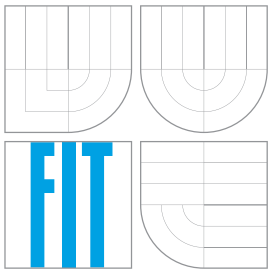
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MODELOVÁNÍ PROTOKOLŮ HSRP A GLBP PRO REDUNDANCI BRÁNY

MODELLING HSRP AND GLBP GATEWAY REDUNDANCY PROTOCOLS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAN HOLUŠA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. VLADIMÍR VESELÝ

BRNO 2016

Abstrakt

Tato diplomová práce se zabývá rozborem protokolů zajišťujících redundanci síťové brány. Jsou zde popsány protokoly *Hot standby router protocol*, *Virtual router redundancy protocol* a *Gateway load balancing protocol*. Zároveň jsou u jednotlivých protokolů uvedeny možnosti konfigurace na zařízeních Cisco s uvedením podporované verze Cisco IOS. Dále je součástí práce návrh a implementace dvou těchto protokolů *Hot standby router protocol* a *Gateway load balancing protocol* do simulačního prostředí OMNeT++ do knihovny *Automated network simulation and analysis*. Také je zde uvedeno testování správnosti těchto implementací v porovnání s reálnými zařízeními Cisco.

Abstract

This thesis deals with theoretical analysis of First Hop Redundancy Protocols. It describes Hot Standby Router Protocol, Virtual Router Redundancy Protocol and Gateway Load Balancing Protocol. It also shows examples of configuration of each protocol on Cisco devices with supported version of the Cisco IOS. Furthermore, this thesis includes design of two of these protocols, Hot Standby Router Protocol and Gateway Load Balancing Protocol, and their implementation in discrete event simulator OMNeT++ and Automated Network Simulation and Analysis library. Finally, the thesis presents results of testing of the implementations in comparison with actual Cisco devices.

Klíčová slova

FHRP, Protokoly zajišťující redundanci síťové brány, HSRP, Hot Standby Redundancy Protocol, VRRP, Virtual Router Redundancy Protocol, GLBP, Gateway Load Balancing Protocol, CARP, Common address redundancy protocol, Cisco, OMNeT++, INET, ANSAINET.

Keywords

FHRP, First Hop Redundancy Protocol, HSRP, Hot Standby Redundancy Protocol, VRRP, Virtual Router Redundancy Protocol, GLBP, Gateway Load Balancing Protocol, CARP, Common address redundancy protocol, Cisco, OMNeT++, INET, ANSAINET.

Citace

HOLUŠA, Jan. *Modelování protokolů HSRP a GLBP pro redundanci brány.* Brno, 2016. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Veselý Vladimír.

Modelování protokolů HSRP a GLBP pro redundanci brány

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana inženýra Vladimíra Veselého. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jan Holuša
23. května 2016

Poděkování

Na tomto místě chci poděkovat inženýru Vladimíru Veselému za jeho čas, vstřícnost, odborné rady a také za jeho způsob výuky, kterým mě kontinuálně motivoval k píli během magisterského studia. Velice děkuji mým milým rodičům, kteří mi dali možnost studovat a kteří mě podporovali při studiu a při vytváření této práce. V neposlední řadě děkuji přítelkyni za její obrovskou trpělivost a za gramatickou kontrolu textu. Zvláštní dík pak patří Tomovi Rajcovi jehož zdravá soutěživost a smysl pro dokonalost mi byly inspirací.

Ve znamení díků zde uvádím recept španělské kuchyně s názvem *Pechuga de pollo a la naranja*. Jsou zapotřebí 4 kuřecí prsa, která se osolí, opepří a v hluboké pánvi opečou ve čtyřech lžicích olivového oleje dozlatova. Po opečení se vyjmou a ve zbylém oleji osmahneme 2 mrkve nakrájené na kostičky a 1 nasekanou cibuli. Přidáme cukr a necháme jemně zkaramelizovat. Přilijeme šťávu ze 4 pomerančů, 1 limetky a 125 ml bílého vína. Omáčku necháme vařit a odpařovat do mírného zhoustnutí. Poté rozmixujeme, dosolíme a dopepříme. Vložíme kuřecí prsa a necháme táhnout. Po deseti minutách podáváme nakrojená kuřecí prsa s vloženým plátkem pomeranče. Vhodná příloha je rýže.

Obsah

Kapitola 1

Úvod

V dnešní době stále se zrychlujících komponent síťové infrastruktury je zároveň důležité zabezpečit maximální spolehlivost a s tím související dostupnost sítě. K tomu napomáhá redundance jednotlivých síťových komponent, kdy v případě výpadku jednoho zařízení začne jeho práci vykonávat jiné, záložní, zařízení. Během zavádění záložních zařízení je potřeba myslet i na protokoly, které patřičně zajistí plynulý přechod síťového provozu z jednoho zařízení na druhé. Například protokoly pro redundanci síťové brány, kde je snaha o zachování maximální dostupnosti připojení k vnější síti. A právě vysvětlením, popisem a nastavením těchto protokolů pro redundanci síťové brány (FHRP, *First hop redundancy protocol*) se zabývám v této diplomové práci. V druhé polovině práce se pak zabývám implementací protokolů *Hot standby router protocol* (HSRP) a *Gateway load balancing protocol* (GLBP) a jejich verifikací vůči Cisco implementaci.

Tato diplomová práce navazuje na semestrální projekt, ve kterém jsem zpracoval teoretické podklady protokolů FHRP.

1.1 Struktura práce

V kapitolách 2, 3 a 4 popisuji funkcionalitu FHRP protokolů. V každé kapitole uvádím formát paketů, kterým protokoly komunikují. Dále způsob volby virtuální síťové brány mezi směrovači a s tím související stavový automat jednotlivých protokolů. V závěru každé z těchto kapitol vypisuji možnosti konfigurace na Cisco zařízeních s popisem jednotlivých parametrů a s příklady složitějších konfigurací. U každého příkazu uvádím verzi Cisco IOS, od které je daný příkaz podporován.

V kapitole 5 popisuji simulační prostředí OMNeT++, ve kterém je zpracovaná praktická část této diplomové práce. Zároveň zde popisuji projekt ANSA (*Automated Network Simulation and Analysis*), která se zabývá simulací protokolů nad TCP/IP v OMNeT++ a jejíž rozšířením se věnuji ve své práci.

V následujících dvou kapitolách 6 a 7 popisuji návrh a implementaci HSRP a GLBP modulů do jako součástí projektu ANSA. Také zde uvádím formát zpracování zpráv a způsob konfigurace modulů. Z důvodu nedostatečně detailní specifikace proprietárního protokolu GLBP vykazuje simulace jemně odlišné chování oproti Cisco implementaci. Tyto odlišnosti jsou popsány v kapitole 7.5.

Kapitola 8 je věnována testování mé implementace. Na několika modelových případech ukazuji, do jaké míry se HSRP a GLBP shoduje s chováním těchto protokolů na Cisco směrovačích. Všechny testy jsou prováděny na jednotné topologii sítě, která je popsána v úvodu

kapitoly. Scénář jednotlivých testů spolu se zhodnocením výsledků uvádím v jednotlivých podkapitolách.

V závěrečné kapitole 9 shrnuji dokončenou práci na protokolech FHRP, upozorňuji na nedostatky aktuálního zpracování a navrhuji možná řešení.

Kapitola 2

Hot standby router protocol

HSRP (*Hot standby router protocol*) [12, 14, 1] je Cisco proprietární protokol běžící na multi-access, multicast a broadcast sítích. Protokol umožňuje zlepšit redundanci směrovacích zařízení v síti tak, že se uživatelům jeví skupina směrovačů jako jedno zařízení (*virtual router*). V momentě výpadku jednoho zařízení (*active router*) protokol výpadek detekuje a aktivuje záložní zařízení (*standby router*), aniž by uživatelé zaznamenali výpadek spojení.

Protokol je dostupný ve dvou verzích. V této kapitole popisuji verzi 1 a v poslední části této kapitoly uvádím odlišnosti vyskytující se v HSRPv2 oproti verzi 1.

2.1 Vysvětlení terminologie

Pro podrobné vysvětlení principu fungování HSRP uvádím vysvětlení následujících termínů:

- *Virtual router* je skupina směrovačů tvářících se jako jeden směrovač. Nazývá se také HSRP skupina (*HSRP group*) nebo *standby group*. V jedné HSRP skupině může být libovolný počet směrovačů.
- *Active router* je směrovač, který v rámci HSRP skupiny jako jediný směruje pakety od uživatelů.
- *Standby router* je záložní směrovač.
- *Hello time* je časový interval mezi následnými *Hello* zprávami z daného směrovače.
- *Hold time* je doba, po kterou má směrovač čekat na odpověď na *Hello* zprávu, než prohlásí protějšek za nefunkční.

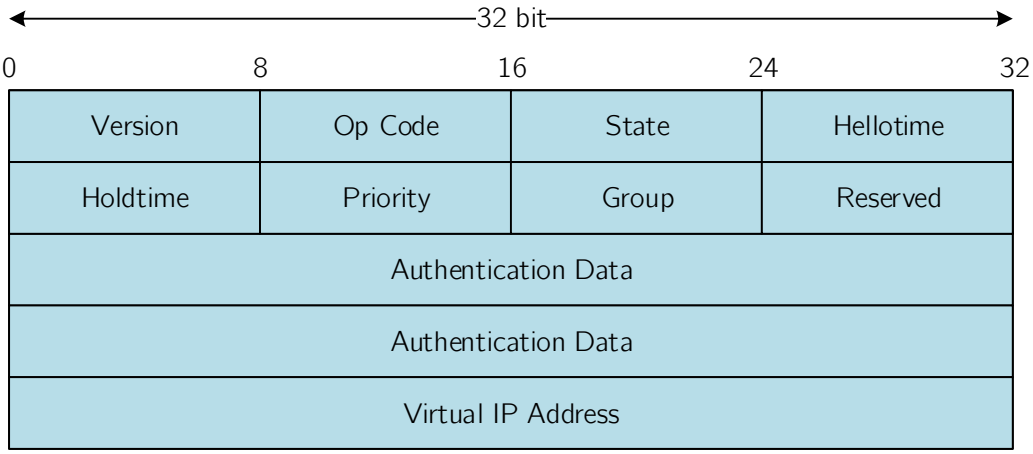
2.2 Formát paketu

Na obrázku 2.1 uvádím schéma UDP paketu, který používá HSRP pro komunikaci mezi jednotlivými směrovači.

Zde je vysvětlení jednotlivých polí paketu:

- `Version` udává verzi HSRP zprávy.
- `Op Code` popisuje, jaký typ zprávy paket obsahuje. Možnosti jsou následující:
  - 0: *Hello* zpráva

- 1: *Coup* zpráva
- 2: *Resign* zpráva



Obrázek 2.1: Struktura HSRP paketu. Převzato z [12].

Hello zprávy dávají najevo, že směrovač je funkční a je ochotný stát se *active*, nebo *standby router*. *Coup* zprávu zasílá směrovač, který si přeje stát se *active*. *Resign* zprávu zasílá směrovač, který už nechce být *active*.

- `State` udává, ve kterém stavu se daný směrovač nachází. Jednotlivé stavy jsou popsány níže (viz obrázek 2.3). Možné hodnoty jsou následující:
  - 0: *Init*
  - 1: *Learn*
  - 2: *Listen*
  - 4: *Speak*
  - 8: *Standby*
  - 16: *Active*
- `Hellotime`[1] udává časovou prodlevu v sekundách mezi odesláním jednotlivých *Hello* zpráv. Pokud na směrovači není tento parametr nastavený, může se hodnotu naučit z autentizovaných zpráv od *active routeru*. Výchozí doporučená hodnota jsou 3 sekundy.
- `Holdtime`[1] udává čas v sekundách, po který je *Hello* zpráva pokládána za validní. Hodnota *holdtime* musí být větší než hodnota *hellotime*. Doporučená hodnota je nejméně dvojnásobek *hellotime*. Pakliže parametr není na směrovači nastavený, může se ho naučit z autentizované *Hello* zprávy. Výchozí hodnota je 10 sekund.

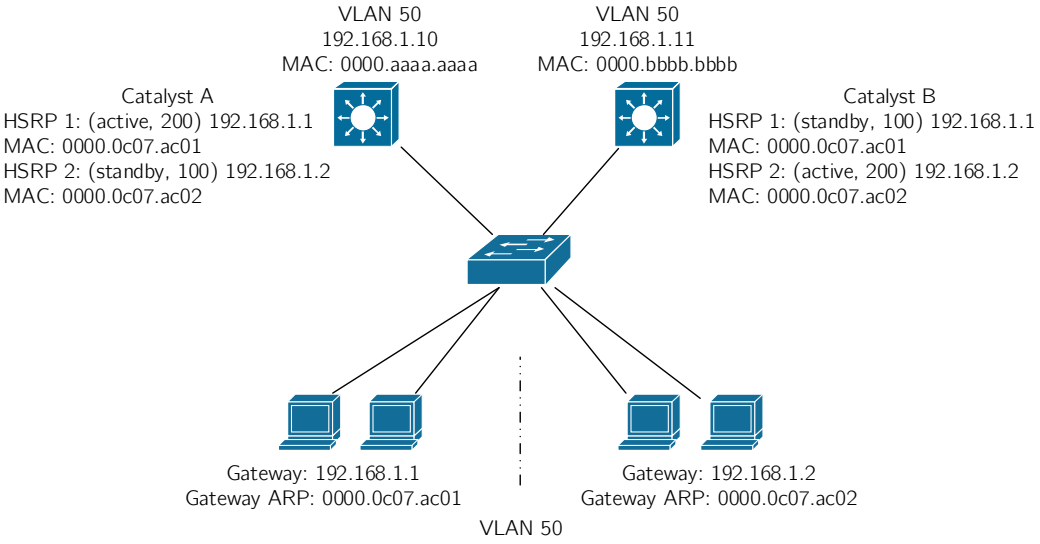
[1]Hodnoty *hellotime* a *holdtime* musejí být u *active routeru* obě zadány ručně, nebo obě naučeny od předchozího *active routeru*. Nesmí nastat kombinace naučení se a ručního zadání. Zabrání se tím situaci, kdy *holdtime* bude nedostatečný.

