuje jedinou metodu callback(), která se provede po vypršení stanoveného času. Vytvoří zprávu instance třídy GhostBusterTimeoutMessage, do jejíhož konstruktoru se vloží cesta z datové položky.

4.5.3. Třída GhostBusterTimeoutMessage

Tato třída rozšiřuje třídu TimeoutMessage přidáním datové položky pro cestu, která se vztahovala k vypršenému delta časovači. Při vzniku instance se v systému SSFNet vyvolá v příslušné BGP relaci událost GhostBusterTimerExp, kterou zachytí a zpracuje metoda handle_event() z třídy BGPSession.

4.5.4. Modifikace ve třídě BGPSession

Detailnější přehled implementace metody Ghost-Buster v systému SSFNet ukazuje následující pseudokód.

```
void try_send_update(UpdateMessage msg, ArrayList senders,
                     PeerEntry peer) {
   //Ghost-Buster
   if (msg obsahuje cestu rte) {
      if (delta časovač příslušný k rte ještě nevypršel){
        odstraň cestu rte z Update zprávy msg
      }
   }
   //konec Ghost-Buster

   if (MRAI časovač ještě nevypršel) {
     vlož cestu z Update zprávy do seznamu cest
     čekajících k odeslání po vypršení MRAI časovače
     a odstraň ji z Update zprávy msg
     return; (konec procedury)
   }
   odstraň ze seznamu cest čekajících na propagaci cesty,
   které byly staženy

   pošli Update zprávu sousedu peer
}

boolean handle_event() {
   původní kód
   ...
   if (uzel je ve stavu ESTABLISHED){

      //Ghost-Buster
      if (je zachycena interní zpráva oznamující
          vypršení delta časovače){
        ri = cesta jejíž delta časovač vypršel
        if (cesta ri je v Loc-RIB){
         odešli ji v Update zprávě svým sousedům
        }
      }
      //konec Ghost-Buster

    ...
    původní kód
```

```
  }
}
```

4.5.5. Aktivace metody Ghost-Buster v DML souboru

Do metody config() ve třídě BGPSession jsme přidali podporu metody Ghost-Buster pro její aktivaci v DML souboru. Je možné ji aktivovat globálně pro všechny uzly (`bgpoptions`) nebo v lokální konfiguraci protokolu BGP v každém uzlu zvlášť pomocí parametru

```
ghostbuster <unsigned int>
```

Hodnotou je čas delta v sekundách. Výchozí hodnotou je 0, která označuje, že metoda Ghost-Buster není aktivována.

Pro vypisování dodatečných informací během užití metody Ghost-Buster stačí použít v sekci pro globální konfiguraci BGP protokolu (`bgpoptions`) parametr `ghostdebug`.

4.6. Implementace metody konzistentních pravidel

Úpravy implementace SSFNet pro metodu konzistentních pravidel byly prováděny ve třídě BGPSession a byla inspirována [20].

V 1. fázi výpočetního procesu, kdy se počítají ohodnocení nových a aktualizovaných cest, se provede podle pravidel označení cest, které jsou nežádoucí. Pro tento účel byla přidána hašovací tabulka Conflicts, do které se ukládají destinace a množina čísel AS, do nichž jsou cesty pro směrování přes daného souseda pro tyto AS nežádoucí. Tato datová struktura Conflicts existuje pro každého souseda zvlášť, a byla tedy implementována do třídy PeerEntry.

V 2. fázi výpočetního procesu se hledají nové nejlepší cesty jen z těch cest z tabulek Adj-RIBs-In, které nejsou nežádoucí.

Při přijetí zprávy oznamující nedostupnost destinace se musí provést ověření, zda ostatní dostupné cesty do stahované destinace (dle tabulek Adj-RIBs-In) nevedou přes BGP uzel, který oznámil její stažení.

4.6.1. Modifikace ve třídě BGPSession

Detailnější přehled implementace metody konzistentních pravidel v systému SSFNet ukazuje následující pseudokód.

```
ArrayList decision_process_1(ArrayList infolist) {
  foreach (novou/aktualizovanou cestu infolist[i]) {
    původní kód
    ...
```

```
    //Consistency Assertions
    foreach (mého souseda j) {
      RouteInfo tmpinfo = cesta do destinace infolist[i].NLRI
                          přes souseda j
      RouteInfo info = infolist[i]
      change_peer = číslo AS odesílatele cesty infolist[i]
      if (tmpinfo != null){
        if (tmpinfo obsahuje v ASpath change_peer)
            a ASpath cesty tmpinfo nekončí ASpath cesty info
          přidej change_peer do Conflict[j][destinace]
        if (ASpath cesty tmpinfo končí ASpath cesty info
            a Conflict[j][destinace] obsahuje change_peer)
          odeber change_peer z Conflict[j][destinace]
        if (info obsahuje v ASpath číslo AS souseda j
            a ASpath cesty info nekončí ASpath cesty tmpinfo)
          přidej číslo AS souseda j do Conflict[change_peer][destinace]
      }
    }
    //konec Consistency Assertions
    return changelist; //změněné cesty oproti dosavadní Loc-RIB
  }
}

void handle_update(UpdateMessage msg) {
  původní kód
  ...
  v části zachytávání zpráv oznamující nedostupnost destinace
  foreach (withdrawal[i]){
    vymaž cestu k destinaci podle withdrawal[i]
     z tabulky Adj-RIB-In patřící odesílateli

    //Consistency Assertions
    foreach (mého souseda j){
      tmpinfo = cesta do stažené destinace přes souseda j
      rmvdinfo = stažená cesta
      lost_peer = číslo AS odesílatele withdrawal[i]
      if (tmpinfo != null){
        if (ASpath cesty tmpinfo obsahuje lost_peer)
          přidej lost_peer do Conflict[j][destinace]
        else{
          foreach (zbylí sousedé k kromě j){
            if (Conflict[k][destinace] obsahuje lost_peer){
               odeber lost_peer z Conflict[j][destinace]
```

```
            }
          }
        }
      }
    }
    //konec Consistency Assertions
  }
  ...
  původní kód
}
```

Nakonec je třeba v proceduře `decission_process_2()` přidat další podmínku pro výběr nejlepší cesty k destinaci. Pro cestu, která se následně vloží do Loc-RIB, musí být `Conflict[next-hop cesty][destinace cesty]` prázdný (bez konfliktů).

4.6.2. Aktivace metody konzistentních pravidel v DML souboru

Do metody config() ve třídě BGPSession jsme přidali podporu metody konzistentních pravidel pro její aktivaci v DML souboru. Je možné ji aktivovat globálně pro všechny uzly (`bgpoptions`) nebo v lokální konfiguraci protokolu BGP v každém uzlu zvlášť pomocí parametru

```
cons_assert true | false
```

Výchozí hodnota je v SSFNet nastaven na false.

4.7. Implementace metody určující původ změny

Úpravy implementace SSFNet pro metodu určující původ změny byly prováděny ve třídě BGPSession na základě algoritmu pro protokol SPVP z [21].

Přidali jsme hašovací tabulku *seqnum* pro evidenci přijatých dosud největších sekvenčních čísel, a tuto datovou strukturu bylo rovněž třeba zavést v atributech cesty ve třídě Route. Dále se ve třídě BGPSession nachází hashovací tabulka *ts* s aktuálním sekvenčním číslem podle destinací, která se vkládají do atributu cesty v odesílaných Update zprávách.

Inkrementace aktuálního sekvenčního čísla probíhá při změně (přidání nebo nahrazení) cesty k dané destinaci ve směrovací tabulce Loc-RIB.

Při přijetí zprávy inzerující novou cestu se musí ověřit, zda inzerovaná cesta není starší než cesta, která je v tabulce Adj-RIB-In. V případě, že nová cesta z inzerující zprávy obsahuje sekvenční čísla jednotlivých AS větší nebo rovno než jsou sekvenční čísla v odpovídající Adj-RIB-In tabulce, nahradí inzerovaná cesta cestu uloženou v této tabulce. Dále proběhne srovnání sekvenčních čísel cest ke všem destinacím z Adj-RIB-In tabulek podle tabulky *seqnum*, jejichž sekvenční čísla byly přijatou zprávou změněna.

4.7.1. Modifikace ve třídě BGPSession

Detailnější přehled implementace metody určující původ změny v systému SSFNet ukazuje následující pseudokód.

```
void handle_update(UpdateMessage msg) {
  původní kód
  ...
  v~části zachytávání cest z přijatých zpráv inzerující cestu
  foreach (newinfo[i]){

    //Root Cause Notification
    changes = uzly, u kterých došlo ke zvýšení sekvenčního čísla
    seqnum = tabulka destinací a jejich dosud největších sekv. čísla

    if (seqnum neobsahuje destinaci cesty new_routes[i]){
      vlož do seqnum destinaci newinfo[i].nlri
      a příslušné sekvenční číslo newinfo[i].ts
    }
    foreach (seqnum[j]){
      if (newinfo[i].nlri obsahuje seqnum[j].nlri
          a seqnum[j].ts < newinfo[i].ts){
        vlož do seqnum destinaci newinfo[i].nlri
         a příslušné sekvenční číslo newinfo[i].ts
        vlož newinfo[i].nlri do changes
      }
    }
    if (existuje cestu ke stejné destinaci v Adj-RIB-In
        patřící odesílateli newinfo[i]){
      if (všechna sekvenční čísla z newinfo[i])
          >= sekv. číslo cesty z Adj-RIB-In){
        nahraď starou cestu v Adj-RIB-In cestou newinfo[i]
      }
    }
    foreach (destinace x z changes) {
      foreach (cesty k destinaci x v Adj-RIB-In) {
        ri = cesta k destinaci x v Adj-RIB-In
        if (ri.ts < seqnum[j].ts){
          odstraň cestu ri z Adj-RIB-In
        }
      }
    }
    //konec Root Cause Notification
```

```
  }
  ...
  původní kód
}
```

4.7.2. Aktivace metody určující původ změny v DML souboru

Do metody config() ve třídě BGPSession jsme přidali podporu metody určující původ změny pro její aktivaci v DML souboru. Je možné ji aktivovat globálně pro všechny uzly (`bgpoptions`).

```
root_cause_notification true | false
```

Výchozí hodnota je v SSFNet nastavena na false.

4.8. Další pomocné implementace v SSFNet

Pro získávání dat ze simulací jsme provedli několik pomocných implementací. Mezi ně patří vypisování, že právě došlo ke změně směrovací tabulky Loc-RIB, počítání odeslaných zpráv během simulace a zobrazování dosud největší délky cesty *ASpath* k vybranému uzlu.

V DML souboru se aktivují v globální konfiguraci protokolu BGP (`bgpoptions`) pomocí

```
show_loc_rib_change true | false
count_snd_update    true | false
max_aspath          <unsigned int>
```

Všechny nově implementované metody a možnosti jsou předkompilovány do knihovny `bgpconv.jar`. Abychom je mohli v simulacích používat je třeba před spuštění simulace vložit na začátek systémové proměnné `CLASSPATH` cestu k této knihovně.

5. Simulace sítě s různým počtem hran

V článcích, které popisovaly metody pro snížení doby konvergence [19], [20], [21] byly tyto metody testovány v simulátorech sítí v síťové topologií úplného grafu s proměnlivým počtem uzlů. V jednom z uzlů došlo k výpadku destinace (stavu E_{down}). Tento případ je záměrně uváděn proto, protože jde o nejhorší případ pro časovou a komunikační složitost konvergence protokolu BGP. Z předchozí kapitoly 3.1. víme na základě měření provedených Labovitzem et al. [9] [10], že doba konvergence a počet zpráv závisí na délce nejdelší cesty *ASpath*, která se v síti vyskytla.

Provedli jsme měření efektivnosti metod i původního BGP protokolu při různém počtu hran, od nejmenšího možného počtu hran [2] tak, aby byla zachována souvislost grafu topologie až po největší počet hran[3] - topologii úplného grafu, protože s různou hustotou hran se liší délka nejdelší cesty k uzlu s nedostupnou destinací a tím i doba konvergence a počet zpráv. Metody jsme analyzovali a srovnali jejich účinnost v různých podmínkách. Data k analýze se získaly pomocí simulací v simulátoru SSFNet, ve kterém byly implementovány metody pro snížení doby konvergence, které byly popsány v předchozích kapitolách. Při každé simulaci byly měřeny doba konvergence, počet zpráv a délka nejdelší cesty *ASpath* k nedostupné destinaci.