- `Priority` je hodnota, která se využívá při volbě *active* a *standby routeru*. Vyšší priorita vítězí. V případě shody vyhrává směrovač s vyšší hodnotou IP adresy.
- `Group` znázorňuje identifikátor skupiny. U *Token Ring* sítí jsou povolené hodnoty mezi 0 a 2. U ostatních sítí mezi 0 a 255.
- `Authentication data` udává 8 znakové nezašifrované heslo. Výchozí nastavení má hodnotu 0x63 0x69 0x73 0x63 0x6F 0x00 0x00 0x00 (neboli „cisco“). Neslouží k zabezpečení protokolu, ale spíš k omezení chyb při konfiguraci.
- `Virtual IP address` udává jakou virtuální IP adresu bude daná HSRP skupina používat. Pakliže není zadána, může se ji směrovač naučit z autentizovaných *Hello* zpráv od *active routeru*. Virtuální adresa musí být zadána z adresního prostoru HSRP rozhraní. Zároveň nesmí být stejná, jako jakákoli IP adresa z HSRP skupiny.

2.3 Další možnosti protokolu

Protokol HSRP umožňuje, aby směrovač byl součástí více HSRP skupin (takzvané *Multigroup HSRP*, MHSRP). Tuto funkci lze povolit pouze na směrovačích, které obsahují hardware umožňující asociovat ethernetové rozhraní s více MAC adresami.

S MHSRP souvisí možnost zavést *load balancing*, kdy se nastaví 2 směrovače ve 2 HSRP skupinách tak, že jeden je ve stavu *active* pro jednu skupinu a druhý je ve stavu *active* pro druhou skupinu, ale zároveň jsou oba v obou skupinách. Uvádím příklad (viz obrázek 2.2), kde počítače na levé straně patří do HSRP skupiny 1 a počítače na pravé straně do HSRP skupiny 2. L3 switche *Catalyst A* a *Catalyst B* patří do obou skupin. *Catalyst A* je *active router* pro HSRP skupinu 1 a *Catalyst B* je *active router* pro HSRP skupinu 2. Probíhá *load balancing*. V případě výpadku jednoho směrovače ho zastoupí ten druhý a *load balancing* přestane probíhat až do opětovného zprovoznění odstaveného směrovače.



Obrázek 2.2: Příklad *load balancingu* při použití dvou HSRP skupin. Převzato z [11].

Protokol HSRP umožňuje nastavení těchto voleb:

- *Preemption* neboli preempce, která umožňuje okamžitou výměnu *active routeru* pomocí *Coup* zpráv v případě, že se v topologii objeví směrovač s vyšší prioritou. Ve výchozím nastavení je preempce vypnutá. Detaily ohledně procesu volby jsou popsány v kapitole 2.4.
- *Preempt delay* umožňuje zpožděnou preempci za účelem umožnění směrovači dokončit konvergenci sítě předtím, než se stane *active routerem*.
- *Interface tracking* umožňuje směrovači sledovat stav rozhraní a v případě výpadku sledovaného rozhraní upravit prioritu daného směrovače.

2.4 Proces volby

Priorita každého zařízení v HSRP skupině je zjištěna na základě zadaného parametru priority a poté, v případě shody, podle IP adresy zařízení. Směrovač s nejvyšší prioritou v HSRP skupině se stává *active router*. Směrovač s druhou nejvyšší prioritou se stává *backup router*.

V případě připojení směrovače s vyšší prioritou do HSRP skupiny nastane při nastavené preempci následující situace. Směrovač s vyšší prioritou pošle *active routeru Coup* zprávu. Když *active router Coup* zprávu přijme, tak se přepne do stavu *Speak* a odešle *Resign* zprávu. Tím umožní nově připojenému směrovači stát se *active routerem*.

2.5 Popis protokolu

Všechny směrovače, nad kterými chceme provozovat HSRP, musí obsahovat stejnou sadu směrovacích pravidel. Směrovače v HSRP skupině si pravidelně posílají zprávy o svém stavu nad protokolem UDP na portu číslo 1985. Po zvolení *active* a *standby routeru* si kvůli minimalizaci síťového provozu zasílají HSRP zprávy pouze *active* a *standby routery*. Využívají multicastovou adresu 224.0.0.2 s TTL = 1. Pro tyto informační zprávy používají jako zdrojovou IP adresu svou IP adresu rozhraní. Formát UDP datagramu sloužícího pro komunikaci v rámci HSRP skupiny je popsán v kapitole 2.2.

Active router přijímá a přeposílá provoz určený pro MAC adresu HSRP skupiny. V momentě, kdy přestane být *active router*, musí přestat. Do pole zdrojové MAC adresy *Hello* zprávy ukládá *active router* MAC adresu HSRP skupiny. MAC adresa HSRP skupiny závisí na čísle skupiny (`0000.0C07.ACxx`, kde `xx` je hexadecimálně zapsané číslo skupiny).

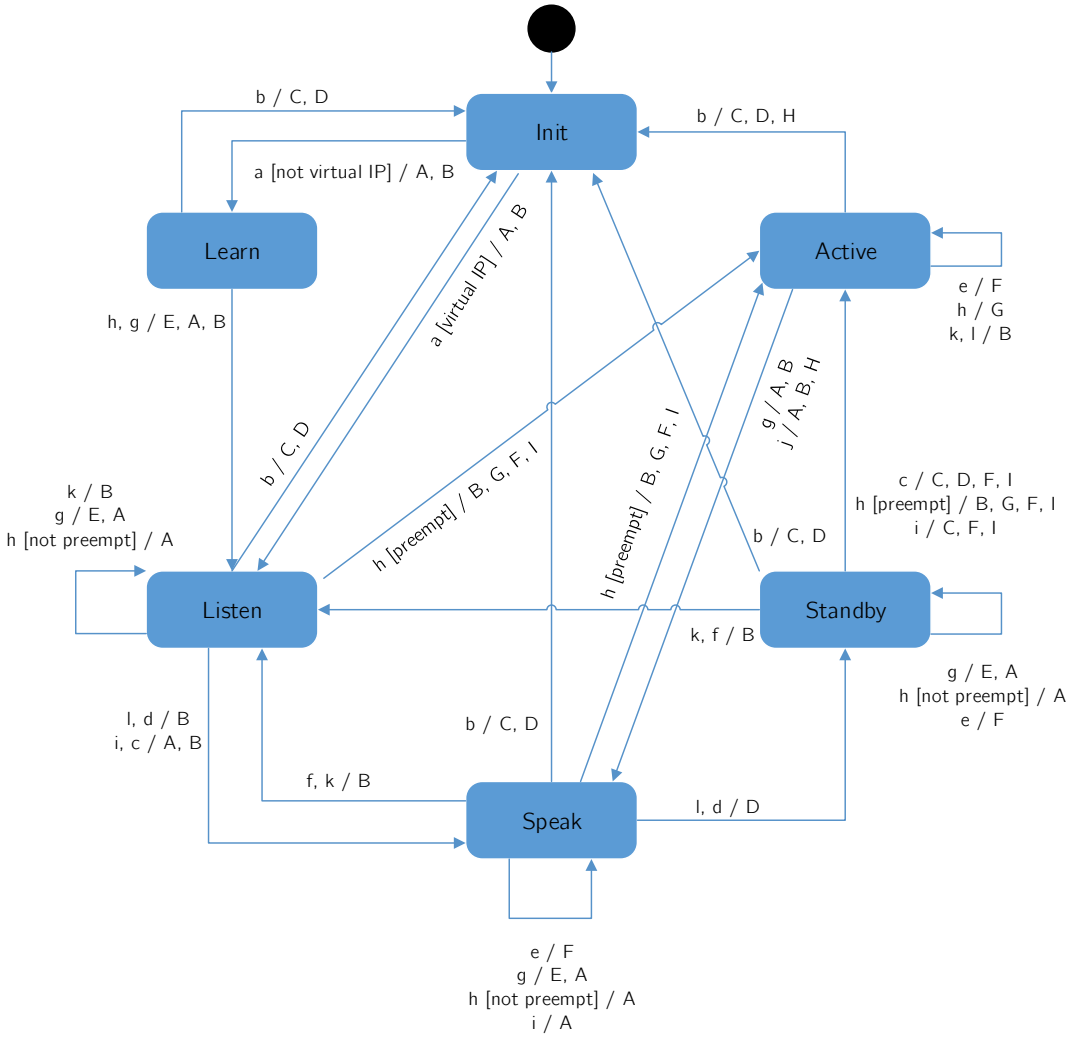
Následující parametry UDP datagramu musí být známy každému směrovači v HSRP skupině:

- `Číslo HSRP skupiny`
- `Virtuální MAC adresa`
- `Priorita`
- `Autentizační data`
- `Hellotime`
- `Holdtime`

Parametr `Virtuální IP adresa` musí být znám alespoň jednomu směrovači v každé HSRP skupině.

2.5.1 Stavový automat

Každý směrovač se nachází v jednom ze stavů následujícího stavového automatu. Všechny jsou na začátku (po zprovoznění HSRP protokolu) ve stavu *init*.



Obrázek 2.3: Stavový automat protokolu HSRP. Převzato z [3]. V příloze je přiložena přechodová tabulka (viz C).

Jednotlivé stavy stavového automatu znázorňují následující:

- **Init**
  Je startovní stav a znamená, že HSRP ještě neběží.
- **Learn**
  Znázorňuje stav, ve kterém směrovač zatím nezná virtuální IP adresu a zatím nepřijal autentizační *Hello* zprávu od *active routeru* ze které by se virtuální IP adresu naučil.
- **Listen**
  Stav, kdy směrovač zná svou virtuální IP adresu, ale není *active router*, ani *standby*

router. Přijímá *Hello* zprávy od *active* a *standby routeru*.

- **Speak**
  Stav, ve kterém směrovač zasílá pravidelné *Hello* zprávy a podílí se na volbě *active* a *standby routeru*. Směrovač se nemůže dostat do tohoto stavu, pokud nezná svou virtuální IP adresu.

- **Standby**
  Stav, kdy je směrovač záložní pro *active router*. Nanejvýš jeden směrovač může být ve stavu *standby*.

- **Active**
  Jedná se o jediný stav, kdy směrovač přeposílá pakety, které jsou zaslané na virtuální IP adresu HSRP skupiny. Nanejvýš jeden směrovač ve skupině může být ve stavu *active*.

Každý směrovač pracuje se třemi časovači, které využívá při přeposílání stavových zpráv:

- *Active timer* monitoruje *active router*. Je spuštěn, když směrovač přijme autentizovanou *Hello* zprávu od *active routeru*. *Active timer* vyprší za dobu určenou parametrem *Holdtime*.

- *Standby timer* monitoruje *standby router*. Je spuštěn, když směrovač přijme autentizovanou *Hello* zprávu od *standby routeru*. *Standby timer* vyprší za dobu určenou parametrem *Holdtime*.

- *Hello timer* určuje, kdy má směrovač generovat *Hello* zprávu. Jakmile u směrovačů ve stavu *speak*, *standby* nebo *active* vyprší *hello timer* (tj. jednou za *hellotime*), tak vygenerují *Hello* zprávu. *Hello timery* na jednotlivých zařízeních se musí opožďovat, jinak by docházelo pravidelně ke generování *Hello* zpráv všemi zařízeními v HSRP skupině a v závislosti na velikosti HSRP skupiny by to mohlo vést k nepřípustnému zatížení sítě.

Popis událostí, které se objevují v konečném automatu, je následující:

- `a`: HSRP je nastaveno na spuštěném rozhraní.

- `b`: HSRP není nastaveno na spuštěném rozhraní, nebo je rozhraní vypnuto.

- `c`: Vypršel *active timer*. *Active timer* nastaven na *holdtime*, když obdržel poslední *Hello* zprávu od směrovače *active*.

- `d`: Vypršel *standby timer*. *Standby timer* nastaven na *holdtime*, když obdržel poslední *Hello* zprávu od *standby routeru*.

- `e`: Vypršel *hello timer*. Časovač pro pravidelné odesílání *Hello* zpráv vypršel.

- `f`: Přijetí *Hello* zprávy vyšší priority od směrovače ve stavu *speak*.

- `g`: Přijetí *Hello* zprávy vyšší priority od *active routeru*.