Velikost sítě byla zvolena 10, 20, 50 a 100 uzlů. K jednomu uzlu této sítě byl připojen další uzel, který představuje destinaci, která v průběhu simulace se stane nedostupnou. Během stabilizace ze stavu E_{down} je potřeba stáhnout destinaci.

Měření a získávávání výsledků probíhalo následujícím způsobem. Nejprve byl spuštěn generátor topologií. Na počátku byla topologie reprezentována náhodným souvislým grafem s daným počtem uzlů a nejmenším počtem hran. Postupně se náhodně přidávaly hrany - u sítí s 10 a 20 uzly po jednom, u 50 uzlů po dvaceti hranách a u 100 uzlů po 100 hranách, protože simulace a tím i získání výsledků od 20 uzlů výše trvá výrazně delší dobu. Přidávání hran pokračovalo až do přidání maximálního počtu hran. Pro stejný počet hran bylo vytvořeno 5 náhodně různých topologií (výjimku tvoří topologie úplného grafu).

Pak probíhaly simulace jednotlivých metod i původního BGP protokolu ve vytvořených topologiích. Zpoždění na všech linkách mezi jednotlivými uzly sítě bylo přibližně 1,01 sekundy.

Z výstupů simulací byly pro daný počet hran z pěti změřených hodnot spočteny průměrné doby konvergence, počty zpráv a nejdelší délky cest *ASpath* k nedostupnému uzlu, která se v síti vyskytla. Z těchto všech dat byly vytvořeny grafy pro 10, 20, 50 a 100 uzlů, jejichž přehled odkazů na obrázky, které příslušné grafy obsahují, je uveden v tabulce 3.

Simulace byly dále rozděleny na 4 části, ve kterých byly postupně aktivo-

[2] Nejmenší počet hran v souvislém grafu je roven $n - 1$, kde n je počet uzlů grafu.

[3] Největší počet hran v souvislém grafu je roven $\frac{n^2-n}{2}$, kde n je počet uzlů grafu.

Měřená veličina	10 uzlů	20 uzlů	50 uzlů	100 uzlů
Doba konvergence	obr. 2.	obr. 3.	obr. 4.	obr. 5.
Počet zpráv	obr. 6.	obr. 7.	obr. 8.	obr. 9.
Délka nejdelší cesty	obr. 10.	obr. 11.	obr. 12.	obr. 13.

Tabulka 3.: Rozcestník na obrázky podle počtu uzlů

vány parametry Split Horizon a Jitter, s nimiž se při teoretickém zkoumání BGP protokolu nepočítá. V praxi bývají tyto parametry defaultně ve směrovačích aktivovány, protože mají pozitivní vliv na snížení doby konvergence protokolu BGP. Tyto dva parametry působí následovně:

- Split Horizon - zabraňuje odesílání zpráv inzerující cestu původnímu odesílateli, od něj tato zpráva přišla. V DML souboru se tato volba nastavuje v části `bgpoptions` pomocí parametru `split_horizon`. Snižuje tak počet zpráv.
- Jitter - náhodné zvolení startu prvního odpočítávání časovačů MRAI, Keepalive a MASOI v každém BGP uzlu. V DML souboru se tato volba nastavuje v části `bgpoptions` pomocí parametrů `jitter_mrai`, `jitter_keepalive` a `jitter_masoi`. Je doporučeno jej používat dle RFC 1771 a 4271, protože rozkládá rovnoměrně zátěž na výpočetní prostředky při zpracování zpráv v jednotlivých uzlech sítě. Přispívá tak částečně ke snížení doby konvergence.

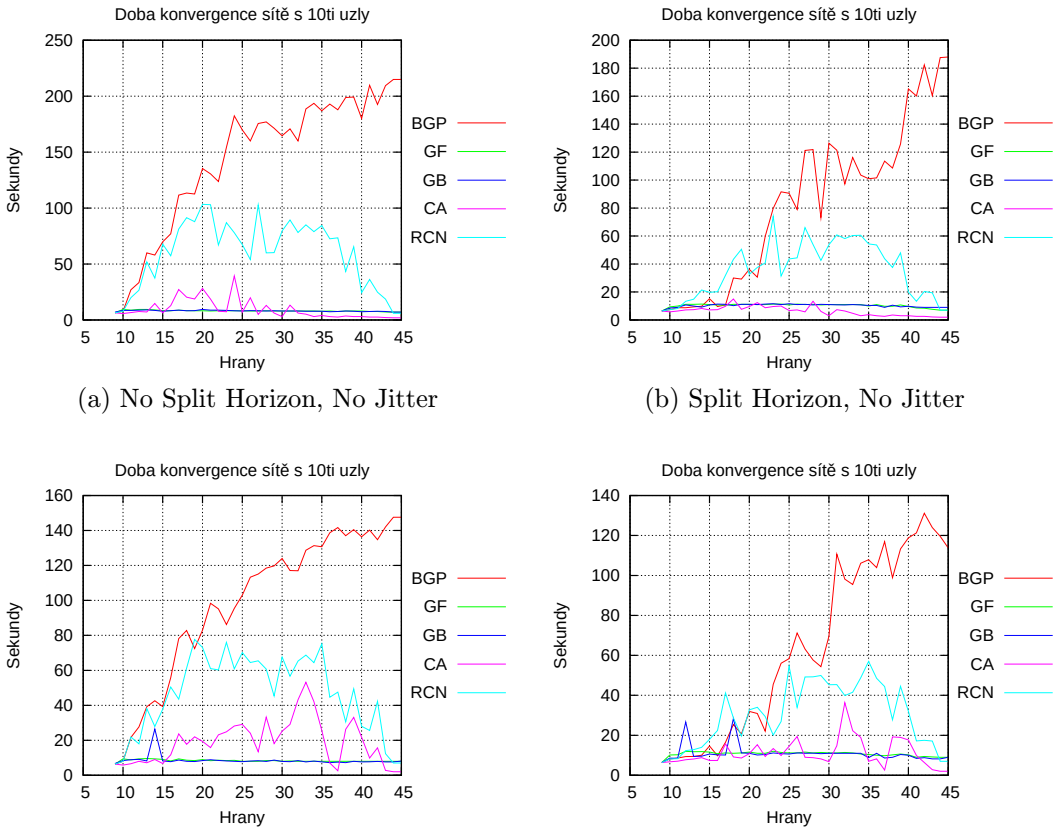
Výchozí hodnotou všech parametrů je v SSFNet true, proto nemusejí být v části `bgpoptions` v DML souboru, pokud se mají být aktivní při simulaci.

5.1. Doba konvergence

Nejdříve popíšeme získané průměrné výsledky doby konvergence (stabilizace) sítě po výpadku destinace při čtyřech možnostech nastavení parametrů Split Horizon a Jitter.

5.1.1. 10 uzlů

Grafy na obrázku 2. znázorňují, jak s rostoucím počtem hran roste logaritmicky doba konvergence protokolu BGP, která je s porovnáním s metodami pro snižování doby konvergence největší. S metodami Ghost-Flushing a Ghost-Buster je doba konvergence nezávislá na počtu hran a dává nejlepší výsledky. Doba konvergence metody konzistentních pravidel je nejmenší ze všech metod v topologii úplného grafu. Doba konvergence metody určující původ změny nejprve roste



(a) No Split Horizon, No Jitter

(b) Split Horizon, No Jitter

(c) No Split Horizon, Jitter

(d) Split Horizon, Jitter

Obrázek 2.: Doba konvergence sítě s 10ti uzly

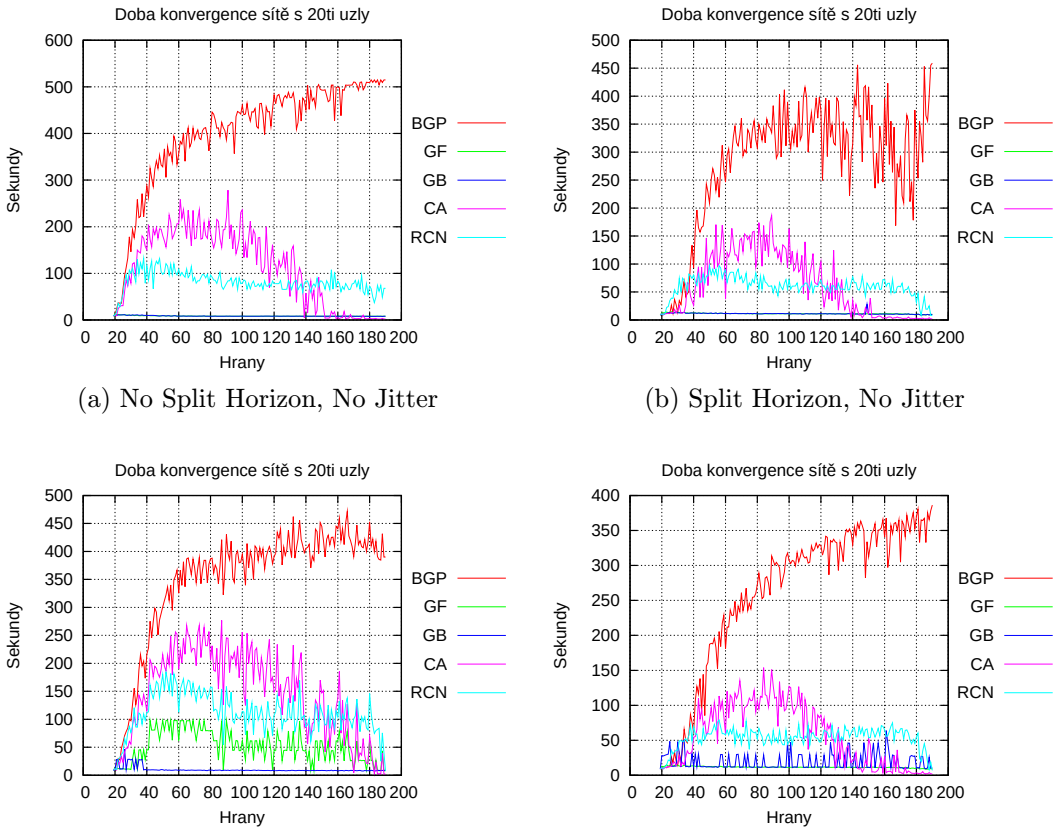
srovnatelně s protokolem BGP a pak od určitého počtu hran zůstává doba konvergence přibližně stejná.

Aktivací parametru Split Horizon (viz obrázek 2.b.) se snížila o pětinu doba konvergence protokolu BGP a všech metod a počtu hran oproti grafu na obrázku 2.a. V topologiích s malým počtem hran dává nejhorší výsledky metoda určující původ změny. Nejmenší dobu konvergence dává metoda konzistentních pravidel.

Aktivací parametru Jitter (viz obrázek 2.c.) se snížila o čtvrtinu doba konvergence většiny metod a počtu hran oproti grafu na obrázku 2.a. Doba konvergence metody konzistentních pravidel se zhoršila v topologiích s velkým počtem hran. Nejmenší dobu konvergence dávají metody Ghost-Flushing a Ghost-Buster.

Aktivací obou parametrů Split Horizon a Jitter (viz obrázek 2.d.) se snížila o třetinu doba konvergence protokolu BGP a všech metod kromě Ghost-Flushing a Ghost-Buster, ve kterých došlo ke zhoršení doby konvergence oproti grafu na obrázku 2.a.

5.1.2. 20 uzlů



(a) No Split Horizon, No Jitter

(b) Split Horizon, No Jitter

(c) No Split Horizon, Jitter

(d) Split Horizon, Jitter

Obrázek 3.: Doba konvergence sítě s 20ti uzly

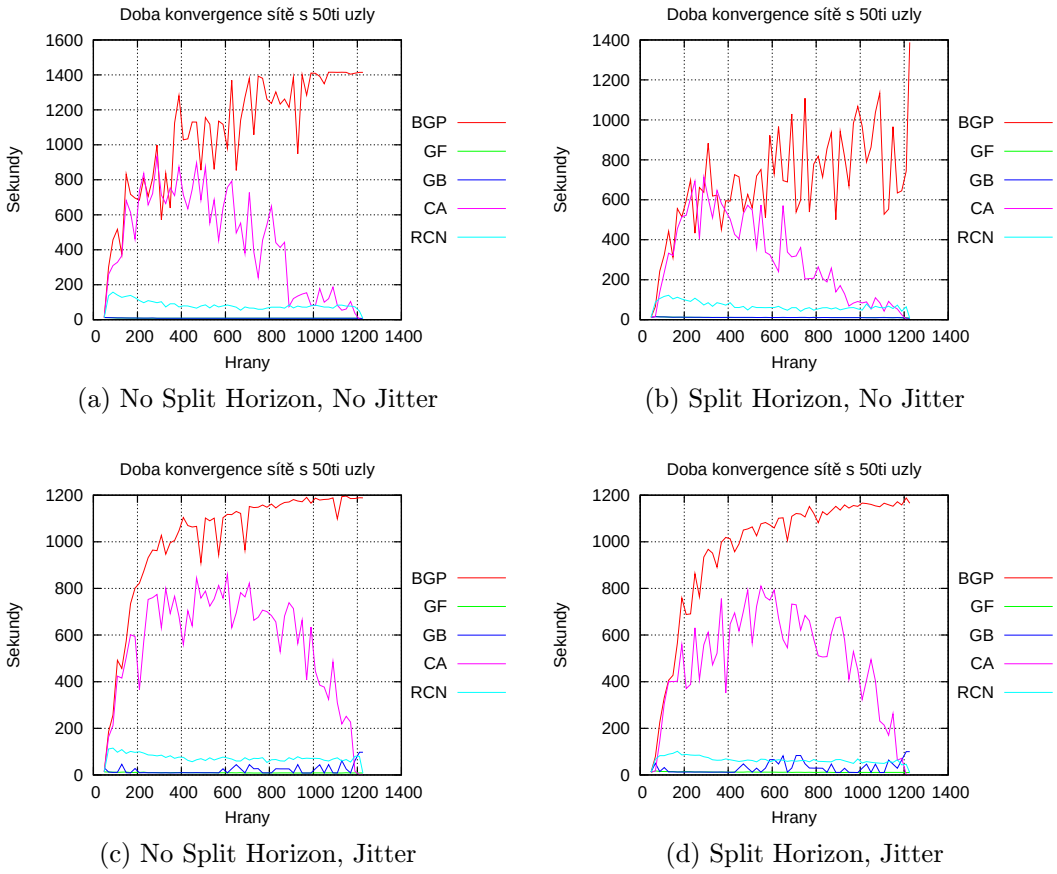
Při srovnání doby konvergence protokolu BGP a dalších metod na grafech na obrázku 3. je vidět, že došlo k nárůstu doby konvergence metody konzistentních pravidel, jejíž doba konvergence je ze všech metod pro snižování doby konvergence největší ve všech topologiích kromě topologie úplného grafu.

Aktivací parametru Split Horizon (viz obrázek 3.b.) se snížila o desetinu doba konvergence protokolu BGP oproti grafu z obrázku 3.a. V topologiích s malým počtem hran však dává horší výsledky metoda určující původ změny než samotný protokol BGP. Metoda konzistentních pravidel dává nejlepší výsledky pouze v topologiích s velkým počtem hran.

Aktivací parametru Jitter (viz obrázek 3.c.) se doba konvergence nepatrně zlepšila u protokolu BGP, ale zhoršila se u všech ostatních metod oproti grafu na obrázku 3.a.

Aktivací obou parametrů Split Horizon a Jitter (viz obrázek 3.d.) se snížila o čtvrtinu doba konvergence protokolu BGP. U metod pro snížení doby konvergence došlo ke snížení doby konvergence až o polovinu, kromě Ghost-Flushing a Ghost-Buster, u kterých došlo ke zhoršení doby konvergence oproti obrázku 3.a.

5.1.3. 50 uzlů



(a) No Split Horizon, No Jitter

(b) Split Horizon, No Jitter

(c) No Split Horizon, Jitter

(d) Split Horizon, Jitter

Obrázek 4.: Doba konvergence sítě s 50ti uzly

Na grafech na obrázku 4. došlo k nárustu doby konvergence metody konzistentních pravidel, která ve více topologiích s malým počtem uzlů odpovídá dobám konvergence protokolu BGP.

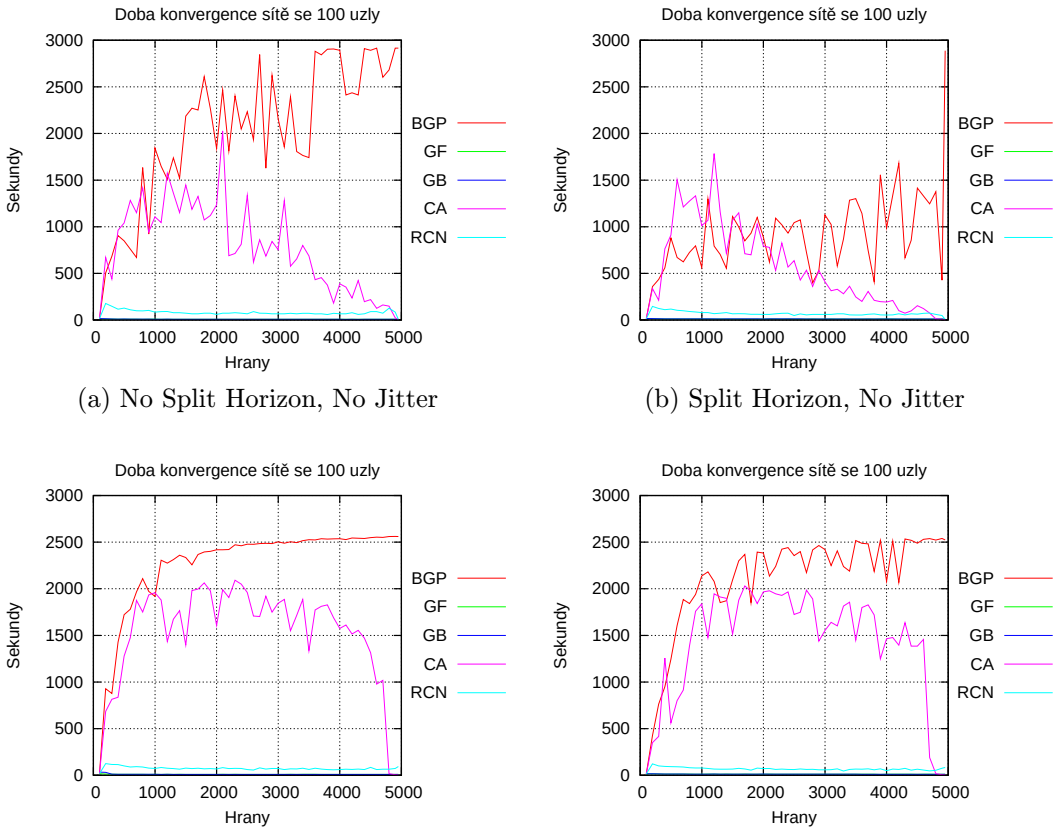
Aktivací parametru Split Horizon (viz obrázek 4.b.) se snížila o šestinu doba konvergence protokolu BGP a metody konzistentních pravidel oproti frafu na obrázku 4.a.

Aktivací parametru Jitter (viz obrázek 4.c.) se doba konvergence BGP protokolu a metody určující původ změny snížila o šestinu oproti grafu na obrázku 4.a.

Aktivace obou parametrů Split Horizon a Jitter (viz obrázek 4.d.) má na dobu konvergence téměř stejný vliv jako jen aktivovaný Jitter.

5.1.4. 100 uzlů

Srovnáním grafů na obrázku 5. s předchozími grafy se sítěmi s menším počtem



(a) No Split Horizon, No Jitter

(b) Split Horizon, No Jitter

(c) No Split Horizon, Jitter

(d) Split Horizon, Jitter

Obrázek 5.: Doba konvergence sítě se 100 uzly

uzlů doba konvergence protokolu BGP rovněž roste i přes občasné výkyvy. Doba konvergence metody konzistentních pravidel v topologiích s malým počtem uzlů převyšuje dobu konvergence protokolu BGP.

Aktivací parametru Split Horizon (viz obrázek 5.b.) se snížila téměř o polovinu doba konvergence protokolu BGP oproti grafu na obrázku 5.a., zatímco se zvýšila v topologii s úplným grafem.

Aktivace parametru Jitter (viz obrázek 5.c.) způsobila, že doba konvergence protokolu BGP jednak o šestinu klesla a zároveň růst doby konvergence je bez větších výkyvů. Zvýšila se doba konvergence metody konzistentních pravidel oproti grafu na obrázku 5.a.

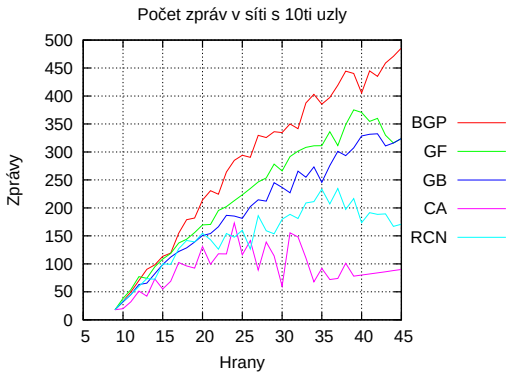
Aktivace obou parametrů Split Horizon a Jitter (viz obrázek 5.d.) má na dobu konvergence podobný vliv jako jen aktivovaný Jitter.

5.2. Počet odeslaných/přijatých zpráv

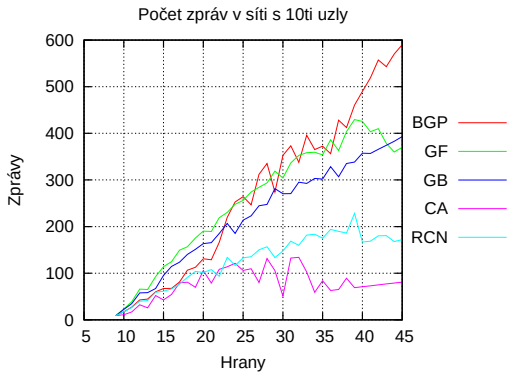
Nyní popíšeme získané průměrné výsledky počtu přijatých/odeslaných zpráv v průběhu stabilizace sítě po výpadku destinace při čtyřech možnostech nastavení

parametrů Split Horizon a Jitter.