- `h`: Přijetí *Hello* zprávy nižší priority od *active routeru*.

- `i`: Přijetí *Resign* zprávy od *active routeru*.

- `j`: Přijetí *Coup* zprávy od směrovače s vyšší prioritou.
- `k`: Přijetí *Hello* zprávy vyšší priority od *standby routeru*.
- `l`: Přijetí *Hello* zprávy nižší priority od *standby routeru*.

Popis vykonaných akcí:

- `A`: Spustí *active timer*. Jestliže tato akce nastane jako výsledek přijetí autentizované *Hello* zprávy od *active routeru*, tak nastaví *active timer* na hodnotu *holdtime* v *Hello* zprávě. V opačném případě je *active timer* nastaven na aktuální lokální hodnotu *holdtime*.
- `B`: Spustí *standby timer*. Jestliže tato akce nastane jako výsledek přijetí autentizované *Hello* zprávy od *standby routeru*, tak nastaví *standby timer* na hodnotu *holdtime* v *Hello* zprávě. V opačném případě je *standby timer* nastaven na aktuální lokální hodnotu *holdtime*.
- `C`: Zastaví *active timer*.
- `D`: Zastaví *standby timer*.
- `E`: Naučí se parametry. Tato akce se vykoná, jestliže je přijata autentizovaná zpráva od *active routeru*. Jestliže nebyla ručně nastavena virtuální IP adresa pro tuto HSRP skupinu, tak se ji může naučit ze zprávy. Také *hellotime* a *holdtime* se může naučit ze zprávy.
- `F`: Zašle *Hello* zprávu se svým aktuálním stavem, *hellotime* a *holdtime*.
- `G`: Zašle *Coup* zprávu, aby informoval *active router*, že je v topologii směrovač s vyšší prioritou.
- `H`: Zašle *Resign* zprávu, aby umožnil jinému směrovači být *active router*.
- `I`: Zašle ARP paket *Reply*, aby informoval ostatní o virtuální IP a MAC adrese.

2.6 Podpora na Cisco zařízeních

Cisco zařízení podporují protokol HSRP. Uvádím zde možné konfigurační příkazy, jejich popis a u složitějších příkazů přidávám příklad. U každého příkazu je uvedena verze Cisco IOS, od které je podporován. Jednotlivé příkazy čerpám z Cisco IOS referenční příručky pro FHRP příkazy [7]. Ladící příkazy potom z Cisco příručky pro `debug` příkazy [4].

Cisco zařízení začala podporovat protokol HSRP od verze Cisco IOS 10.0. Ve verzi 12.4(4)T je přidána podpora pro IPv6.

2.6.1 Základní příkazy

Protokol se spouští na požadovaném rozhraní následujícím příkazem:

```
Router(config-if)# standby [group-number] ip [ip-address
[secondary]]
```

Kde jednotlivé parametry znamenají následující:

- *group-number* znázorňuje číslo skupiny. Tento parametr je možné vynechat, v tom případě se použije výchozí hodnota 0.
- *ip-address* je volitelný, ale musí být zadán alespoň na jednom zařízení v HSRP skupině. Znázorňuje virtuální IP adresu HSRP rozhraní.
- `secondary` je volitelný parametr a udává, že zadaná IP adresa je vázaná na sekundární fyzické rozhraní.

Chceme-li protokol vypnout použijeme příkaz `no standby` *group-number*. Uvádím příklad spuštění protokolu na rozhraní Ethernet1 s adresou sítě 192.168.10.0/24, kde zvolená adresa rozhraní je 192.168.10.1/24 a zvolená adresa HSRP brány pro HSRP skupinu 1 je 192.168.10.254/24. Dále je přidána na rozhraní nová, sekundární síť s adresou 192.168.50.0/24, kde adresa rozhraní je 192.168.50.1/24 a adresa HSRP brány pro HSRP skupinu 2 je 192.168.50.254/24.

Příklad:

```
Router(config)# interface Ethernet1
Router(config-if)# ip address 192.168.10.1 255.255.255.0
Router(config-if)# standby 1 ip 192.168.10.254
Router(config-if)# ip address 192.168.50.1 255.255.255.0
secondary
Router(config-if)# standby 2 ip 192.168.50.254 secondary
```

Příkaz je dostupný v Cisco IOS od verze 10.0. Ve verzi 10.3 přibyl argument *group-number*. Ve verzi 11.1 pak přibylo klíčové slovo `secondary`.

Dále je možné zvolit verzi protokolu příkazem:

```
Router(config-if)# standby version {1 | 2}
```

Výchozí nastavení verze je 1. Příkaz je dostupný v Cisco IOS od verze 12.3(4)T. Od verze 12.4(4)T je přidána podpora pro IPv6.

Pro změnu priority na rozhraní, na kterém je spuštěné HSRP se používá následující příkaz:

```
Router(config-if)# standby [group-number] priority priority
```

Kde parametr *priority* může být v rozsahu 0 až 255 a jeho výchozí hodnota je 100. Vyšší číslo znamená, že má zařízení větší prioritu stát se *active router*. Příkaz je dostupný v Cisco IOS od verze 11.3. Od verze 12.4(4)T je přidána podpora pro IPv6.

Nastavení preempce a zpožděné preempce probíhá následujícím příkazem:

```
Router(config-if)# standby [group-number] preempt
[delay{[minimum seconds][reload seconds][sync seconds]}]
```

Kde jednotlivé parametry znamenají následující:

- `delay minimum` *seconds* je volitelný parametr udávající minimální čas, o který se zpozdí inicializace HSRP skupiny po zprovoznění HSRP rozhraní. Rozsah možných hodnot je 0 až 3600. Výchozí hodnota je 0 (bez zpoždění).

- `delay reload` *`seconds`* je volitelný parametr udávající, o kolik sekund se opozdí inicializace HSRP skupiny po restartu zařízení.
- `delay sync` *`seconds`* je volitelný parametr a udává maximální počet sekund, o který lokální směrovač odloží převzetí role *active* nebo *standby routeru*. Jakmile tento časový interval vyprší, tak je proveden proces volby a převzetí rolí.

Po zadání následujícího příkazu na HSRP rozhraní bude zařízení čekat 300 milisekund předtím, než se bude pokoušet být *active routerem* v HSRP skupině číslo 1.

Příklad:
```
Router(config-if)# standby preempt delay minimum 300
```

Příkaz pro nastavení preempce je součástí Cisco IOS od verze 11.3. Od verze 12.0(2)T jsou přidána klíčová slova `minimum` a `sync`. Od verze 12.2 je přidáno klíčové slovo `reload`. Podpora pro IPv6 byla přidána ve verzi 12.4(4)T.

Pro nastavení časovačů *hellotime* a *holdtime* slouží následující příkaz:

```
Router(config-if)# standby [group-number] timers [msec]
hellotime [msec] holdtime
```

Kde jednotlivé parametry znamenají:

- `msec` je volitelný příznak toho, že se bude čas zadávat v milisekundách.
- *`hellotime`* je interval zasílání *Hello* zpráv zadaný v sekundách. Je možné zadávat hodnoty od 1 do 254. Výchozí hodnota je 3 sekundy. V případě příznaku `msec` je možné zadat od 15 do 999 milisekund.
- *`holdtime`* je čas v sekundách předtím, než je *active* nebo *standby router* považován za nedostupný. Může nabývat hodnot od x do 255 sekund. Hodnota x se vypočítá následovně: $x = \lceil hellotime + 50ms \rceil$, kde závorky $\lceil$ a $\rceil$ znázorňují zaokrouhlení na nejbližší sekundu nahoru. Výchozí hodnota *holdtime* je 10 sekund. V případě příznaku `msec` může nabývat od y do 3000 milisekund. Kde y je větší nebo rovno trojnásobku hodnoty *hellotime* a zároveň není menší než 50 milisekund.

Příkaz pro nastavení časovačů je dostupný v Cisco IOS od verze 10.0. Ve verzi 11.2 bylo přidáno klíčové slovo `msec`, ve verzi 12.2 se změnila spodní hranice pro zadání parametrů *`hellotime`* a *`holdtime`* na hodnoty uvedené zde.

2.6.2 Další příkazy

Pro nastavení sledování objektu nebo rozhraní slouží následující příkaz:

```
Router(config-if)# standby [group-number] track {object-id |
interface [decrement penalty]} [shutdown]
```

Kde jednotlivé parametry znamenají:

- *`object-id`* je číslo objektu, který má být sledován.
- *`interface`* znázorňuje typ a číslo rozhraní, které bude sledováno.

- *penalty* je volitelný parametr znázorňující, o kolik se sníží priorita směrovače v momentě, kdy vypadne sledovaný objekt či rozhraní. Priorita směrovače se zvýší o zadanou hodnotu v momentě, kdy objekt či rozhraní začne být opět aktivní. Možné hodnoty jsou od 0 do 255. Výchozí hodnota je 10.

- `shutdown` volitelný příznak změní HSRP skupinu do inicializačního stavu na základě stavu sledovaného objektu.

Příkaz je součástí Cisco IOS od verze 10.3. Od verze 12.4(9)T, kdy bylo přidáno klíčové slovo `shutdown`, je příkaz v podobě zmíněné výše.

Pro nastavení autentizačního řetězce se používá příkaz `standby authentication` v následující podobě:

```
Router(config-if)# standby [group-number] authentication
{text string | md5 {key-string [0|7] key [timeout seconds] |
key-chain name-of-chain}}
```

Kde jednotlivé parametry znamenají:

- *string* je autentizační řetězec. Může být 8 znaků dlouhý a výchozí hodnota je „cisco“.

- `md5` je příznak pro *message digest 5* autentizaci.

- *key* specifikuje tajný klíč pro md5 autentizaci. Může obsahovat až 64 znaků. Je doporučeno použít alespoň 16 znaků.

- `0` je volitelný parametr znázorňující nešifrovaný klíč.

- `7` je volitelný parametr znázorňující zašifrovaný klíč[2].

- *seconds* je volitelný parametr udávající dobu v sekundách, po kterou HSRP bude přijímat zprávy autentizované starým i novým klíčem. To umožňuje po tuto dobu provést změnu klíče u všech zařízení v HSRP skupině. Tímto se dá zamezit zbytečnému přehazování *active routerů*.

- *name-of-chain* identifikuje skupinu autentizačních klíčů.

Následující příklad ukazuje konfiguraci MD5 autentizace na konkrétním rozhraní pro HSRP skupinu číslo 1 používající tajný klíč „345890“ s dobou 30 sekund pro změnu.

Příklad:
```
Router(config-if)# standby 1 authentication md5 key-string
345890 timeout 30
```

Příkaz pro nastavení autentizace je v Cisco IOS od verze 10.0. Od verze 12.1 je přidáno klíčové slovo `text`. Od verze 12.3(2)T je přidáno klíčové slovo `md5` s jeho parametry.

Příkaz užívající se pro HSRP, který běží nad FDDI (*fiber distributed data interface*, standard pro přenos dat v sítích LAN), slouží pro změnu časového intervalu, ve kterém probíhá ARP komunikace za účelem obnovení MAC tabulky, je následující:

[2]Při volbě `key-string 7` je šifrování prováděno pomocí Vigenerovy šifry [8], která není považována za dostatečně silnou a lze prolomit. Stejně je tomu i u VRRP a GLBP.

```
Router(config-if)# standby mac-refresh seconds
```

Kde *seconds* udává velikost intervalu v sekundách. V tomto intervalu je odeslán paket pro aktualizaci MAC tabulky. Maximální hodnota je 255 sekund. Výchozí je 10 sekund. Podpora příkazu na Cisco IOS je od verze 12.0.

2.6.3 Kontrola konfigurace

Pro kontrolu se používají příkazy:

- `Router# show standby [interface-id [group]]` zobrazí podrobný popis rozhraní. V jakém je stavu, virtuální IP adresu, virtuální MAC adresu, podrobnosti o časovačích, IP adresu *active routeru* a další informace.
- `Router# show standby [brief]` přehledně zobrazí, v jakém stavu jsou jednotlivá rozhraní a adresu *active* a *standby routeru*.
- `Router# debug standby [errors|events|packets|terse]` zapne vypisování všech informací ohledně změn stavu a *Hello* paketů. Výpis lze omezit volitelným parametrem:
  - `errors` zobrazí chybové zprávy při HSRP komunikaci.
  - `events` zobrazí události při HSRP komunikaci.
  - `packets` zobrazí ladící informace o paketech týkajících se HSRP.
  - `terse` zobrazí v omezené míře všechny výše zmíněné.