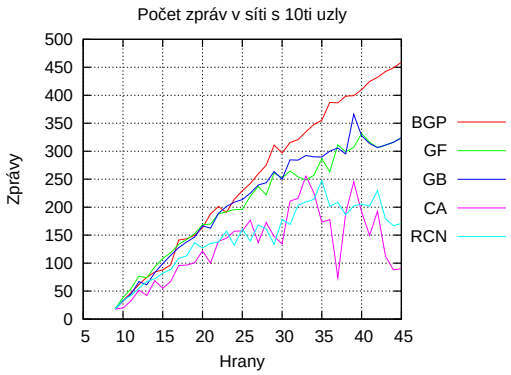
5.2.1. 10 uzlů



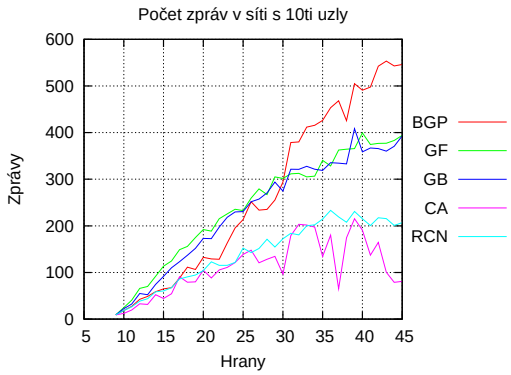
(a) No Split Horizon, No Jitter



(b) Split Horizon, No Jitter



(c) No Split Horizon, Jitter



(d) Split Horizon, Jitter

Obrázek 6.: Počet zpráv v síti s 10ti uzly

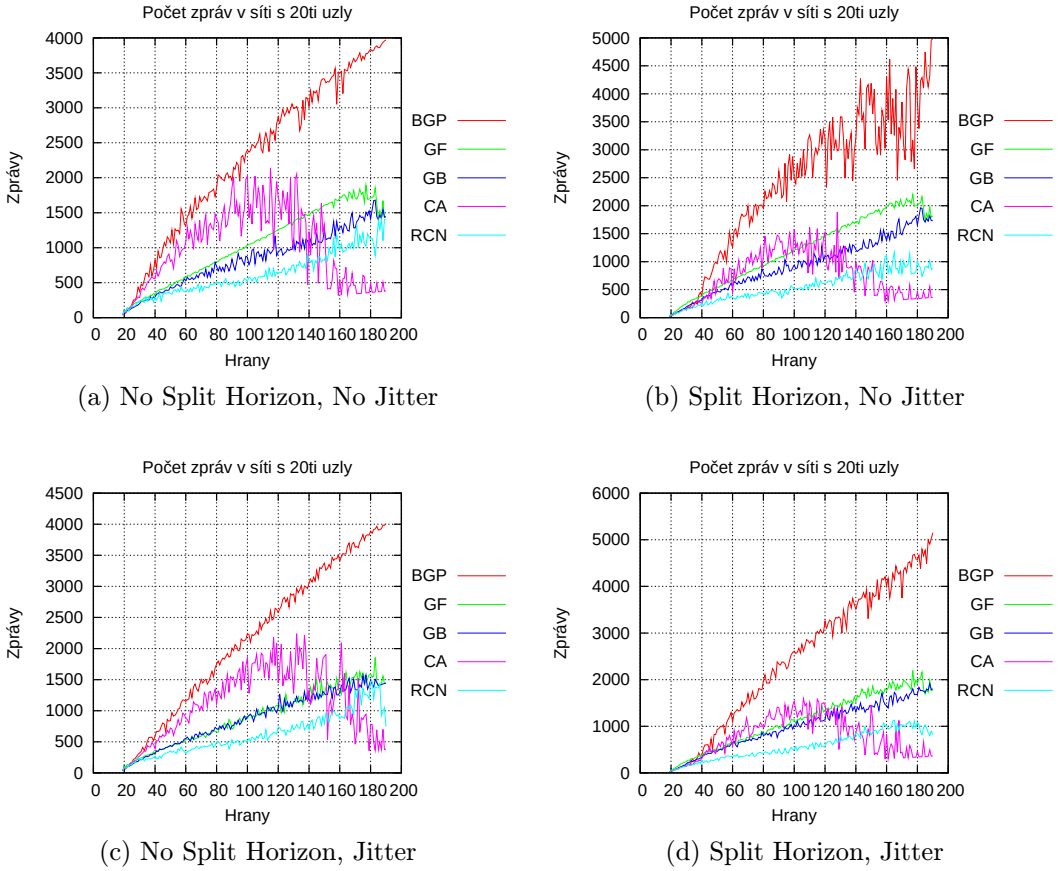
Grafy na obrázku 6. znázorňují, jak s rostoucím počtem hran rostou lineárně počty zpráv při užití protokolu BGP a metod pro snížení doby konvergence kromě metody konzistentních pravidel. Nejvíce roste počet zpráv při použití protokolu BGP a nejméně při užití metody určující původ změny. Počet zpráv u metody konzistentních pravidel roste nejprve srovnatelně s protokolem BGP a následně s rostoucím počtem hran počet zpráv klesá. Také dává nejmenší počet zpráv v porovnáním s ostatními metodami.

Aktivací parametru Split Horizon (viz obrázek 6.b.) se snížil téměř o polovinu počet zpráv u protokolu BGP a zvýšil se při užití metod Ghost-Flushing a Ghost-Buster v topologiích s malým počtem hran oproti grafu na obrázku 6.a. Zvýšil se o šestinu rovněž počet zpráv u protokolu BGP v topologiích s velkým počtem hran. Nejmenší počet zpráv dává metoda konzistentních pravidel.

Aktivací parametru Jitter (viz obrázek 6.c.) se snížil o desetinu počet zpráv u protokolu BGP a metody Ghost-Flushing, jejichž průběhy počtu zpráv jsou stejné od topologie s nejmenším počtem hran do topologií s polovinou všech možných hran. Narostl počet zpráv u metody konzistentních pravidel v topologiích s větším počtem hran, která i přesto má nejmenší počet zpráv.

Aktivací obou parametrů Split Horizon a Jitter (viz obrázek 6.d.) se snížil počet zpráv u protokolu BGP v prvních dvou třetinách možného počtu hran, jejichž počet je menší než u metod Ghost-Flushing a Ghost-Buster oproti grafu na obrázku 6.a. Zvýšil se rovněž počet zpráv u protokolu BGP v topologiích s velkým počtem hran. Nejmenší počet zpráv dává metoda konzistentních pravidel.

5.2.2. 20 uzlů



(a) No Split Horizon, No Jitter

(b) Split Horizon, No Jitter

(c) No Split Horizon, Jitter

(d) Split Horizon, Jitter

Obrázek 7.: Počet zpráv v síti s 20ti uzly

Při srovnání počtu zpráv u protokolu BGP a dalších metod na grafech na obrázku 7. je vidět, že počet zpráv u metody konzistentních pravidel se více přibližuje počtu zpráv u protokolu BGP v topologiích s malým počtem hran.

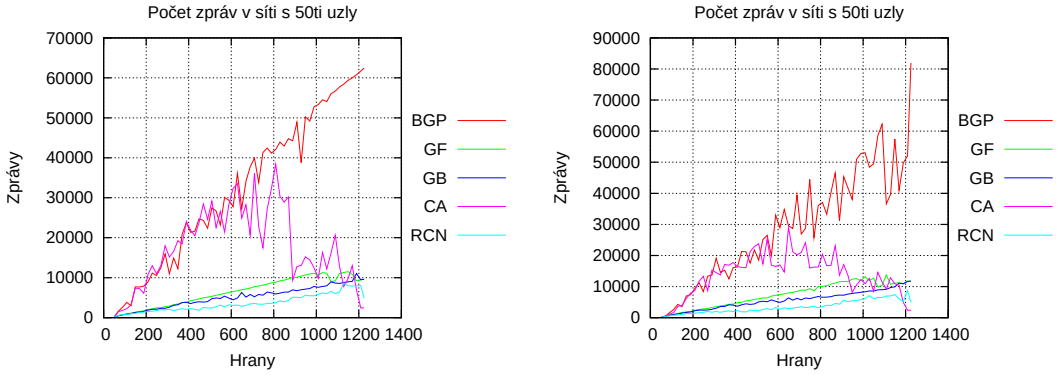
Nejmenší počet zpráv dává metoda určující původ změny následovaná metodou konzistentních pravidel v topologiích s velkým počtem hran.

Aktivací parametru Split Horizon (viz obrázek 7.b.) se o pětinu se zvýšil počet zpráv u protokolu BGP v topologiích s velkým počtem hran a snížil se o čtvrtinu počet zpráv u metody konzistentních pravidel oproti grafu na obrázku 7.a.

Aktivací parametru Jitter (viz obrázek 7.c.) se snížil počet zpráv metody Ghost-Flushing na úroveň metody Ghost-Buster oproti grafu na obrázku 7.a.

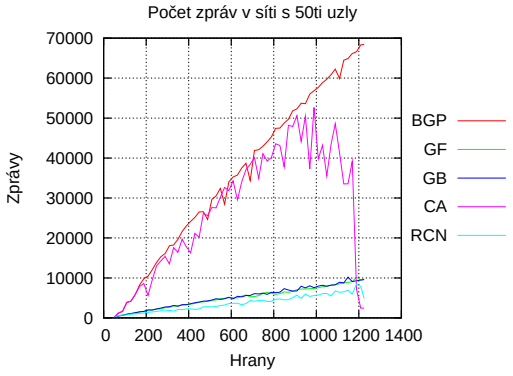
Aktivací obou parametrů Split Horizon a Jitter (viz obrázek 7.d.) se zvýšil počet zpráv u protokolu BGP a u metody Ghost-Buster oproti grafu na obrázku 7.a, zatímco u dalších metod se snížil počet zpráv.

5.2.3. 50 uzlů

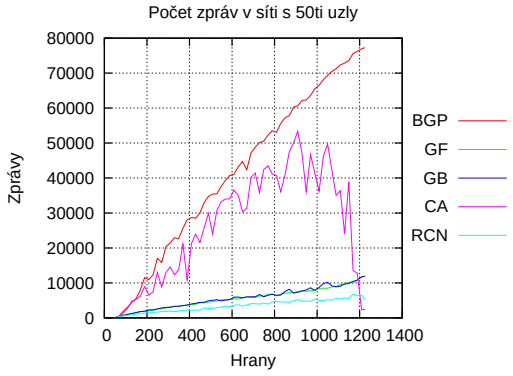


(a) No Split Horizon, No Jitter

(b) Split Horizon, No Jitter



(c) No Split Horizon, Jitter



(d) Split Horizon, Jitter

Obrázek 8.: Počet zpráv v síti s 50ti uzly

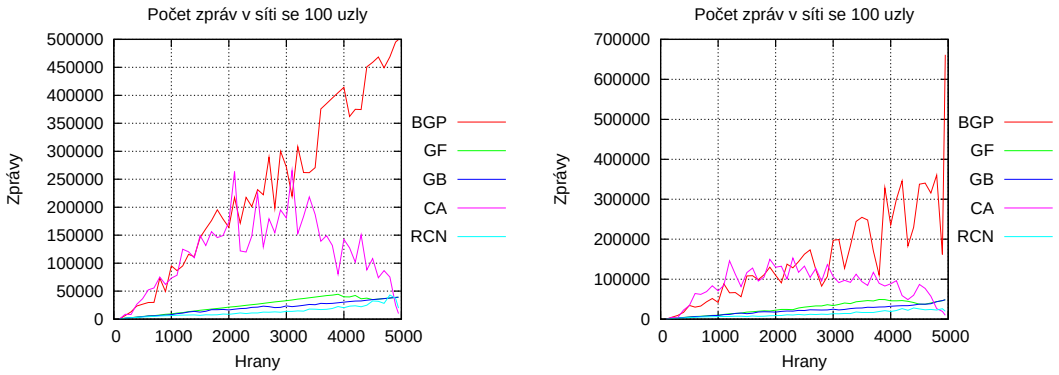
Na grafech na obrázku 8. došlo k nárustu počtu zpráv u metody konzistentních pravidel, který ve více topologiích s malým počtem uzlů odpovídá počtu zpráv u protokolu BGP.

Aktivací parametru Split Horizon (viz obrázek 8.b.) se snížil o šestinu počet zpráv u metody konzistentních pravidel oproti grafu na obrázku 8.a.

Aktivací parametru Jitter (viz obrázek 8.c.) se snížil počet zpráv metody Ghost-Flushing na úroveň metody Ghost-Buster a došlo ke zvýšení počtu zpráv u metody konzistentních pravidel oproti obrázku 8.a.

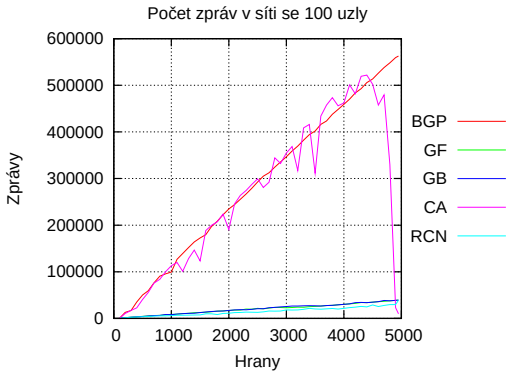
Aktivací obou parametrů Split Horizon a Jitter (viz obrázek 8.d.) došlo ke zvýšení počtu zpráv u protokolu BGP oproti obrázku 8.a., zatímco na ostatní metody mají stejný vliv jako jen aktivovaný Jitter.