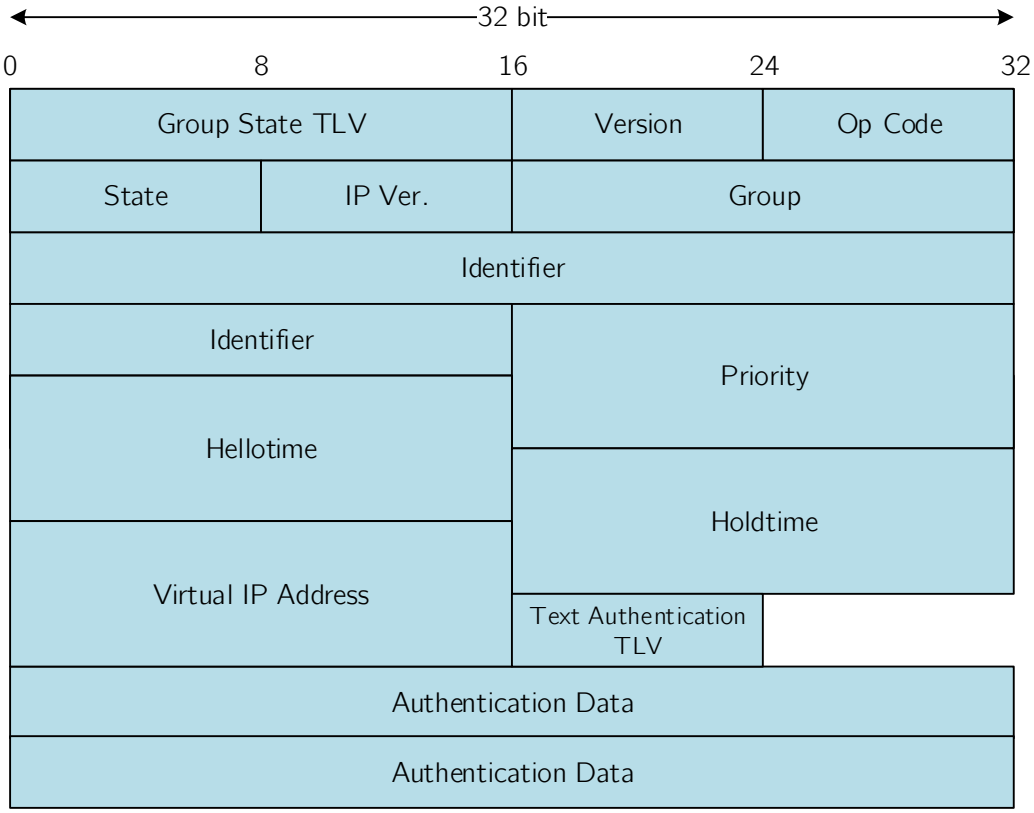
Příkazy jsou dostupné v Cisco IOS od verze 10.0. Poslední úpravy výpisů proběhly ve verzi 12.4(24)T.

2.7 Hot standby router protocol version 2

HSRP verze 2 obsahuje tyto následující odlišnosti [6].

- Protokol rozesílá a je schopný se učit hodnoty jednotlivých časovačů v milisekundách (HSRP verze 1 podporuje pouze rozesílání a učení se hodnot v sekundách). To umožňuje lepší konzistenci.
- Rozšíření číslování skupin (*group number*) na rozsah 0 až 4095.
- S rozšířením rozsahu skupin souvisí změna MAC adresy pro HSRP skupinu. Ta je nyní `0000.0C9F.Fxxx`, kde `xxx` znázorňuje číslo HSRP skupiny v hexadecimálním tvaru.
- Za účelem zlepšení ladění a hledání problému při nastavování HSRP je do paketu verze 2 přidán šesti bytový identifikátor, který jednoznačně identifikuje odesilatele zprávy. Typicky se jedná o MAC adresu rozhraní.
- Multicastová adresa využívaná v *Hello* zprávách je 224.0.0.102.

Při změně verze se HSRP skupina musí znovu inicializovat, protože má novou MAC adresu. HSRP verze 2 má jiný formát paketu (viz obrázek 2.4), a tak protokol není schopný spolupracovat s verzí 1. Avšak různé verze mohou být spuštěné na různých fyzických rozhraních jednoho zařízení (viz výstup příkazu `show standby` v příloze D.1).



Obrázek 2.4: Struktura HSRPv2 paketu odchycená ve Wiresharku (viz příloha D.2).

Kapitola 3

Virtual router redundancy protocol

VRRP (*Virtual router redundancy protocol*) [11, 18] je velmi podobný protokolu HSRP (viz předchozí kapitola 2). Jedná se o IETF standard, který podporují Cisco zařízení. Jeho funkcionalita je popsána v RFC 3768 [10] a verze 3 v RFC [13]. Já zde popisuji zejména verzi 3. Vzhledem k tomu, že protokol pracuje jak nad IPv4, tak nad IPv6, používám v textu zkratku IPvX, kde X znázorňuje verzi 4 nebo verzi 6.

VRRP protokol vytváří ze dvou a více směrovačů jednu výchozí bránu. O přeposílání paketů se stará jeden aktivní směrovač v rámci VRRP skupiny, takzvaný *master router*. Všechny ostatní směrovače jsou záložní, neboli *backup routery*. Aktivní směrovač je zvolen mezi směrovači ve VRRP skupině na základě priority.

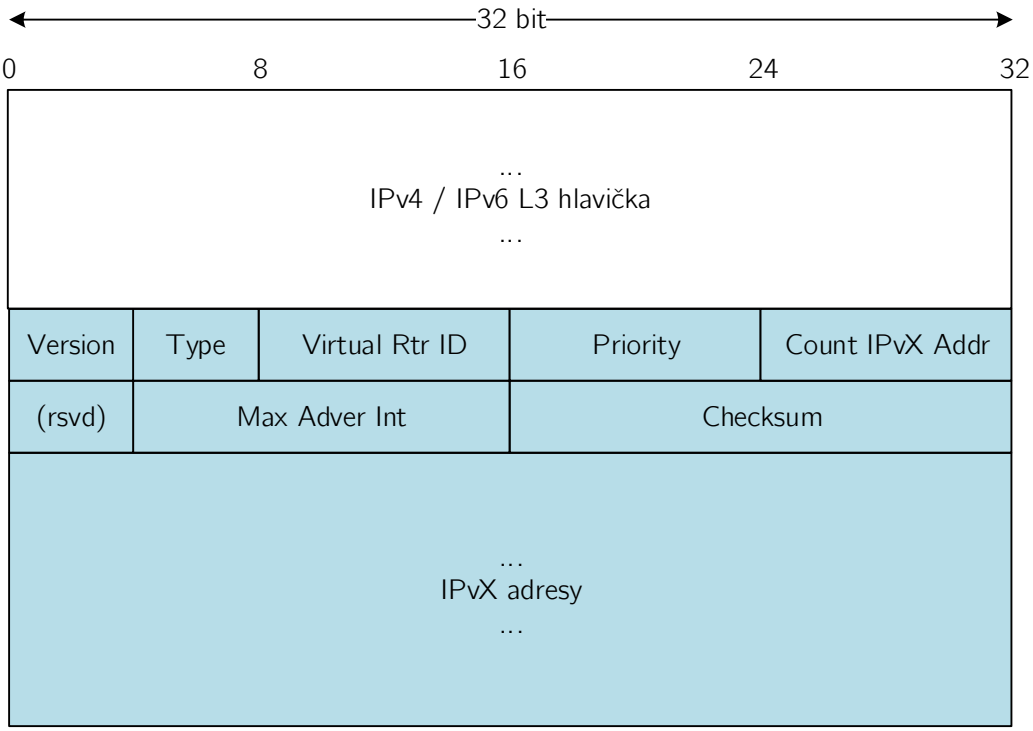
3.1 Vysvětlení terminologie

Protokol VRRP využívá následující termíny:

- *VRRP router* je směrovač, na kterém je spuštěn VRRP protokol. Může být součástí jednoho nebo více virtuálních směrovačů.
- *Virtual router* neboli virtuální směrovač (taktéž *VRRP group* nebo VRRP skupina) je objekt spravovaný VRRP, který se jeví jako výchozí brána pro uživatele sítě. Skládá se z *virtual router ID* a jedné nebo více přidružených IPvX adres.
- *IP address owner* je VRRP router, jehož adresa rozhraní se shoduje s virtuální adresou.
- *Primary IP address* je u IPv4 adresa vybrána z množiny adres jednotlivých rozhraní. U IPv6 je to link-local adresa rozhraní, přes které jsou pakety přeposílány.
- *Virtual router master* nebo jen *master* je VRRP router, který se stará o směrování paketů. Jestliže je dostupný *IP address owner*, tak je vždy *master*, jinak je to směrovač s nejvyšší prioritou.
- *Virtual router backup* nebo jen *backup* jsou ostatní VRRP směrovače, které přebírají zodpovědnost za přeposílání paketů v případě, že *master* přestane fungovat.

- *Advertisement* nebo oznámení je označení typu zprávy, pomocí které mezi sebou jednotlivé směrovače ve VRRP skupině komunikují.
- *Master advertisement interval* je časový interval mezi jednotlivými oznamovacími zprávami od *master routeru*.

3.2 Formát paketu



Obrázek 3.1: Struktura VRRP paketu. Převzato z [13].

Informace týkající se VRRP jsou v paketu řazeny za IP hlavičku. Podstatné informace skryté v IP hlavičce jsou následující:

- `Source Address` je primární IPv4 (resp. IPv6) adresa rozhraní, ze kterého byl paket odeslán.
- `Destination Address` je IPv4 multicastová adresa, která je u VRRP `224.0.0.18`. Pro IPv6 je to `FF02::12`. Jedná se o adresu pro lokální multicast. Směrovač nesmí přeposlat paket s touto cílovou adresou bez ohledu na jeho TTL (respektive Hop Limit).
- `TTL a Hop Limit` je hodnota, která musí být nastavena na 255. Směrovač, který přijme jinou hodnotu, musí paket zahodit.
- `Protocol a Next Header` je číslo protokolu pro VRRP (112).

Jednotlivá pole paketu týkající se VRRP jsou popsána níže:

- `Version` specifikuje verzi protokolu. Může nabývat hodnot 1 – 3.
- `Type` udává typ VRRP paketu. Jediný typ, který tento protokol obsahuje je `advertisement` s binární hodnotou 1. Pakety jiného typu jsou zahozeny.
- `Virtual Rtr ID (VRID)` je identifikace virtuálního směrovače v lokální síti. Možné hodnoty jsou 1 – 255.
- `Priority` znázorňuje prioritu směrovače, který zprávu odeslal. Vyšší hodnota značí vyšší prioritu. Priorita VRRP směrovače, který je *IP address owner* musí být 255. *Backup routery* musí mít prioritu v rozmezí od 1 do 254. Výchozí hodnota pro *backup routery* je 100. Hodnota 0 se využívá v případě, že *master router* chce sdělit *backup* směrovačům, že potřebuje, aby se některý z nich stal *master*.
- `Count IPvX address` je počet adres obsažených v aktuálním VRRP oznámení.
- `Rsvd` je příznak, který musí být při odeslání nastaven na 0 a při příjmu ignorován.
- `Maximum advertisement interval (Max adver int)` je časový interval mezi jednotlivými zprávami typu `Advertisement`. Udává se v setinách sekundy a výchozí hodnota je 100 (neboli 1 sekunda).
- `Checksum` je kontrolní součet pro detekci chyb ve zprávě.
- `IPvX address(es)` je jedna nebo více IPvX adres asociovaných s *virtual routerem*, jejichž počet udává pole `count IPvX address`. Pole obsahuje buď jen IPv4 adresy, nebo jen IPv6 adresy. U IPv6 musí být první zadaná adresa lokální adresou asociovanou s virtuálním routerem.

3.3 Další možnosti protokolu

Uvádím rozšiřující parametry a možnosti, které protokol poskytuje:

- *Preempt mode* umožňuje zapnout, či vypnout preempci. Ve výchozím stavu je zapnuta.
- *Accept Mode* kontroluje, zda *master* akceptuje pakety adresované pro *IP address ownera*, i když *master* není *IP address owner*. Ve výchozím stavu tyto pakety neakceptuje.
- *Load balancing* umožněn pomocí možnosti nastavení více VRRP skupin na jednom fyzickém rozhraní. Těchto skupin může být až 255.
- *Object tracking* umožňuje optimalizovat volbu *master routeru* tím, že je sledován stav objektu a v případě výpadku se sníží priorita směrovače.
- Možnost nastavení vícenásobných IP adres, kdy VRRP může být nakonfigurováno pro každou podsíť na ethernetovém rozhraní zvlášť.

3.4 Popis protokolu

Jednotlivé směrovače patřící do VRRP skupiny mezi sebou komunikují *Advertisement* zprávami pomocí multicastové adresy 224.0.0.12 (FFE002::12 u IPv6) na protokolu IP a portu číslo 112. VRRP skupiny mají stanovenou MAC adresu závisející na čísle jednotlivých VRRP skupiny. MAC adresa má následující tvar:

```
IPv4: 0000.5e00.01xx
IPv6: 0000.5e00.02xx
```

Kde `xx` znázorňuje hexadecimálně zapsané číslo VRRP skupiny. Každý směrovač pracuje s následujícími časovači:

- *Advertisement interval* je časový interval mezi jednotlivými oznamovacími zprávami.
- *Skew time* je doba, za kterou zaručuje volbu nového *master routeru*. Vypočítá se jako:

$$\frac{(256 - priorita) \cdot MasterAdvertInterval}{256}$$

- *Master down interval* je doba, po kterou *backup routery* zjišťují, že *master* je nefunkční. Vypočítá se jako $(3 \cdot MasterAdverInterval) + Skew\_Time$.

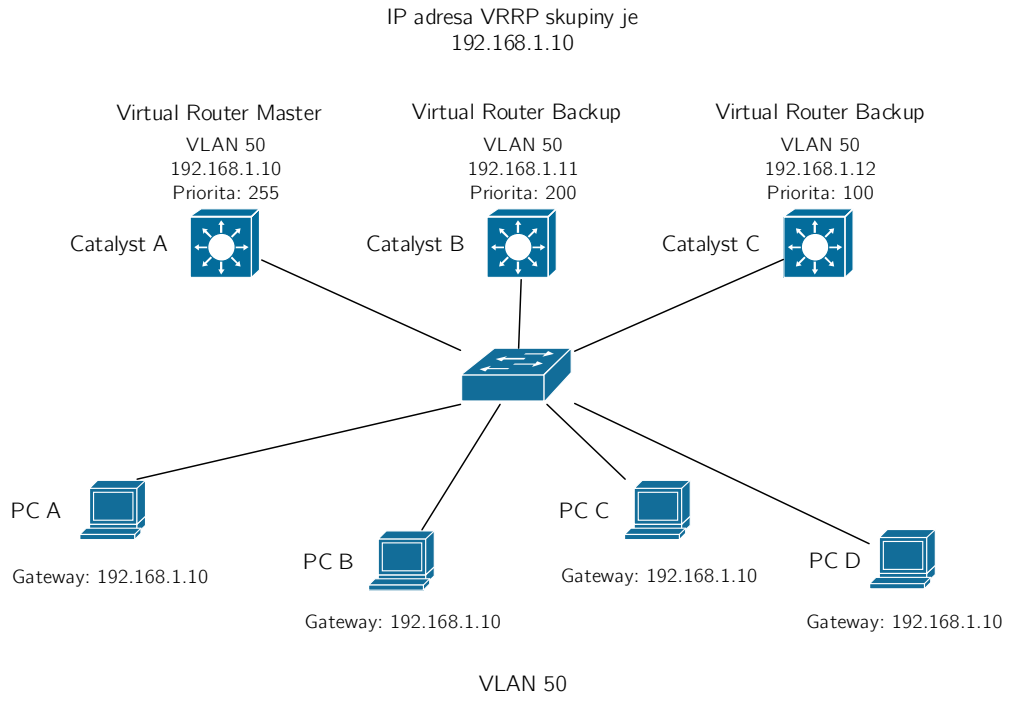
Proces převzetí role pak probíhá tak, že směrovač v roli *master* posílá po uplynutí *advertisement intervalu* ohlášení o svém stavu. Jestliže směrovač nepošle ohlášení, ostatní směrovače ve skupině to chápou jako selhání směrovače. Po uplynutí *master down* intervalu začne běžet *skew time* časovač. Jakmile skončí, tak směrovač s vyšší prioritou přejde do stavu *master* jako první.

Na obrázku 3.2 je znázorněno zapojení VRRP skupiny. Roli *master* získal L3 switch *Catalyst A*, a tak k němu přistupují všechny 4 PC jako k výchozí bráně. V případě, že by se *Catalyst A* odpojil, tak by mezi sebou zbývající dva L3 switche ve stavu *backup* soupeřili, jak je popsáno výše. Roli *master* by získal L3 switch *Catalyst B*.

3.4.1 Stavový automat

Každý směrovač VRRP skupiny se nachází v jednom ze stavů následujícího stavového automatu (viz obrázek 3.3).

- **Init**
  Účel tohoto stavu je čekat na spouštěcí akci. Po přijetí spouštěcí akce je zkontrolována priorita. Pokud je rovna 255, tak router odešle oznámení a přepne se do stavu *master*. V případě priority různé od 255 se router přepne do stavu *backup*.
- **Backup**
  Úkolem směrovačů ve stavu *backup* je sledovat dostupnost a stav *master routeru*. Tyto směrovače se neúčastní aktivního provozu, pouze sledují VRRP oznámení od *master routeru*. Jestliže směrovač nepřijme oznámení před vypršením *master down intervalu*, tak provede přechod do stavu *master*. Je-li spuštěná preempce a směrovač ve stavu *backup* zjistí, že má vyšší prioritu než aktuální *master router*, tak dojde k předání rolí.



Obrázek 3.2: Ukázka zapojení VRRP nad L3 přepínači.

- **Master**
  Směrovač v tomto stavu směruje všechny pakety určené pro *virtual router* a jeví se tak jako výchozí brána pro lokální síť. Zároveň odesílá pravidelně zprávy typu *Advertisement*. V případě obdržení zprávy *Advertisement* porovná priority a jestliže je lokální priorita menší než priorita ve zprávě, tak směrovač přejde do stavu *backup*. V případě shody priorit dojde k porovnání IP adres odesilatele a příjemce, které rozhodne o tom, zda přejde do stavu *backup*, nebo zda zůstane ve stavu *master*.

3.5 Podpora na Cisco zařízeních

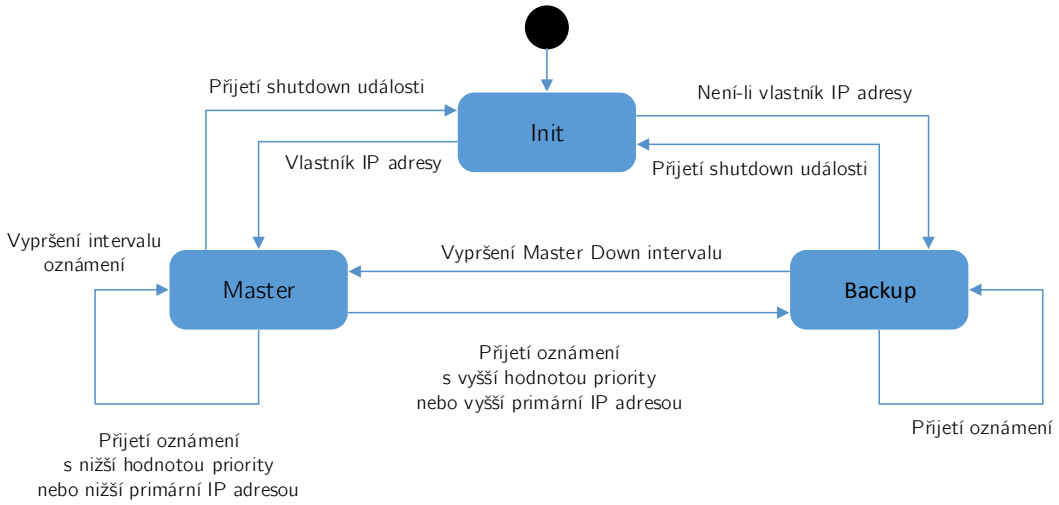
Cisco zařízení podporují protokol VRRP. Uvádím zde možné konfigurační příkazy, jejich popis a u složitějších příkazů ukazuji příklad. U každého příkazu je uvedena verze Cisco IOS, od které je podporován. Jednotlivé příkazy čerpám z Cisco IOS referenční příručky pro FHRP příkazy [7]. Ladící příkazy potom z Cisco příručky pro `debug` příkazy [4].

Protokol je podporován na zařízeních s Cisco IOS od verze 12.0(18)ST.

3.5.1 Základní příkazy

Protokol se na Cisco zařízeních spouští následujícím příkazem na rozhraní, na kterém bude vytvořena VRRP skupina:

```
Router(config-if)# vrrp group ip ip-address [secondary]
```



Obrázek 3.3: Stavový automat protokolu VRRP. Převzato z [18].

Kde jednotlivé parametry znamenají následující:

- ***group*** znázorňuje číslo VRRP skupiny.
- ***ip-address*** znázorňuje IP adresu VRRP skupiny. Adresa musí být ve stejné podsíti jako IP adresa rozhraní.

Příkaz je podporován na Cisco IOS od verze 12.0(18)ST.

Dále je možné nastavit prioritu příkazem:

```
Router(config-if)# vrrp group priority level
```

Kde parametr ***level*** určuje prioritu zařízení. Vyšší priorita určuje, který směrovač ve skupině bude sloužit jako primární. V případě shodných priorit se stává výchozím směrovačem směrovač s vyšší primární IP adresou. Výchozí hodnota priority je 100. Příkaz je na Cisco IOS podporován od verze 12.0(18)ST.

Preempce se nastavuje následujícím příkazem:

```
Router(config-if)# vrrp group preempt [delay minimum seconds]
```

Ve výchozím nastavení je preempce povolená. Lze vypnout příkazem `no vrrp group preempt`. Parametr `delay` určuje zpoždění v sekundách po němž směrovač převezme funkci od stávajícího *master* směrovače. Výchozí hodnota zpoždění je 0 sekund. Příkaz je na Cisco IOS podporován od verze 12.0(18)ST.

Nastavení intervalu mezi oznámeními *master virtual routeru* ostatním směrovačům ve VRRP skupině se provádí následovně:

```
Router(config-if)# vrrp group timers advertise [msec] interval
```

Kde jednotlivé parametry znamenají následující:

- `msec` je volitelný parametr, který změní jednotku oznamovacího intervalu ze sekund na milisekundy.
- *`interval`* je parametr udávající časový interval mezi jednotlivými oznámeními od *master virtual router*. Výchozí hodnota je 1 sekunda. Rozsah možných hodnot je 1 až 255 sekund. Při zadání parametru `msec` jsou možné hodnoty 50 až 999 milisekund.

Příkaz je na Cisco IOS podporován od verze 12.0(18)ST.

V případě, že mají směrovače nastaveny rozdílné hodnoty `interval`, nebude *backup* přijímat ohlášení a přejde do stavu *master*. Tomuto chování je možné zabránit následujícím příkazem, který směrovači povoluje naučit se hodnotu oznamovacího intervalu od *master* routeru:

```
Router(config-if)# vrrp group timers learn
```

Příkaz je na Cisco IOS podporován od verze 12.0(18)ST.

Pro nastavení sledování objektu se používá tento příkaz:

```
Router(config-if)# vrrp group track object-number [decrement
priority]
```

Kde *`object-number`* znázorňuje číslo sledovaného objektu v rozmezí od 1 do 500. Volitelný parametr `decrement` *`priority`* udává, o kolik se sníží (nebo zvýší) priorita směrovače, když sledovaný objekt přestane fungovat (nebo začne). Výchozí hodnota je 10. Může nabývat hodnot od 1 do 255. Příkaz je na Cisco IOS podporován od verze 12.3(2)T.

Pro nastavení autentizace paketů VRRP v rámci skupiny se používá tento příkaz:

```
Router(config-if)# vrrp group authentication {string | md5
{key-chain key-chain | key-string [0|7] string [timeout seconds
}}
```

Kde jednotlivé možnosti autentizace jsou následující:

- *`string`* nastaví *plain text* autentizaci zadaným řetězcem. Tento parametr je možné zadat také s klíčovým slovem *text*. Příkaz bude vypadat následovně: `text` *`string`*.
- `md5` nastaví autentizaci pomocí MD5. Možnosti jsou následující:
  - `key-chain` nastavuje autentizaci využívající vytvořenou klíčenku. Parametr *`key-chain`* se musí shodovat s názvem klíčenky.
  - `key-string` nastaví tajné heslo pro MD5 autentizaci. Má dva volitelné parametry 0 a 7. Kde 0 udává, že klíč bude nezašifrovaný. 7 udává, že klíč bude šifrovaný. Parametr *`string`* může být až 64 znaků dlouhý a udává tajné heslo. Doporučuje se délka alespoň 16 znaků.
  - `timeout` *`seconds`* je volitelný parametr udávající dobu v sekundách, po kterou bude VRRP přijímat zprávy založené na starém a na novém klíči.

Například konfigurace hesla v čisté podobě může vypadat následovně:

Příklad:

```
Router(config-if)# vrrp 1 authentication xb93arw
```

Konfigurace hesla pomocí klíčenky se šifrováním MD5 může vypadat následovně:

Příklad:

```
Router(config)# key chain klicenka
Router(config-keychain)# key 0
Router(config-keychain-key)# key-string 49391409513afstge
Router(config)# interface Ethernet0/1
Router(config-if)# vrrp 1 ip 10.21.0.10
Router(config-if)# vrrp 1 authentication md5 key-chain klicenka
```

Příkaz je na Cisco IOS podporován od verze 12.0(18)ST. Ve verzi 12.3(14)5 bylo přidáno klíčové slovo `md5` a parametry s ním související.

3.5.2 Kontrola konfigurace

Pro kontrolu správné funkčnosti konfigurace se používá příkaz `show vrrp`, který zobrazí, na kterých rozhraních je protokol spuštěný, jakou má směrovač roli ve VRRP skupině a informace o časových intervalech. Výpis je možné omezit pouze na zobrazení všech rozhraní, nebo pouze na některá rozhraní, eventuálně pomocí parametru *all* zobrazit i VRRP skupiny v neaktivním stavu.

```
Router# show vrrp [all | brief | interface]
```

Příkaz je dostupný na Cisco IOS od verze 12.0(18)ST. Od verze 12.3(14)T jsou upraveny informace týkající se MD5 autentizace. Od verze 12.4(24)T je ve výstupu skryto heslo při použití MD5 autentizace nebo *plain text* autentizace. Od verze 15.3(3)M výstup zobrazuje informace o sledovaném objektu.