5.2.4. 100 uzlů

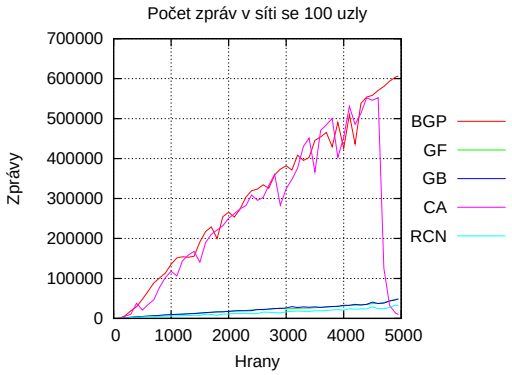


(a) No Split Horizon, No Jitter

(b) Split Horizon, No Jitter



(c) No Split Horizon, Jitter



(d) Split Horizon, Jitter

Obrázek 9.: Počet zpráv v síti se 100 uzly

Srovnáním grafů na obrázku 9. s předchozími grafy se sítěmi s menším počtem uzlů počet zpráv u protokolu BGP rovněž roste i přes občasné výkyvy. Počet zpráv u metody konzistentních pravidel v topologiích s malým počtem uzlů převyšuje dobu konvergence protokolu BGP.

Aktivací parametru Split Horizon (viz obrázek 9.b.) se snížil téměř o polovinu počet zpráv u protokolu BGP oproti grafu na obrázku 9.a., zatímco se zvýšila

v topologii s úplným grafem.

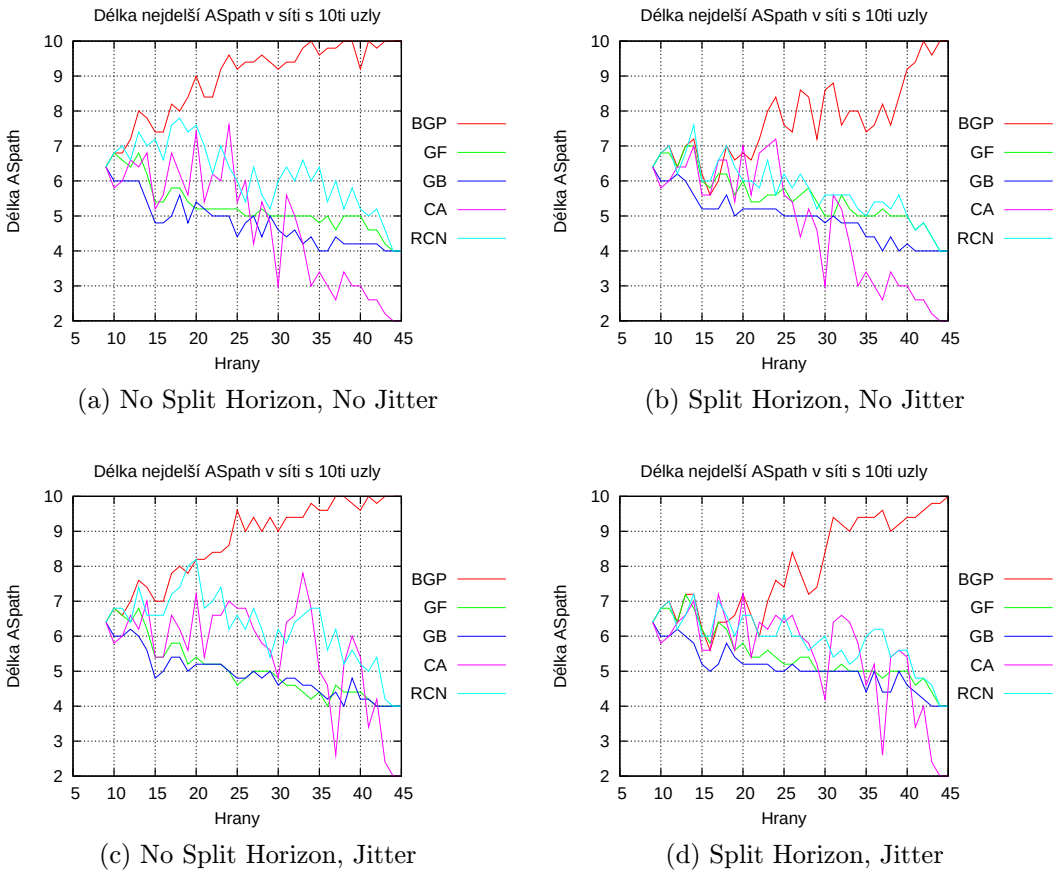
Aktivací parametru Jitter (viz obrázek 9.c.) narostl počet zpráv u protokolu BGP oproti grafu na obrázku 9.a. Počet zpráv u metody konzistentních pravidel narostl na úroveň počtu zpráv u protokolu BGP.

Aktivace obou parametrů Split Horizon a Jitter (viz obrázek 9.d.) má na počet zpráv podobný vliv jako jen aktivovaný Jitter.

5.3. Délka nejdelší cesty *ASpath*

Nakonec popíšeme získané průměrné výsledky délek nejdelších cest *ASpath*, jaké se objevily v sítí během stabilizace sítě po výpadku destinace a čtyřech možnostech nastavení parametrů Split Horizon a Jitter.

5.3.1. 10 uzlů



(a) No Split Horizon, No Jitter

(b) Split Horizon, No Jitter

(c) No Split Horizon, Jitter

(d) Split Horizon, Jitter

Obrázek 10.: Délka nejdelší cesty *ASpath* v síti s 10ti uzly

Grafy na obrázku 10. znázorňují růst délky nejdelší cesty *ASpath* k nedostupné destinaci s rostoucím počtem hran u protokolu BGP. U metod pro snižo-

vání doby konvergence je délka nejdelší cesty největší v topologiích s velmi malým počtem hran. S rostoucím počtem hran dochází k jejímu zmenšování z důvodu zabránění šíření delších alternativních slepých cest k nedostupnému destinaci. Nejvíce délku nejdelší cesty zkracuje metoda Ghost-Buster a metoda konzistentních pravidel. Metoda určující původ změny nejvíce zkracuje délku nejdelší cesty v topologiích s velkým počtem hran.

Aktivací parametru Split Horizon (viz obrázek 10.b.) došlo ke zkrácení délky nejdelší cesty u protokolu BGP oproti grafu na obrázku 10.a.

Aktivací parametru Jitter (viz obrázek 10.c.) se prodloužila délka nejdelší cesty u metody konzistentních pravidel v topologiích s větším počtem hran oproti grafu na obrázku 10.a.

Aktivací obou parametrů Split Horizon a Jitter (viz obrázek 10.d.) se zkrátila délka nejdelší cesty u protokolu BGP v topologiích s malým počtem hran, která je shodná s délkou nejdelší cesty u metod Ghost-Flushing a Ghost-Buster oproti grafu na obrázku 10.a.

5.3.2. 20 uzlů

Při srovnání délky nejdelší cesty u protokolu BGP a dalších metod na grafech na obrázku 11. je vidět, že délky nejdelší cesty u metody konzistentních pravidel se více přibližuje počtu zpráv u protokolu BGP v topologiích s malým počtem hran. Nejkratší délka nejdelší cesty má metoda Ghost-Buster následovaná metodou konzistentních pravidel v topologiích s velkým počtem hran.

Aktivací parametru Split Horizon (viz obrázek 11.b.) se zkrátila délka nejdelší cesty u protokolu BGP i u metod pro snižování doby konvergence oproti grafu na obrázku 11.a.

Aktivace parametru Jitter (viz obrázek 11.c.) nemá na délku nejdelší cesty skoro žádný vliv oproti grafu na obrázku 11.a.

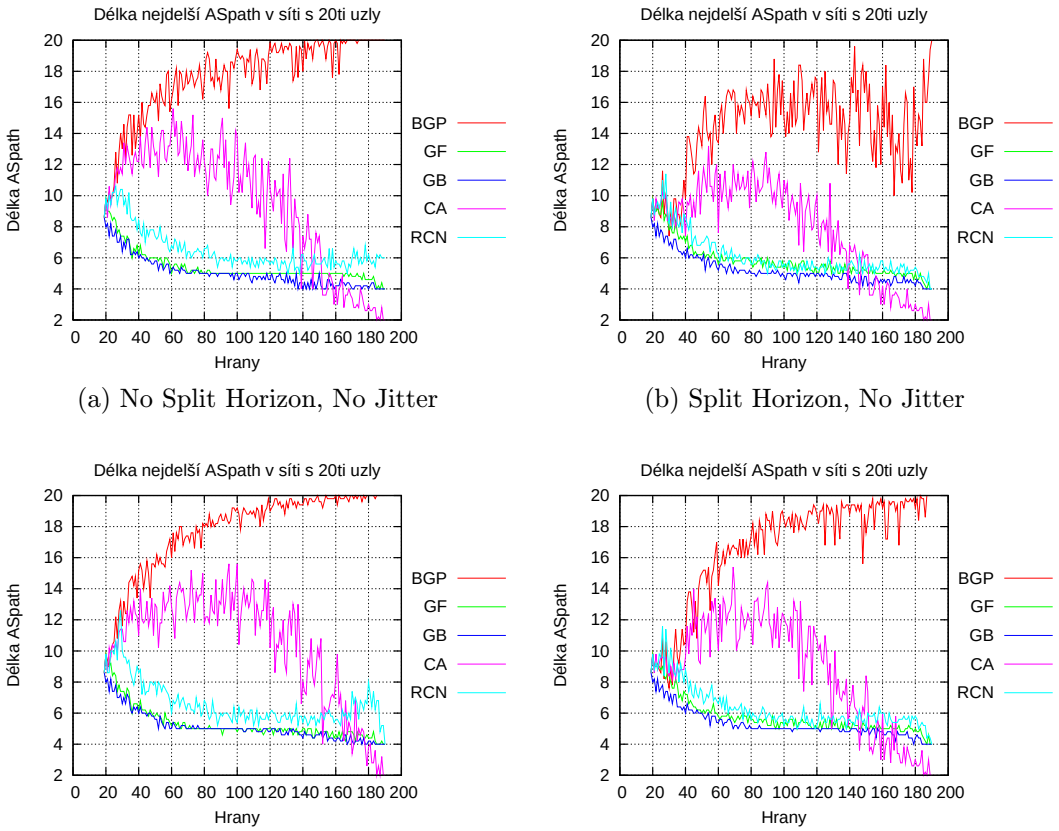
Aktivace obou parametrů Split Horizon a Jitter (viz obrázek 11.d.) má na délku nejdelší cesty podobný vliv jako jen aktivovaný Split Horizon.

5.3.3. 50 uzlů

Na grafech na obrázku 12. došlo k prodloužení délky nejdelší cesty u metody konzistentních pravidel, která ve více topologiích s malým počtem uzlů odpovídá délce nejdelší cesty u protokolu BGP.

Aktivací parametru Split Horizon (viz obrázek 12.b.) se zkrátila délka nejdelší cesty u protokolu BGP a u metody konzistentních pravidel oproti grafu na obrázku 8.a.

Aktivací parametru Jitter (viz obrázek 12.c.) se prodloužila délka nejdelší cesty u protokolu BGP a u metody konzistentních pravidel oproti grafu na obrázku 12.a.



(a) No Split Horizon, No Jitter

(b) Split Horizon, No Jitter

(c) No Split Horizon, Jitter

(d) Split Horizon, Jitter

Obrázek 11.: Délka nejdelší cesty *ASpath* v síti s 20ti uzly

Aktivace obou parametrů Split Horizon a Jitter (viz obrázek 12.d.) má na délku nejdelší cesty podobný vliv jako jen aktivovaný Jitter.

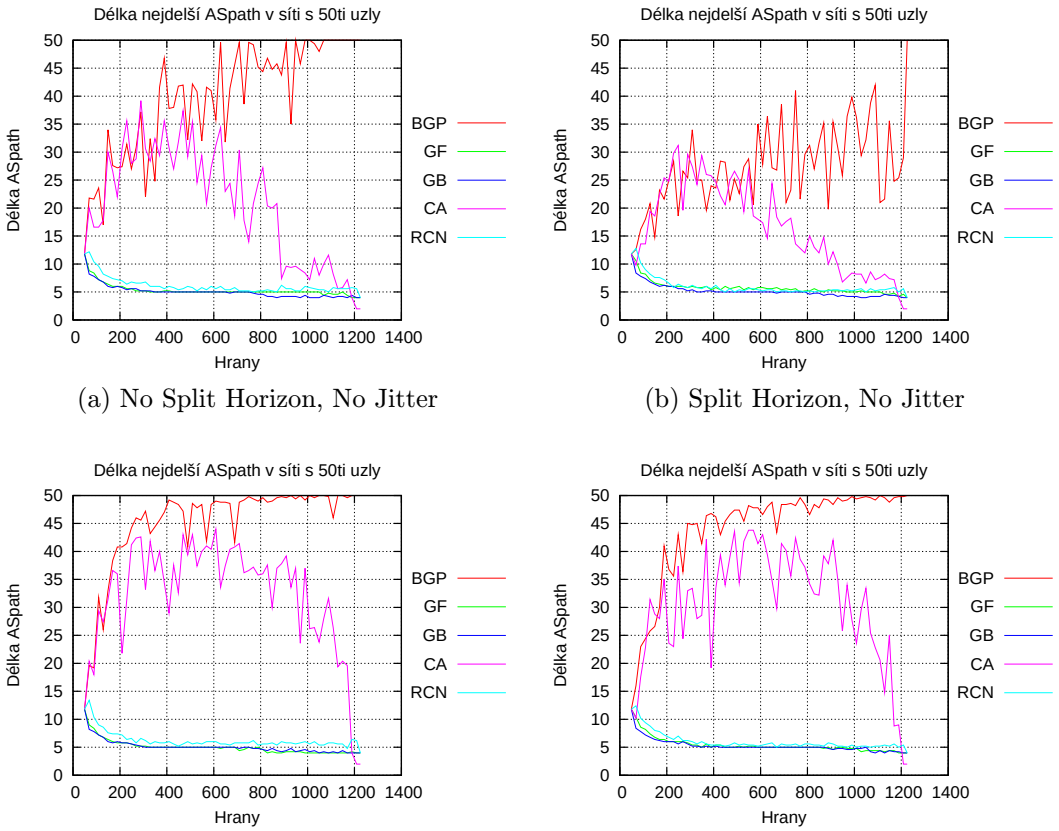
5.3.4. 100 uzlů

Srovnáním grafů na obrázku 13. s předchozími grafy se sítěmi s menším počtem uzlů délka nejdelší cesty u protokolu BGP rovněž roste i přes občasné výkyvy.

Aktivací parametru Split Horizon (viz obrázek 13.b.) se zkrátila téměř o polovinu délka nejdelší cesty u protokolu BGP oproti grafu na obrázku 13.a. Také se zkrátila délka nejdelší cesty u metody konzistentních pravidel.

Aktivací parametru Jitter (viz obrázek 13.c.) se prodloužila délka nejdelší cesty u protokolu BGP oproti grafu na obrázku 13.a. Délka nejdelší cesty u metody konzistentních pravidel se prodloužila na úroveň délky nejdelší cesty u protokolu BGP.

Aktivace obou parametrů Split Horizon a Jitter (viz obrázek 13.d.) má na délku nejdelší cesty podobný vliv jako jen aktivovaný Jitter.



(a) No Split Horizon, No Jitter

(b) Split Horizon, No Jitter

(c) No Split Horizon, Jitter

(d) Split Horizon, Jitter

Obrázek 12.: Délka nejdelší cesty *ASpath* v síti s 50ti uzly

5.4. Srovnání podle počtu uzlů

Dále jsme srovnali získané naměřené výsledky napříč sítěmi s rozdílnými počty uzlů.

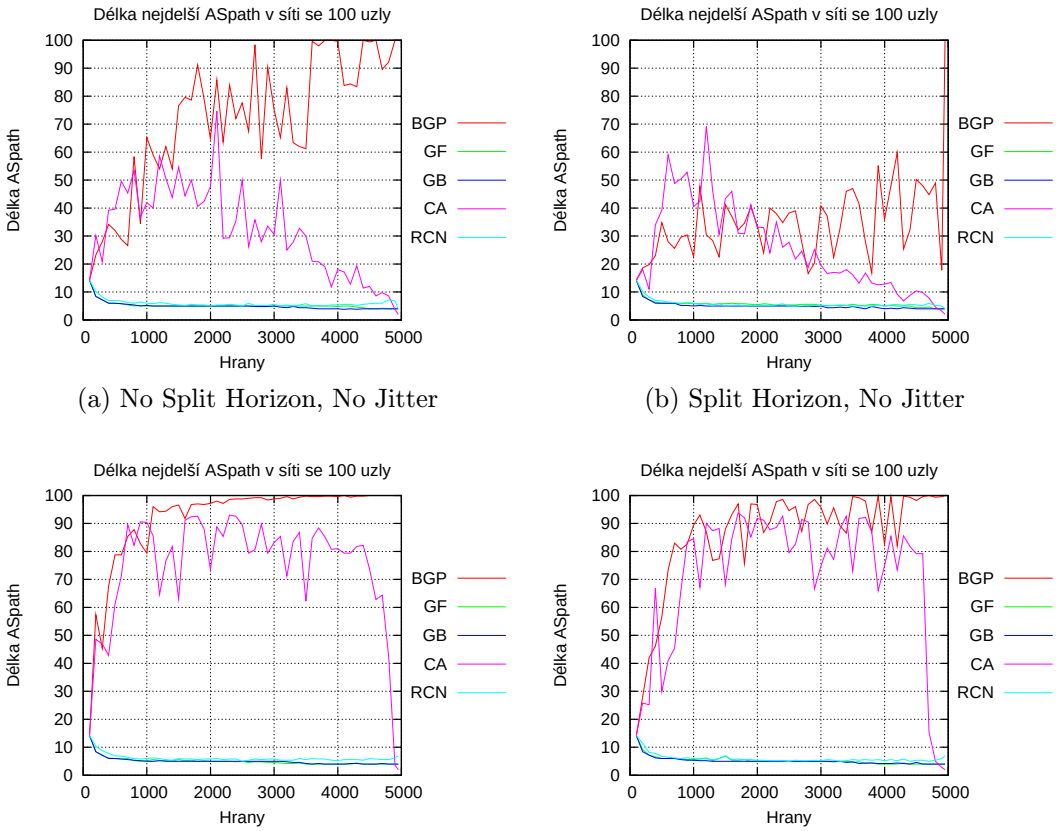
5.4.1. Doba konvergence

Grafy na obrázcích 2., 3., 4. a 5. ukazují, že s rostoucím počtem hran roste logaritmicky doba konvergence protokolu BGP.

Doba konvergence metody Ghost-Flushing a metody Ghost-Buster nezávisí na počtu hran.

Doba konvergence metody konzistentních pravidel u topologií s malým počtem hran roste srovnatelně s protokolem BGP a následně s rostoucím počtem hran pomalu klesá a nejlepší výsledky dává v síti s topologií úplného grafu. S rostoucím počtem uzlů metoda konzistentních pravidel se více přibližuje době konvergence protokolu BGP.

Doba konvergence metody určující původ změny nejprve roste srovnatelně



(a) No Split Horizon, No Jitter

(b) Split Horizon, No Jitter

(c) No Split Horizon, Jitter

(d) Split Horizon, Jitter

Obrázek 13.: Délka nejdelší cesty *ASpath* v síti se 100 uzly

s dobou konvergence protokolu BGP a pak od určitého počtu hran zůstává přibližně vyrovnaná.

Nejmenší dobu konvergence dávají metody Ghost-Flushing a Ghost-Buster. V některých topologiích sítí s 10ti uzly (viz. obrázek 2.) má nejmenší dobu konvergence metoda konzistentních pravidel.

5.4.2. Počet přijatých/odeslaných zpráv

Grafy na obrázcích 6., 7., 8. a 9. ukazují, že s rostoucím počtem hran roste lineárně počet zpráv u protokolu BGP.

U metod Ghost-Flushing a Ghost-Buster počet zpráv roste lineárně s rostoucím počtem hran. S rostoucím počtem uzlů dochází ke zvětšení rozdílu v počtu zpráv mezi těmito metodami a protokolem BGP. Zatímco v topologiích sítí s 10ti uzly (viz. obrázek 6.) se počty zpráv příliš neliší od počtu zpráv u protokolu BGP, tak v topologiích sítí se 100 uzly (viz. obrázek 9.) je vidět podstatný rozdíl vzhledem k počtu zpráv u protokolu BGP.

U metody konzistentních pravidel se počet zpráv s různým počtem uzlů liší,

kdy s rostoucím počtem uzlů počet zpráv u metody konzistentních pravidel je srovnatelný s počtem zpráv u protokolu BGP v topologiích s menším počtem hran.

U metody určující původ změny počet zpráv roste lineárně s rostoucím počtem hran a je menší než počet zpráv u metod GhostFlushing a Ghost-Buster. S rostoucím počtem uzlů se však tento rozdíl v počtu zpráv mezi těmito metodami zmenšuje.

Nejmenší počet zpráv dává metoda určující výskyt změny. V některých topologiích sítí s 10ti uzly (viz. obrázek 6.) má nejmenší počet zpráv metoda konzistentních pravidel.

5.4.3. Délka nejdelší cesty *ASpath*

Grafy na obrázcích 10., 11., 12. a 13. ukazují, že délka nejdelší cesty *ASpath* k nedostupné destinaci roste logaritmicky s rostoucím počtem hran u protokolu BGP.

U metod Ghost-Flushing, Ghost-Buster a metody určující původ změny se délka nejdelší cesty zkracuje s rostoucím počtem hran.

U metody konzistentních pravidel se délka nejdelší cesty se v porovnání s ostatními metodami nejprve prodlužuje s rostoucím počtem hran. Čím síť má více uzlů, tím více se délky nejdelších cest v topologiích s menším počtem hran přibližují délce nejdelší cesty u protokolu BGP. Naopak v topologiích s větším počtem hran se délka nejdelší cesty zkracuje s rostoucím počtem hran.

Nejmenší délku nejdelší cesty dosáhneme při užití metody Ghost-Buster. V některých topologiích sítí s 10ti uzly (viz. obrázek 10.) nejmenší délky nejdelší cesty dosáhneme při užití metody konzistentních pravidel.

5.5. Závěrečné shrnutí výsledků

Srovnáním grafů doby konvergence, počtu zpráv s grafy délek nejdelší cesty *ASpath* vidíme závislost doby konvergence a počtu zpráv na délce nejdelší cesty *ASpath* k nedostupné destinaci. U BGP protokolu je tato délka prodlužována s rostoucím počtem hran zatímco u metod pro snižování doby konvergence dochází s rostoucím počtem hran k jejímu zkracování z důvodu rychlejšímu zabránění šíření slepých alternativních cest.

Ze získaných výsledků pomocí předchozích simulací a měření plyne, že nezávisle na hustotě sítě je nejmenší doby konvergence ze stavu E_{down} je dosaženo pomocí metod Ghost-Flushing a Ghost-Buster. Metody Ghost-Flushing a Ghost-Buster jsou v případě časové konvergence a počtu zpráv přibližně vyrovnané, kromě případů, kde je lepší protokol BGP v topologiích s malým počtem hran při použitých parametrech Split Horizon a Jitter.

Vzhledem k náročnosti implementace metod do stávajícího BGP protokolu, je však výhodné implementovat metodu Ghost-Flushing, pro kterou v samotném

simulátoru SSFNet stačilo upravit jenom jednu třídu a použít navíc jen jednu datovou strukturu pro evidenci slepých cest.

Z naměřených výsledků lze si všimnout i další zajímavosti. Podle našich měření doba konvergence novější metody určující původ změny nepřekonala dobu konvergence starší metody Ghost-Flushing, ačkoliv je o tři roky novější.