Také je možné zapnout vypisování ladících výpisů pomocí příkazu:

```
Router# debug vrrp [all | errors | events | packet]
```

Jednotlivé parametry ovlivňují vypisované informace následovně:

- `all` zapne zobrazování výpisů týkajících se chyb, událostí a přechodů mezi stavy.
- `errors` zapne zobrazování chybových oznámení týkajících se VRRP.
- `events` zapne zobrazování událostí, které nastanou při běhu VRRP.
- `packet` zapne zobrazování výpisů týkajících se odeslaných a přijatých VRRP paketů.

Příkaz je dostupný od verze Cisco IOS 12.0(18)ST.

Kapitola 4

Gateway load balancing protocol

GLBP neboli *Gateway load balancing protocol* [15, 2, 11] je Cisco proprietární protokol umožňující podobně jako HSRP a VRRP zajistit redundanci síťové brány. U protokolů HSRP a VRRP je také možnost zajistit *load balancing*, ale je potřeba více skupin a tím pádem i více nastavování a vzniká tak větší prostor pro chyby. GLBP byl navržen tak, aby umožnil přímočařeji nastavit *load balancing* při zajišťování redundance síťové brány.

K poskytování redundance brány využívá GLBP několik směrovačů. Tyto směrovače jsou přiřazeny do GLBP skupiny. Na rozdíl od HSRP nebo VRRP, kde byl pouze jeden směrovač aktivní, zde se podílejí na přeposílání síťového provozu všechny směrovače a poskytují tak *load balancing*. Výhodou je, že klienti mohou mít nastavenou stejnou adresu výchozí brány směřující na virtuální adresu GLBP skupiny. *Load balancing* je poskytován pomocí ARP zpráv, kdy aktivní směrovač GLBP skupiny zašle klientům v ARP odpovědi MAC adresu konkrétního směrovače. Výsledkem je, že různí klienti mají v ARP tabulce k jedné IP adrese výchozí brány přiřazeny různé MAC adresy směrovačů.

4.1 Vysvětlení terminologie

Ačkoli je názvosloví odlišné oproti protokolům VRRP a HSRP, tak jsou významy jednotlivých rolí směrovačů v GLBP skupině podobné jako tomu bylo u rolí směrovačů v HSRP a VRRP.

- *Active virtual gateway* neboli AVG je směrovač s nejvyšší prioritou. V případě shody priorit je to směrovač s nejvyšší hodnotou IP adresy. AVG může být pouze jeden směrovač v GLBP skupině. Tento směrovač se stará o zasílání ARP odpovědí.
- *Active virtual forwarder* neboli AVF jsou směrovače starající se o přeposílání paketů. V GLBP skupině mohou být maximálně 4. AVG je také AVF.
- *Backup AVG/AVF* jsou záložní směrovače ve skupině, které nejsou ani AVG ani AVF. Jestliže přestane být AVG, nebo jeden z AVF směrovačů dostupný, tak za něj *backup* směrovač převezme roli.
- *Virtual forwarder* (VF) je směrovač v rámci GLBP brány, který obdržel virtuální MAC adresu.
- *Primary virtual forwarder* (PVF) je VF, kterému přiřadí virtuální MAC adresu AVG.

- *Secondary virtual forwarder* (SVF) je VF, který se naučil virtuální MAC adresu z *Hello* zprávy od *primary virtual forwardera*. Jedná se o záložní směrovače k AVF.

Směrovač může být PVF pro jednu virtuální MAC adresu a zároveň SVF pro jinou.

4.2 Load Balancing algoritmy

GLBP využívá následující algoritmy pro zvládání rozložení zátěže:

- **Round robin** je výchozí metodou pro rozložení zátěže. Principem je, že se postupně střídají všechny MAC adresy v ARP odpovědi. Předpokládá, že každý klient odesílá a přijímá stejné množství paketů.
- **Váhový**, kde váha GLBP rozhraní určuje poměr provozu v rámci skupiny směrovačů. Vyšší váha znamená vyšší provoz tohoto rozhraní směrovače.
- **Podle koncových stanic**, kdy každý klient generující ARP dotaz dostává vždy ARP odpověď se stejnou MAC adresou. Kvůli zachování rovnováhy provozu se tato metoda nedoporučuje u malého počtu klientů.

4.3 Formát paketu

GLBP využívá tři typy TLV[1] (*type-length-value*). Pakety pro zasílání informací ohledně VG, pakety pro zasílání informací ohledně VF a pakety zajišťující autentizaci. Tyto tři typy s sebou nesou vždy část paketu, která obsahuje následující informace (viz obrázek 4.1):

- `Version`, udávající verzi GLBP zprávy. Hodnota je vždy 1.
- `Group`, znázorňující číslo GLBP skupiny.
- `Owner ID`, znázorňující jedinečný identifikátor zařízení. Jedná se o MAC adresu rozhraní.

Část paketu sloužící pro komunikaci VG je obsažena ve zprávě maximálně jednou a obsahuje tato pole (viz obrázek 4.2):

- `Type` popisuje, jaký typ zprávy paket obsahuje. Možnosti jsou následující:
  - 1: *Hello* zpráva. Jedná se o typ zprávy při komunikaci VG.
  - 2: *Request – response* zpráva. Jedná se o typ zprávy při komunikaci VF.
  - 3: *Auth* zpráva. Používá se při nastavení autentizace. Struktura těchto paketů se liší v závislosti na zvolené metodě autentizace.
- `Length` udává délku celé zprávy.
- `VG State` popisuje aktuální stav virtuální brány. Možné hodnoty jsou následující:
  - 04: *Listen*

[1] V této práci nazývám GLBP zprávu obsahující různé kombinace TLV jako *Hello* zpráva. V případě, že se jedná jen o *Hello* část TLV, je to zdůrazněno.

  - 08: *Speak*
  - 16: *Standby*
  - 32: *Active*

  Zbývající stavy *Disabled* a *Init* jsou stavy, kdy směrovač ještě nemá všechny potřebné informace, aby mohl zasílat zprávy pod těmito stavy po síti. Tyto stavy se proto nepřenášejí.

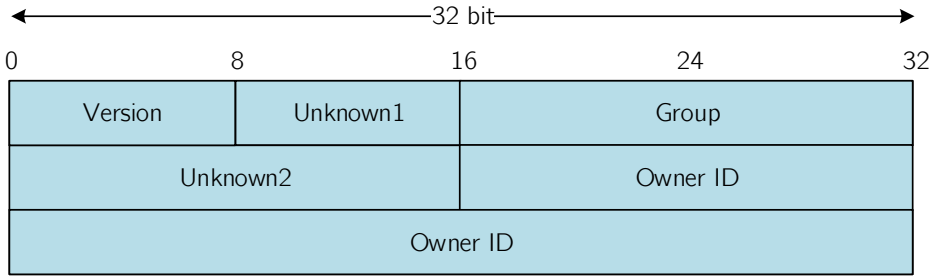
- `Priority` je hodnota využívaná při volbě VG.
- `Hellotime` je časová prodleva v sekundách mezi odesláním jednotlivých *Hello* zpráv. Ve Wiresharku se značí jako *helloint*. Výchozí hodnota je 3 sekundy.
- `Holdtime` je časová prodleva v sekundách určující maximální povolený interval mezi dvěma *Hello* zprávami po kterém směrovač usoudí, že AVG je vypnutý. Ve Wiresharku se značí jako *holdint*. Výchozí hodnota je 10 sekund.
- `Redirect` určuje dobu zadanou v sekundách, po kterou AVG bude přesměrovávat nové ARP požadavky v případě výpadku některého AVF na tento nefunkční AVF. Po vypršení *redirect timeru* s nefunkčním VF přestane AVG tento VF zahrnovat do *load balancingu*. Výchozí hodnota je 600 sekund.
- `Timeout` je doba zadaná v sekundách, po kterou AVG bude mít uloženou virtuální MAC adresu AVF v případě, že tento AVF vypadne. Jestliže se nestihne za tuto dobu obnovit komunikace s poškozeným AVF, tak je adresa smazána a klienti musí zaslat nový ARP dotaz. Výchozí hodnota je 14400 sekund.
- `Address type` udává typ adresy. Možné hodnoty jsou:
  - 1: IPv4
  - 2: IPv6
- `Address length` udává délku zadané virtuální IPvX adresy.
- `Virtual IPvX` udává jakou virtuální IPvX adresu bude daná GLBP brána používat.

TLV sloužící pro komunikaci jednotlivých VF může být obsazeno ve zprávě až čtyřikrát (v případě, že směrovač je aktuálně v roli *active* u všech VF). Oproti výše uvedené zprávě obsahuje navíc tyto odlišnosti (viz 4.3):

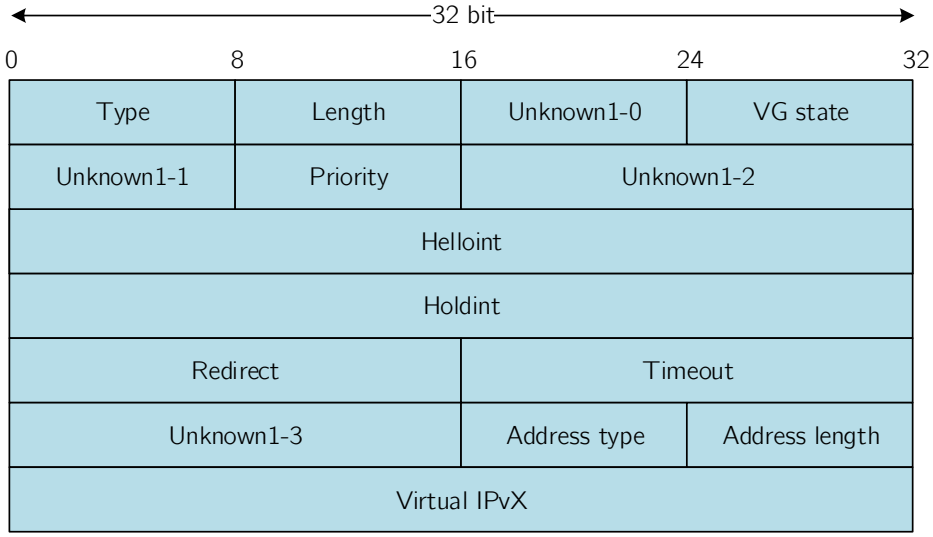
- `Forwarder` je identifikační číslo v rámci GLBP skupiny přidělené od AVG.
- `VF state` znázorňující v jakém stavu se nachází *virtual forwarder*. Možné hodnoty jsou následující:
  - 00: *Unknown*
  - 04: *Listen*
  - 32: *Active*

  Stav *Unknown* sdružuje stavy *Disabled* a *Init*, kdy směrovač nemá ještě kompletní konfiguraci.

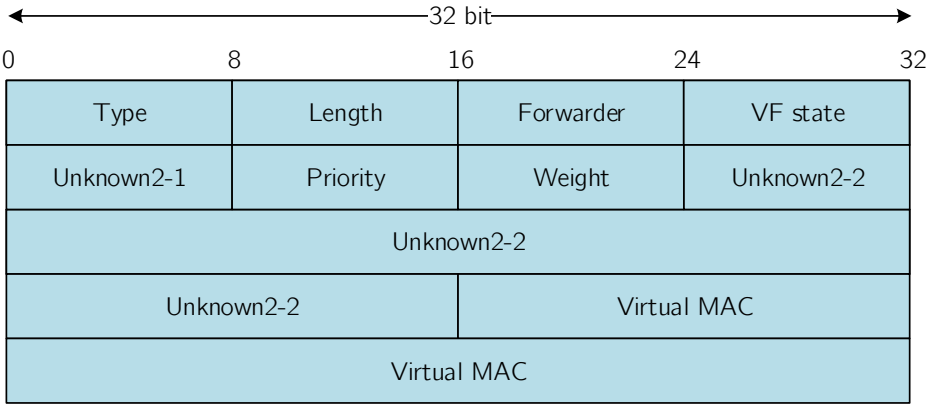
- `Priority` udává odlišnou prioritu než v případě VG priority. Tato priorita nabývá dvou hodnot:
  - 167 - V případě, že se jedná o *Primary virtual forwarder*.
  - 135 - V případě, že se jedná o *Secondary virtual forwarder*.
- `Weight` udává nastavenou váhu aktuálního směrovače. Výchozí hodnota je 100.
- `Virtual MAC` udává virtuální MAC adresu, která byla směrovači přidělena od AVG směrovače.



Obrázek 4.1: Obecná část struktury GLBP paketu odchycená ve Wiresharku (viz příloha E.1).



Obrázek 4.2: Struktura části GLBP paketu typu *Hello* odchycená ve Wiresharku (viz příloha E.1).



Obrázek 4.3: Struktura části paketu typu *Request – response* odchycená ve Wiresharku (viz příloha E.1).