5.6. Použité skripty pro simulaci a získávání výsledků

Pro běh simulací, měření a zpracování výsledků byly vyrobeny tyto skripty:

- `genMat.py` - generuje matice sousednosti grafu sítě do souborů *.mat.
- `genDML.py` - generuje DML soubory pro SSFnet podle souboru *.mat.
- `extract.py` - nalezne z výstupního souboru simulace zvolený údaj jako je čas konvergence, počet zpráv nebo délka nejdelší ASpath k nedostupné destinaci.
- `genAvgFile.py` - sestaví tabulku (v textovém souboru, ve kterém jsou záznamy odděleny mezerou) s průměrnými časy konvergence, počty zpráv nebo délek nejdelší cesty ASpath k nedostupné destinaci ze získaných naměřených dat, který je možné použít k importu do některého spreadsheetu jako je Microsoft Excel nebo OpenOffice Calc.
- `genMat.sh` - generuje matice sousednosti grafu sítě do souborů *.mat pomocí skriptu genMat.py pro zadané parametry.
- `run.sh` - z vygenerovaných souborů *.mat sestavuje odpovídající DML soubory skriptem genDML.py, spouští simulace s těmito DLM soubory pro zadané parametry a hledá ve výstupních souborech simulací požadované hodnoty skriptem extract.py. Takto naměřené hodnoty jsou ukládány do souborů ve složce `outputs`.
- `genAvgFile.sh` - sestaví tabulku ze zvolených naměřených dat pomocí skriptu gen-AvgFile.py.
- `genGraph.sh` - sestaví odpovídající graf z tabulky s průměrnými časy konvergence, počty zpráv nebo délek nejdelší ASpath pomocí Gnuplot.

Jejich podrobnější popis, včetně všech dostupných parametrů, kterými se mění chování skriptu, se nachází na konci této práce v příloze H.

Závěr

V této práci jsme nejprve podrobně popsali směrovací protokol BGP-4. Následně byl rozveden problém konvergence tohoto protokolu a chronologicky vývoj jeho částečného řešení, protože se jedná o NP těžkou úlohu. Stručně jsme popsali také nedostatky BGP protokolu, které měly v nedávné době vliv na stabilitu připojení k Internetu a dostupnosti hojně využívaných internetových služeb.

Při narušení stabilního stavu sítě vlivem topologických změn nebo směrovacích politik mezi autonomními systémy bývá silně omezena stabilita nebo dostupnost připojení k postiženému místu a tak je výhodné čas stabilizace (dobu konvergence) co nejvíce zkrátit. Pro urychlení konvergence protokolu BGP bylo navrženo několik metod jako je Ghost-Flushing, Ghost-Buster, metoda konzistentních pravidel a metoda určující původ změny.

Tyto metody jsme implementovali v simulátoru SSFNet a tyto implementace se nacházejí na přiloženém CD disku. Nakonec jsme provedli pomocí simulací těchto metod vzájemné srovnání jejich dob konvergence, počtu přijatých/odeslaných zpráv, a délek nejdelší cesty $ASpath$, která se ve směrovacích tabulkách v uzlech sítě vyskytla po selhání destinace.

Autoři jednotlivých metod prováděli analýzu a srovnání metod pomocí simulací v sítích s topologii úplného grafu, který je případem s nejhorší časovou a komunikační složitostí konvergence BGP protokolu. V této práci jsme pomocí simulací změřili dobu konvergence, počet zpráv a délku nejdelší cesty, v sítích s topologiemi s různým počtem hran od řídké sítě až po síť s topologií úplného grafu. Ze získaných výsledků je vidět závislost doby konvergence a počtu zpráv na délce nejdelší cesty $ASpath$ vyskytující se v síti. Tato délka se liší v závislosti na hustotě sítě.

Ze získaných výsledků z plyne, že nezávisle na hustotě sítě nejmenší doby konvergence ze stavu E_{down} je dosaženo pomocí metod Ghost-Flushing a Ghost-Buster, které jsou v případě časové konvergence a počtu zpráv přibližně vyrovnané. V náročnosti implementace těchto dvou metod jasně vede Ghost-Flushing, která je méně náročná na implementaci. Podle našich měření dobu konvergence metody Ghost-Flushing nepřekonala ani o tři roky novější metoda určující původ změny.

Conclusions

In this MSc. thesis we first describe details of the routing protocol BGP-4. Next we descibe the problem of convergence of the protocol and development of different particular solution, because it is a NP hard problem. We also briefly describe BGP protocol problems, which had recently led to instability effect of the Internet and the availability of frequently used Internet services.

The routing instability due to network topology changes or routing policies between autonomous systems cause limited availability of the affected area, so the stabilization time (time convergence) must be as short as possible. It were proposed several methods to accelerate the convergence of the BGP protocol such as the Ghost-Flushing, Ghost-Buster, Consistency Assertions and Root Cause Notification.

We had implemented these methods in the simulator SSFNet and these implementations can be found on the enclosed CD. Finally, we performed simulations using these methods, the correlation between periods of convergence, the number of received / sent messages, and the longest path lengths $ASpath$, which occurred after a destination failure.

The authors of the methods carried out the analysis and comparison of methods using simulations in networks with a complete graph topology, which is the case with the worst time and communication complexity of BGP protocol convergence. We measured the convergence time, the number of messages and the length of the longest path in network topologies with different number of edges from sparse networks to networks with complete graph topology. The results show that the convergence time and number of messages depend on the length of the longest path $ASpath$ occurring in the network. This length varies depending on the density of the network.

The obtained results show that, regardless of density of network the smallest convergence time from state E_{down} is achieved by Ghost-Flushing and Ghost-Buster, which in the case of convergence time and number of messages are balanced. The cost of implementing these two methods clearly favour Ghost-Flushing. According to our measurements of the time convergence show that the newer Root Cause Notification method doesn't outperform the older Ghost-Flushing method.

Reference

[1] J. W. Stewart III, *BGP4 Inter-Domain Routing in the Internet*, The Addison-Wesley, 1999.

[2] Y. Rekhter and T. Li, *A Border Gateway Protocol 4 (BGP-4), RFC 1771 (Obsoleted by RFC 4271)*, IETF, March 1995.

[3] Y. Rekhter and T. Li, *A Border Gateway Protocol 4 (BGP-4), RFC 1654 (Obsoleted by RFC 1771)*, IETF, July 1994.

[4] Y. Rekhter, T. Li, and S. Hares, *A Border Gateway Protocol 4 (BGP-4), RFC 4271*, IETF, January 2006.

[5] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, *Multiprotocol Extensions for BGP-4, RFC 4760*, IETF, January 2007.

[6] A. Heffernan, *Protection of BGP relaces via the TCP MD5 Signature Option, RFC 2385*,IETF, August 1998.

[7] C. Labovitz, G. R. Malan, and F. Jahanian, *Internet Routing Instability*, TON 1998.

[8] C. Labovitz, G. R. Malan, and F. Jahanian, *Origins of Internet Routing Instability*, Infocom 1999.

[9] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanianitz *Delayed internet routing convergence*, In Proc. of ACM SIGCOMM, volume 30, pages 175–187, October 2000.

[10] C. Labovitz, R. Wattenhofer, S. Venkatachary, and A. Ahuja, *The impact of internet policy and topology on delayed routing convergence*, In Proc. Infocom, April 2001.

[11] T. G. Griffin, and B. J. Premore *An experimental analysis of bgp convergence time*, In Proc. of ACM SIGCOMM, pages 277–288, September 1999.

[12] T. Griffin, and G. Wilfong, *An analysis of BGP convergence properties*, in Proc. ACM SIGCOMM, Aug. 1999, pp. 277–288

[13] K. Varadhan, R. Govindan, and D. Estrin, *Persistent route oscillations in interdomain routing*, Dept. of Computer Science, Univ. of Southern California, Los Angeles, USC CS TR 96-631, 1996

[14] T. G. Griffin , and A. J. T. Gurney, *Increasing bisemigroups and algebraic routing*, Proceedings of the 10th international conference on Relational and kleene algebra methods in computer science, and 5th international conference on Applications of kleene algebra, p.123-137, Frauenwörth, Germany, April 2008.

[15] T. Griffin, and G. Wilfong, *A safe path vector protocol*, in Proc. IEEE INFOCOM, vol. 2, Mar. 2000, pp. 490–499.

[16] T. G. Griffin, F. B. Shepherd, and G. Wilfong *Policy Disputes in Path-Vector Protocols*, In Proc. ICNP '99, 1999.

[17] J. Garcia and L. Aceves, *Loop free Routing Using Diffusing Computations*, IEEE ACM Transactions on Networking, February 1993.

[18] A. Abuzneid, B. J. Stark, *Improved BGP Convergence via MRAI Timer*, Novel Algorithms and Techniques in Telecommunications and Networking, Springer Netherlands, 2010.

[19] A. Beamer-Barr, A., Afek, and Y., Schwarz S., *Improved BGP Convergence via Ghost Flushing*, INFOCOM 2003, Volume: 2, On page(s): 927- 937 vol.2, July 2003.

[20] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, F. S. Wu and L. Zhang, *Improving BGP Convergence Through Assertions Approach*, In Proc. of the IEEE INFOCOM, June 2002.

[21] Dan Pei, Matt Azuma, Nam Nguyen, Jiwei Chen, Dan Massey, and Lixia Zhang, *BGP-RCN: Improving BGP Convergence through Root Cause Notification*, Computer Networks: The International Journal of Computer and Telecommunications Networking, June 2005.

[22] J. Chandrashekar, Z. Duan, Z.-L. Zhang, and J. Krasky, *Limiting path exploration in path vector protocols*, INFOCOM 2005, March 2005.

[23] K. Patel, C. Appanna, P. Mohapatra, J. Scudder, and J. Uttaro, *Root cause notification to solve BGP path hunting* http://tools.ietf.org/html/draft-keyupate-bgp-rcn-00, IETF, August 2010.

[24] D. Pei, X. Zhao, D. Massey, and L. Zhang, *A study of bgp path vector route looping behavior*, in International Conference on Distributed Computing Systems, Mar. 2004.

[25] C. Villamizar, R. Chandra and R. Govindan, *BGP Route Flap Damping, RFC 2439*, 1998.

[26] Z. M. Mao, R. Govindan, G. Varghese and R. H. Katz, *Route Flap Damping Exacerbates Internet Routing Convergence*, In Proc. of ACM SIGCOMM, volume 32, pages 221-233, October 2002.

[27] Ch. Panigl, J. Schmitz, P. Smith, and Cristina Vistoli, *RIPE Routing-WG Recommendations for Coordinated Route-flap Damping Parameters*, Document ID: ripe-229, October 2001.

[28] P. Smith, and Ch. Panigl, *RIPE Routing WG Recommendations on Route-flap Damping* http://www.ripe.net/docs/routeflap-damping.html, Document ID: ripe-378, Obsoletes: ripe-229, ripe-210, ripe-178, May 2006.

[29] SSF Research Network, SSFNet, http://www.ssfnet.org.

[30] B. J. Premore, *SSFNet BGP User's Guide*, Document version 2002-10-15.

[31] B. J. Premore, *Multi-AS topologies from BGP routing tables*, http://www.ssfnet.org/exchange/gallery/asgraph/index.html

[32] G. P. R. Álvarez, *Convergence Time reduction in the BGP4 Routing Protocol using the Ghost-Flushing Technique and Other Proposals*, Luleá Tekniska Universitet, 2004

[33] J. Nykvist, and L. Carr-Motyčková, *Simulating convergence properties of BGP*, In Proc. of the 11th International Conference on Computer Communications and Networks (ICCCN 2002), pages 124–129, October 2002.

[34] Y. Rekhter, T. Li, and S. Hares, *Border Gateway Protocol 4.* http://www.ietf.org/internet-drafts/draft-ietf-idrbgp4-20.txt, IETF, April 2003.

[35] T. Li, and G. Huston, *BGP Stability Improvements*, http://tools.ietf.org/html/draft-li-bgp-stability-01, IETF, June 2007.

[36] B. Decraene, P. Francois, C. Pelsser, Z. Ahmad, and A. J. E. Armengol, *Requirements for the graceful shutdown of BGP sessions*, IETF, October 2010.

[37] E. Zmijewski, *Longer is not always better* http://www.renesys.com/blog/2009/02/longer-is-not-better.shtml, February 2009.

[38] *YouTube Hijacking: A RIPE NCC RIS case study*, http://ripe.net/news/study-youtube-hijacking.html

[39] *Chinese ISP hijacked a 10 percent of Internet*, http://http://bgpmon.net/blog/?p=282, April 2010.

[40] A. Toonk, *Internet in Egypt offline*, http://bgpmon.net/blog/?p=450, January 2011.

[41] A. Toonk, *Egypt Back Online*, http://bgpmon.net/blog/?p=480, February 2011.

[42] O. Filip, *Protokol BGP pod útokem*, http://www.lupa.cz/clanky/protokol-bgp-pod-utokem/, September 2008.

[43] *Secure Inter-Domain Routing (sidr)*, http://datatracker.ietf.org/wg/sidr/charter/

[44] *32-bit ASN FAQs*, http://ripe.net/info/faq/rs/asn32.html, RIPE, March 2009.

[45] BGP Routing Table Analysis Reports, http://bgp.potaroo.net/, 2011.

[46] J. Moy, *OSPF Version 2, RFC 2328*, IETF, April 1998.

[47] G. Malkin, *RIP Version 2, RFC 2453*, IETF, November 1998.

[48] G. Malkin, *An Architecture for IP Address Allocation with CIDR, RFC 1518*, IETF, September 1993.

[49] P. Grygárek, *Směrování v počítačových sítích a v Internetu*, FEI VŠB-TU Ostrava.

[50] R. Pužmanová, *Moderní komunikační sítě od A do Z*, 2. vydání, Computer Press, 2006.

[51] R. Bellman *On a Routing Problem* in Quarterly of Applied Mathematics, 16(1), pp. 87–90, 1958.

[52] E. W. Dijkstra, *A note on two problems in connexion with graphs* In: Numerische Mathematik. 1, S. 269–271, 1959.

F. Ukázka DML souboru

Jedná se o vzorový příklad DML souboru pro topologii úplného grafu se čtyřmi uzly z [19], které mají nějakou cestu s uzlem, ve kterém dojde k násilnému ukončení BGP relace (simulace selhání destinace).

```
_schema [ _find .schemas.Net ]

Net [ # the all-encompassing Net
  frequency 100000000    # nanosecond simulation resolution
  bgpoptions [
    base_startup_wait 0.0

    split_horizon false
    jitter_mrai false
    jitter_keepalive false
    jitter_masoi false

    ghostflushing true
    ghostbuster 0
    cons_assert false
    root_cause_notification false

    show_snd_update true
    show_rcv_update true
    show_loc_rib_change true
    count_snd_update true
    max_aspath 5
    radix_trees true
    dump_loc_rib true

    ]

  Net [
    id 1
    AS_status boundary
    router [
      id 1
      graph [
        ProtocolSession [ name bgp  use SSF.OS.BGP4.BGPSession ]
        ProtocolSession [ name socket use SSF.OS.Socket.socketMaster ]
        ProtocolSession [ name tcp    use SSF.OS.TCP.tcpSessionMaster ]
        ProtocolSession [ name ip     use SSF.OS.IP ]
```

```
    ]
    interface [ id 0  virtual true ]
    interface [ id 2 ]
    interface [ id 3 ]
    interface [ id 4 ]
  ]
]
Net [
  id 2
  AS_status boundary
  router [
    id 1
    graph [
      ProtocolSession [ name bgp  use SSF.OS.BGP4.BGPSession ]
      ProtocolSession [ name socket use SSF.OS.Socket.socketMaster ]
      ProtocolSession [ name tcp    use SSF.OS.TCP.tcpSessionMaster ]
      ProtocolSession [ name ip     use SSF.OS.IP ]
    ]
    interface [ id 0  virtual true ]
    interface [ id 1 ]
    interface [ id 3 ]
    interface [ id 4 ]
  ]
]
Net [
  id 3
  AS_status boundary
  router [
    id 1
    graph [
      ProtocolSession [ name bgp  use SSF.OS.BGP4.BGPSession ]
      ProtocolSession [ name socket use SSF.OS.Socket.socketMaster ]
      ProtocolSession [ name tcp    use SSF.OS.TCP.tcpSessionMaster ]
      ProtocolSession [ name ip     use SSF.OS.IP ]
    ]
    interface [ id 0  virtual true ]
    interface [ id 1 ]
    interface [ id 2 ]
    interface [ id 4 ]
  ]
]
Net [
  id 4
```

```
    AS_status boundary
    router [
      id 1
      graph [
        ProtocolSession [ name bgp  use SSF.OS.BGP4.BGPSession ]
        ProtocolSession [ name socket use SSF.OS.Socket.socketMaster ]
        ProtocolSession [ name tcp    use SSF.OS.TCP.tcpSessionMaster ]
        ProtocolSession [ name ip     use SSF.OS.IP ]
      ]
      interface [ id 0  virtual true ]
      interface [ id 1 ]
      interface [ id 2 ]
      interface [ id 3 ]
    ]
  ]
Net [
    id 5
    AS_status boundary
    router [
      id 1
      graph [
        ProtocolSession [
          name bgpkiller  use SSF.OS.BGP4.Widgets.BGPKiller
          kill 100
        ]
        ProtocolSession [ name bgp  use SSF.OS.BGP4.BGPSession ]
        ProtocolSession [ name socket use SSF.OS.Socket.socketMaster ]
        ProtocolSession [ name tcp    use SSF.OS.TCP.tcpSessionMaster ]
        ProtocolSession [ name ip     use SSF.OS.IP ]
      ]
      interface [ id 0  virtual true ]
      interface [ id 4 ]
    ]
  ]
  link [ attach 1:1(2) attach 2:1(1) delay 1 ]
  link [ attach 1:1(3) attach 3:1(1) delay 1 ]
  link [ attach 1:1(4) attach 4:1(1) delay 1 ]
  link [ attach 2:1(3) attach 3:1(2) delay 1 ]
  link [ attach 2:1(4) attach 4:1(2) delay 1 ]
  link [ attach 3:1(4) attach 4:1(3) delay 1 ]
  link [ attach 4:1(5) attach 5:1(4) delay 1 ]
]
```

G. Obsah přiloženého CD

Přiložené CD obsahuje následující adresářovou strukturu:

`bin/`
: Skripty z části H. použité pro spuštění simulace SSFNet a získávání výsledků z výstupních souborů simulace.

`doc/`
: Dokumentace práce ve formátu PDF, vytvořená dle závazného stylu KI PřF pro diplomové práce, včetně všech příloh, a všechny soubory nutné pro bezproblémové vygenerování PDF souboru dokumentace (v ZIP archivu), tj. zdrojový text dokumentace, vložené obrázky, apod.

`lib/`
: Potřebné knihovny JAR pro běh simulace SSFNet, které je potřeba vložit do systémové proměnné `CLASSPATH`. Jejich pořadí je popsáno v readme.txt.

`src/`
: Kompletní zdrojové kódy simulátoru `SSFNet` z [29] a zdrojové kódy implementující metody pro snižovaní konvergence BGP protokolu, z nichž je vytvořena knihovna JAR poskytující tyto metody pro použití v simulátoru.

`readme.txt`
: Instrukce pro spuštění simulací v simulátoru `SSFNet` včetně požadavků pro jeho provoz.

Navíc CD obsahuje:

`data/`
: Veškerá výstupní data ze simulací uvedených v tomto textu. Jsou to textové soubory, které obsahují naměřené časy konvergence, počet zpráv a délky nejdelší ASpath, které byly dál zpracovány pro použití v této práci.

`install/`
: Instalátor kolekce linuxových nástrojů Cygwin pro Microsoft Windows pro spuštění skriptů uvedených v H.. Běhové prostředí Java(TM) Runtime Environment pro spuštění simulace lze nalézt na http://www.java.com.

`literature/`
: Některé položky literatury odkazované v textu této práce.

U veškerých odjinud převzatých materiálů obsažených na CD jejich zahrnutí dovolují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro materiály, u kterých toto není splněno, je uveden jejich zdroj (webová adresa) v textu dokumentace práce nebo v souboru `readme.txt`.

H. Popis skriptů

genMat.py

Generuje matice sousednosti grafu sítě do souborů *.mat.
Parametry:

-c ČÍSLO - počet uzlů grafu sítě

-h ČÍSLO - velkost kroku v počtu přidávaných hran (výchozí hodnota je 1)

genDML.py

Generuje DML soubory pro SSFnet podle souboru *.mat.
Parametry:

-c ČÍSLO - počet uzlů grafu sítě

-dt ČÍSLO - nastaví čas selhání uzlu od začátku simulace

-gf - aktivuje metodu Ghost-Flushing

-gb ČíSLO - aktivuje metodu Ghost-Buster s uvedeným δ zpožděním. Metoda Ghost-Flushing je aktivána automaticky, není nutné zadávat předchozí parametr

-ca - aktivuje metodu konzistentních pravidel

-rc - aktivuje metodu určující původ změny

-sh - aktivuje split horizon

-ji - aktivuje jitter

-o - definice výstupního souboru

-dt - nastavuje čas výpadku uzlu, který představuje nedostupnou destinaci

-tmp - uloží výsledný dml soubor do /tmp

SOUBOR(Y) - použít dané vstupní soubory *.mat

extract.py

Nalezne z výstupního souboru simulace zvolený údaj jako je čas konvergence, počet zpráv nebo délka nejdelší ASpath k nedostupné destinaci.
Parametry:

-time - zobrazí čas konvergence

-msge - zobrazí počet zpráv

-length - zobrazí délku nejdelší *ASpath*

SOUBOR - výstupní logovacím soubor simulace

genAvgFile.py

Sestaví tabulku (v textovém souboru, ve kterém jsou záznamy odděleny mezerou) s průměrnými časy konvergence, počty zpráv nebo délek nejdelší cesty ASpath k nedostupné destinaci ze získaných naměřených dat, který je možné použít k importu do některého spreadsheetu jako je Microsoft Excel nebo OpenOffice Calc.
Parametry:

-o SOUBOR - název výstupního souboru s tabulkou

SOUBOR(Y) - výstupní logovacím soubor(y) simulace

genMat.sh

Generuje matice sousednosti grafu sítě do souborů *.mat pomocí skriptu genMat.py pro zadané parametry.
Parametry:

ČÍSLO - počet uzlů grafu sítě (povinný parametr)

ČÍSLO - velkost kroku v počtu přidávaných hran (povinný parametr)

run.sh

Z vygenerovaných souborů *.mat sestavuje odpovídající DML soubory skriptem genDML.py, spouští simulace s těmito DLM soubory pro zadané parametry a hledá ve výstupních souborech simulací požadované hodnoty skriptem extract.py. Takto naměřené hodnoty jsou ukládány do souborů ve složce `outputs`.
Parametry:

ČÍSLO - počet uzlů grafu sítě (povinný parametr)

ČÍSLO - počet hran, od kterého se začne první simulace (povinný parametr)

-gf - aktivuje metodu Ghost-Flushing

-gb ČíSLO - aktivuje metodu Ghost-Buster s uvedeným δ zpožděním. Metoda Ghost-Flushing je aktivána automaticky, není nutné zadávat předchozí parametr.

-ca - aktivuje metodu konzistentních pravidel

-rc - aktivuje metodu určující původ změny

-sh - aktivuje split horizon

-ji - aktivuje jitter

genAvgFile.sh

Sestaví tabulku ze zvolených naměřených dat pomocí skriptu genAvgFile.py.
Parametry:

times/msges/lengths - typ dat - časová konvergence/počet zpráv/délka nejdelší ASPath (povinný parametr)

ČÍSLO - počet uzlů grafu sítě (povinný parametr)

-sh - použije data s aktivovaným split horizon

-ji - použije data s aktivovaným jitter

genGraph.sh

Sestaví odpovídající graf z tabulky s průměrnými časy konvergence, počty zpráv nebo délek nejdelších cest *Aspath* k nedostupnému uzlu pomocí Gnuplot.
Parametry:

times/msges/lengths - typ dat - časová konvergence/počet zpráv/délka nejdelší *Aspath* (povinný parametr)

ČÍSLO - počet uzlů grafu sítě (povinný parametr)

-sh - použije data s aktivovaným split horizon

-ji - použije data s aktivovaným jitter