4.4 Popis protokolu

Po spuštění protokolu na směrovačích si mezi sebou jednotlivé směrovače volí, který směrovač se stane AVG. Vítězí směrovač s nejvyšší prioritou, eventuálně s nejvyšší IP adresou. AVG se dále stará o odeslání ARP odpovědí na ARP dotazy směrované na virtuální IP adresu GLBP skupiny. Kterou MAC adresu zvolí, záleží na zvoleném *load balancing* algoritmu (viz kapitola 4.2). AVG také přiřazuje virtuální MAC adresy jednotlivým směrovačům v GLBP skupině. Mohou být použity 4 virtuální MAC adresy v každé skupině. Směrovače, kterým je přidělena virtuální MAC adresa se stávají AVF a starají se o přeposílání paketů směrovaných na jejich virtuální MAC adresu. Ostatní směrovače ve skupině slouží jako *backup* nebo *secondary virtual forwarder*. AVG zasílá pravidelné zprávy *Hello* všem směrovačům ve skupině a očekává odpověď. Jestliže ta nepřijde během *holdtime*, tak AVG předpokládá, že směrovač selhal a přiřadí roli AVF jinému směrovači. Tyto časovače fungují obdobně jako u protokolu HSRP.

AVF směrovače obdrží virtuální MAC adresu ve tvaru `0007.b4xx.xxyy`. Kde prvních 16 bitů je šest nul následované deseti bity znázorňujícími číslo GLBP skupiny. Koncová osmibitová hodnota znázorněna `yy` značí *virtual forwarder number*. Vzhledem k tomu, že je možné přiřadit pouze 4 virtuální MAC adresy, tak toto číslo nabývá hodnot 01 - 04.

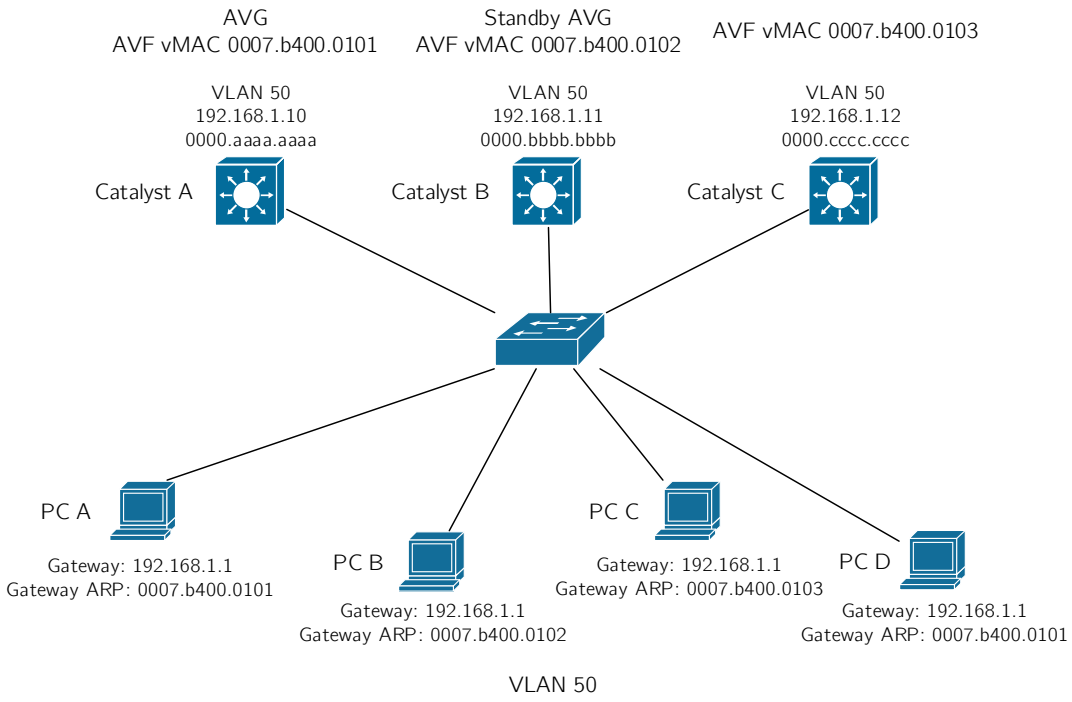
AVG směrovač neustále komunikuje s AVF směrovači pomocí zpráv *Hello* na multicastové adrese 224.0.0.102 a UDP portu 3222.

Protokol využívá následující časovače pro případ, že by byla aktuálnímu AVF přiřazena role dalšího AVF. To může nastat v případě výpadku PVF.

- *Redirect timer* určuje dobu, po kterou AVG přesměrovává příchozí požadavky na nový AVF. Pokud se během průběhu časovače stihne zotavit nahrazený časovač AVF z výpadku, tak se obnoví původní spojení.
- *Timeout timer* je časový limit, po který se může starý AVF zotavit. Jestliže to nestihne, tak jsou stará MAC adresa a AVF, které ji používalo, vyprázdněny z paměti GLBP směrovačů a klienti musí zaslat nový ARP dotaz.

Na obrázku 4.4 je znázorněna síť, kde jsou součástí GLBP skupiny tři L3 switche.

Catalyst A je zvolen jako AVG a koordinuje celý GLBP proces, odpovídá na všechny ARP dotazy na virtuální router 192.168.1.1. Je zde použitý *load balancing* protokol s metodou *Round robin*.



Obrázek 4.4: Ukázka zapojení GLBP nad L3 přepínači. Převzato z [11].

4.4.1 Stavový automat

Jelikož k protokolu GLBP neexistuje dostatečně podrobná specifikace, musel jsem za pomocí Wiresharku sledovat provoz mezi směrovači Cisco a zhotovil jsem z vypozorovaného provozu níže uvedené stavové automaty.

Stavový automat pro volbu VG jsem znázornil kvůli větší přehlednosti do tabulky 4.1. Směrovače soupeřící o roli virtuální brány (VG) se mohou nacházet v jednom z následujících stavů [2]:

- **Disabled**
  Je stav značící, že GLBP je na zařízení nakonfigurováno, ale nebyla mu přiřazena, nebo se ještě nestihl naučit virtuální IP adresu.

- **Init**
  Je stav, kdy virtuální IP adresa je zařízení známa, ale konfigurace virtuální brány není kompletní.

- **Listen**
  Je stav, kdy virtuální brána přijímá *Hello* pakety a je připravena přepnout se do stavu *speak* v případě, že se virtuální brána ve stavu *active* nebo *standby* stane neaktivní.

- **Speak**
  Je stav, kdy se virtuální brána aktivně pokouší stát se *active* nebo *standby*.
- **Standby**
  Je stav, kdy je směrovač připraven přejít do stavu *active* a stát se AVG.
- **Active**
  Je stav indikující, že aktuální směrovač je AVG.

Zde je popis jednotlivých událostí pro volbu VG, které uvádím v tabulce 4.1:

- `a`: IP adresa GLBP skupiny je nakonfigurována.
- `b`: IP adresa GLBP skupiny je zrušena.
- `c`: Rozhraní je vypnuto.
- `d`: GLBP je zapnuto.
- `e`: GLBP je vypnuto.
- `f`: *Standby timer* vypršel.
- `g`: *Active timer* vypršel.
- `h`: Router přijal *Hello* zprávu vyšší priority od *speak* routeru.
- `i`: Router přijal *Hello* zprávu vyšší priority od *standby* routeru.
- `j`: Router přijal *Hello* zprávu nižší priority od *standby* routeru.
- `k`: Router přijal *Hello* zprávu vyšší priority od *active* routeru.
- `l`: Router přijal *Hello* zprávu nižší priority od *active* routeru.
- `m`: *Hello timer* vypršel.
- `n`: Router přijal *Resign* zprávu od *active* routeru.

Níže uvádím popis akcí pro volbu VG uvedených v tabulce 4.1:

- `A`: Spustí *active timer* hlídající dostupnost AVG.
- `B`: Spustí *standby timer* hlídající dostupnost *standby* VG.
- `C`: Vypne *active timer*.
- `D`: Vypne *standby timer*.
- `E`: V případě neshody časovačů *hellotime* nebo *holdtime* oproti AVG se router tyto hodnoty naučí od AVG.
- `F`: Pošle *Hello* zprávu.
- `H`: Pošle *Resign* zprávu. Jedná se o *Hello* TLV zasílané AVG těsně před tím, než se vzdá role *active*. Toto *Hello* TLV obsahuje hodnoty `Virtual IP`, `Address type` a `Priority` rovny nule, respektive *Unknown*.

	Init (1)	Listen (2)	Speak (3)	Standby (4)	Active (5)	Disabled (6)
a	A,B/2					-/1 A,B,F/2[a]
b	C,D/6	C,D/6	C,D/6	C,D/6	C,D,H/6	
c		C,D/1	C,D/1	C,D/1	C,D,H/1	
d	A,B/2					
e		C,D/1	C,D/1	C,D/1	C,D,H/1	
f		A,B,F/3	D,F/4			
g		A,B,F/3	C,F/5	C,D,F/5		
h		B/2	B/2	B/2		
i			B/2			
j		B,F/3	D,F/4			
k		E,A/2	E,A/3	E,A/4	A,B/3	
l		A,E/2 A,B,F,E/3[b]	A,E/3 C,D,F,E/5[b]	A,E/4 C,F,E/5[b]		
m			F/3	F/4	F/5	
n		A,B,F/3	A/3	C,F/5		

Tabulka 4.1: Tabulka přechodů pro volbu *active* VG a *standby* VG. Výchozí stav je *Disabled*.

[a] Jestliže je IP adresa nakonfigurována ručně, tak je proveden přechod do stavu *init*. Jestliže je naučená od směrovače *active*, tak je vykonán přechod do stavu *listen*.

[b] Druhý řádek buňky se vykoná pouze při nastaveném parametru preempce. Jinak se vykoná první řádek tabulky.

Pro směrovače v roli *virtual forwarder* (VF) existují následující stavy (viz obrázek 4.5):

- **Disabled**
  Značí stav, kdy nebyla ještě přiřazena virtuální MAC adresa. Jedná se o přechodný stav, protože směrovače v tomto stavu jsou smazány.
- **Init**
  Je stav, kdy *virtual forwarder* zná virtuální MAC adresu, ale jeho konfigurace není kompletní.
- **Listen**
  Je stav, kdy VF přijímá *Hello* pakety a je připraven přejít do stavu *active* v případě, že aktuální AVF přestane být dosažitelný.
- **Active**
  Je stav značící, že tento směrovač je AVF a má tedy na starost přeposílání paketů.

Popis událostí mezi jednotlivými přechody v konečném automatu pro volbu VF je následující:

- `a`: VF byla přiřazena MAC adresa.
- `b`: MAC adresa byla smazána z VF. Situace nastává po vypnutí GLBP na rozhraní.
- `c`: *Timeout timer* vypršel.
- `d`: VF je spuštěn.

- `e`: VF je vypnut.
- `g`: Vypršel *Active timer*.
- `h`: *Hello timer* vypršel.
- `i`: VF přijal *Hello* zprávu od směrovače *active* s vyšší prioritou.
- `j`: VF přijal *Hello* zprávu od směrovače *active* s nižší prioritou (bez TLV oznamující VG stav). Situace nastává, když v případě výpadku jiného VF vyprší všem směrovačům *active timer* pro tento VF a všichni přejdou do stavu *active* a tento stav oznamují pouze s TLV týkající se VF.
- `k`: VF přijal *Hello* zprávu od směrovače *active* s nižší prioritou (s TLV oznamující VG stav).
- `l`: Přijetí zprávy *Resign*.

Popis vykonaných akcí, které se objevují v konečném automatu pro volbu VF je následující:

- `A`: Spustí *active timer* pro daný VF.
- `B`: Vypne *active timer* pro daný VF.
- `C`: Pošle *Resign* zprávu. Jedná se o zprávu bez *Hello* TLV, obsahující pouze *Request – response* TLV s informacemi o daném VF. Tuto zprávu zasílá směrovač ve stavu *active* pro daný VF v momentě, kdy se chce vzdát role *active*.
- `D`: Pošle *Hello* zprávu. Při odeslání zprávy se restartuje *hello timer*.
- `E`: Pošle *Request – response* TLV oznamující, že VF přešel do stavu *active*.

4.5 Podpora na Cisco zařízeních

Cisco zařízení podporují protokol GLBP. Uvádím zde možné konfigurační příkazy, jejich popis a ke složitějším příkazům přikládám ukázku konfigurace. U každého příkazu je uvedena verze Cisco IOS, od které je podporován. Jednotlivé příkazy čerpám z Cisco IOS referenční příručky pro FHRP příkazy [7]. Ladící příkazy potom z Cisco příručky pro `debug` příkazy [5].

Protokol je podporován na zařízeních s Cisco IOS od verze 12.2(14)S.

4.5.1 Základní příkazy

Protokol se spouští na rozhraní příkazem:

```
Router(config-if)# glbp group ip [ip-address [secondary]]
```

Kde jednotlivé parametry znamenají:

- *group* značí číslo GLBP skupiny. Může nabývat hodnot od 0 do 1023.

- *ip-address* značí IP adresu GLBP skupiny. Adresa musí být ve stejné podsíti jako IP adresa rozhraní. Jestliže není IP adresa zadána, tak se naučí od jiného zařízení ve skupině. Avšak jestliže se jedná o AVG směrovač, pak se musí IP adresa zadat explicitně, jinak ani ostatní směrovače nebudou vědět, jaká je jejich virtuální IP adresa.

Příkaz je na Cisco IOS podporován od verze 12.2(14)S.

Priorita se nastaví následujícím příkazem.

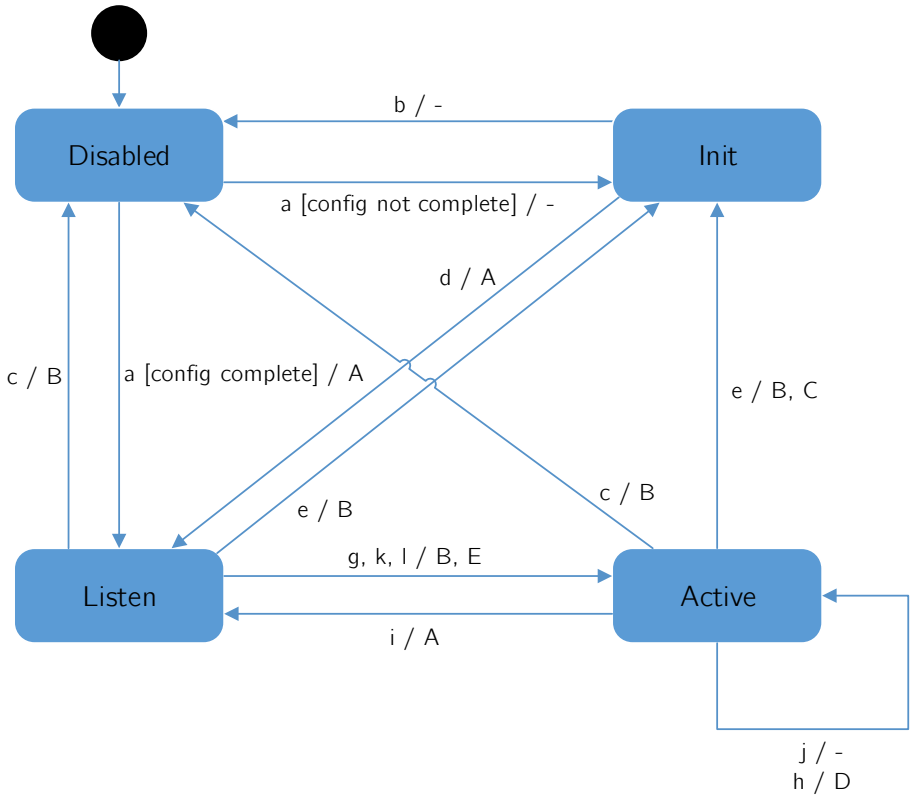
```
Router(config-if)# glbp group priority level
```

Kde *level* značí prioritu směrovače. Možné hodnoty jsou od 1 do 255, kde vyšší číslo značí vyšší prioritu. Výchozí priorita je 100. Příkaz je na Cisco IOS podporován od verze 12.2(14)S.

Preempce se nastavuje příkazem:

```
Router(config-if)# glbp group preempt [delay minimum seconds]
```

Kde *seconds* značí minimální počet sekund předtím, než se směrovač pokusí převzít roli AVG. Může nabývat hodnot od 0 do 3600. Výchozí hodnota je 30 sekund. Preempce je ve výchozím nastavení vypnutá. Příkaz je na Cisco IOS podporován od verze 12.2(14)S.



Obrázek 4.5: Stavový automat protokolu GLBP pro volbu VF.

Časovače *hellotimer* a *holdtime* se nastavují následujícím příkazem:

```
Router(config-if)# glbp group timers [msec] hellotime [msec]
holdtime
```

Kde časovače ***hellotime*** a ***holdtime*** mají stejný význam jako u protokolu HSRP (viz kapitola 2.2). Jejich možné hodnoty jsou následující:

- ***hellotime*** může nabývat hodnot od 1 do 60 sekund. Výchozí hodnota je 3 sekundy.
- ***holdtime*** může nabývat hodnot v rozmezí $< hold time + 1, 160 >$ sekund. Cisco doporučení je zadávat *holdtime* trojnásobně větší, než je hodnota *hellotime*. Výchozí hodnota je 10 sekund.

Příkaz je na Cisco IOS podporován od verze 12.2(14)S.

Časovače *redirect* a *timeout* se nastavují příkazem:

```
glbp group timers redirect redirect timeout
```

Kde parametry znázorňují:

- ***redirect*** značí časový interval v rozmezí od 0 do 3600 sekund. Výchozí hodnota je 600 sekund. Hodnotu 0 se nedoporučuje nastavovat (je zachována spíše z historických důvodů kvůli zpětné kompatibilitě). Jestliže totiž *redirect timer* nikdy nevyprší, tak v případě výpadku směrovače budou uživatelé stále odkazováni na tento směrovač místo *backup* směrovače.
- ***timeout*** je časový interval v rozmezí $< redirect + 600, 64800 >$ sekund udávající dobu, po které se stane *secondary virtual forwarder* nedostupným. Výchozí hodnota je 14400 sekund (4 hodiny).

Příkaz je na Cisco IOS podporován od verze 12.2(14)S.

Pro nastavení váhy se používá tento příkaz:

```
Router(config-if)# glbp group weighting maximum [lower
lower][upper upper]
```

Kde ***maximum*** značí maximální hodnotu váhy v rozmezí od 1 do 254. Výchozí hodnotou je 100. ***Lower*** je volitelný parametr značící spodní hodnotu váhy v rozmezí od 1 do zadané maximální hodnoty. Výchozí hodnota je 1. ***Upper*** specifikuje horní možnou hodnotu váhy v rozmezí od spodní hodnoty do maximální hodnoty. Výchozí hodnota je maximální hodnota váhy. Příkaz je na Cisco IOS podporován od verze 12.2(14)S.

GLBP umožňuje také sledovat objekt na rozhraní a v případě výpadku tohoto objektu snížit hodnotu váhy rozhraní.

```
Router(config-if)# glbp group weightining track object-number
[decrement value]
```

Kde ***object-number*** značí číslo sledovaného objektu. Možné hodnoty jsou od 1 do 1000. Volitelný parametr `decrement` ***value*** značí, o kolik se sníží váha rozhraní při havárii objektu (respektive o kolik se zvýší váha v případě, že se objekt obnoví). Možné hodnoty jsou

od 1 do 254. Výchozí hodnota je 10. Příkaz je na Cisco IOS podporován od verze 12.2(14)S. Od verze 15.1(3)T je zvýšena maximální hodnota parametru *object-number* na 1000.

Metoda, která se použije pro *load balancing*, se nastavuje následujícím příkazem. Ve výchozím stavu je použita metoda *Round robin*.

```
Router(config-if)# glbp group load-balancing [round-robin |
weighted | host-dependent]
```

Příkaz je na Cisco IOS podporován od verze 12.2(14)S. Od verze 12.4(24)T2 je platná změna v případě zadání příkazu `no glbp` *group* `load-balancing`. Jestliže AVG není AVF, odpovídá na ARP dotazy MAC adresou prvního VF ve stavu *listen*.

Pro GLBP authentizaci se používá následující příkaz:

```
Router(config-if)# glbp group authentication {string | md5
{key-chain key-chain | key-string [0|7] string [timeout seconds
}}
```

Kde jednotlivé parametry znamenají následující:

- *string* znázorňuje autentizační řetězec. Počet znaků řetězce nesmí přesáhnout 255.
- `md5` nastaví autentizaci pomocí MD5.
- `key-chain` nastavuje autentizaci využívající vytvořenou klíčenku. Parametr *key-chain* se musí shodovat s názvem klíčenky.
- `key-string` nastaví tajné heslo pro MD5 autentizaci. Má dva volitelné parametry 0 a 7. Kde 0 udává, že klíč bude nezašifrovaný. 7 udává, že klíč bude šifrovaný. Parametr *string* může být až 100 znaků dlouhý a udává tajné heslo. Doporučuje se délka alespoň 16 znaků.

Příkaz je na Cisco IOS podporován od verze 12.2(14)S. Od verze 12.3(2)T je přidáno klíčové slovo `md5` spolu s parametry.

4.5.2 Kontrola konfigurace

Pro ověření správnosti nastavení protokolu se používá příkaz:

```
Router# show glbp [brief | group-number | interface]
```

Eventuálně příkaz:

```
Router# debug glbp [errors | events | packets | terse]
```

Příkaz `show glbp` je na Cisco IOS podporován od verze 12.2(14)S. Od verze 12.3(2)T jsou ve výstupu zobrazovány informace týkající se MD5 autentizace. Od verze 12.3(7)T jsou ve výstupu zobrazeny informace identifikující jednotlivé skupiny.

Příkaz `debug glbp` je na Cisco IOS podporován od verze 12.2(14)S.

Kapitola 5

Simulační prostředí OMNeT++

V této kapitole popisuji vývojové a simulační prostředí OMNeT++ [16] a framework ANSAINET [9] sloužící pro simulování chování jednotlivých prvků počítačových sítí.

5.1 OMNeT++

Jedná se o objektově orientované, modulární, diskrétní, simulační prostředí. Využívá hierarchický systém modulů, které spolu komunikují skrze zasílání zpráv přes brány jednotlivých modulů. Umístění a propojení jednotlivých modulů je definováno jazykem NED. Logika jednotlivých modulů je implementována v jazyce C++. Vývojové prostředí OMNeT++ IDE je postaveno na volně dostupném vývojovém prostředí Eclipse.

Samotná simulace je definována NED souborem s konfiguračním souborem (`omnetpp.ini`) a případně dalšími XML soubory sloužícími pro podrobnější konfiguraci jednotlivých zařízení v simulaci.

OMNeT++ je volně šiřitelný v podobě zdrojových kódů a přenositelný mezi systémy Windows, Linux a Mac OS.

5.2 ANSAINET

Projekt *Automated network simulation and analysis* vyvíjený na Fakultě informačních technologií Vysokého učení technického v Brně se zabývá rozšířením frameworku INET [17], který je součástí instalace OMNeT++. INET se zabývá simulací protokolů nad TCP/IP. Obsahuje implementaci protokolů síťové vrstvy TCP, UDP, IPv4, IPv6, OSPF, BGP a dalších. Dále protokoly linkové vrstvy jako Ethernet, PPP, IEEE 802.11 a podporu pro bezdrátové sítě.

Praktická část této práce vznikla pro ANSAINET postavený na verzi INET 3.2.1.

Kapitola 6

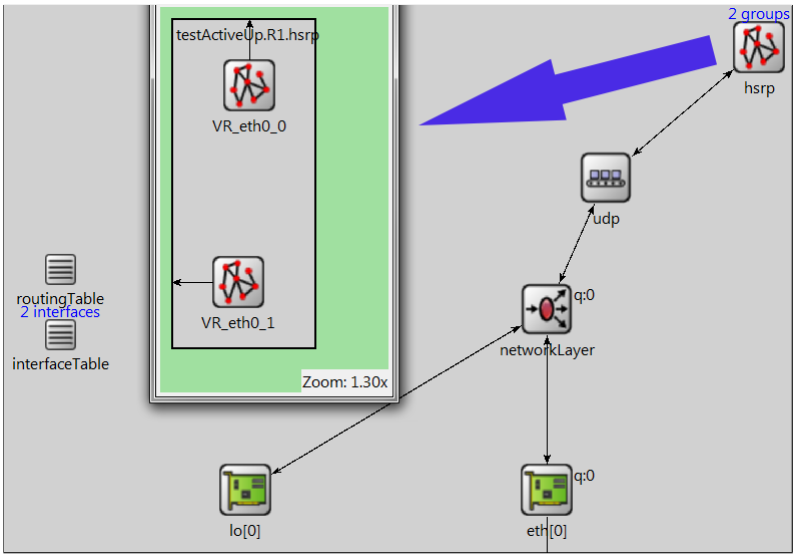
Návrh a implementace HSRP

V této kapitole se zabývám návrhem a implementací protokolu HSRP verze 1. Popisuji zde moduly nutné pro běh protokolu HSRP, formát zprávy a konfigurační soubor.

6.1 Modul HSRP

Jde o jednoduchý modul `HSRP`, který je součástí složeného modulu `ANSA_Router`. Modul komunikuje s modulem `ANSA_MultiNetworkLayer` přes modul `IUDP`. Má na starost načtení HSRP konfigurace a vytváření jednotlivých HSRP skupin na daných rozhraních. Jednotlivé skupiny jsou reprezentovány modulem `HSRPVirtualRouter`.

Za účelem zachování integrity modulů v ANSAINETU jsem si vytvořil vlastní modul `HSRPRouter` rozšiřující `ANSA_Router`, který pouze aktivuje HSRP modul. S tímto modulem směrovače pracuji v simulacích. Jeho vnitřní struktura je znázorněna na obrázku 6.1.



Obrázek 6.1: Ukázka umístění modulu `HSRP`, s dvěma dynamicky vytvořenými HSRP skupinami `VR_eth0_0` a `VR_eth0_1`.

`HSRP` modul je propojen s `UDP` modulem přes brány `udpIn` a `udpOut` na straně modulu `HSRP` a `appOut` a `appIn` na straně modulu `IUDP`. Je zde vytvořen UDP socket naslouchající

na portu číslo 1985. Zprávy přicházející na tento port, jsou odesílány směrem do patřičných `HSRPVirtualRouter` modulů podle čísla HSRP skupiny.

6.2 Modul HSRPVirtualRouter

Součástí tohoto modulu je implementace konečného automatu zajišťující chování HSRP protokolu, jehož popis je v kapitole 2.5. Jeden tento modul zastupuje jednu HSRP skupinu.

Modul odesílá zprávy *ARP Gratuitous* za pomocí modulu `ARP`, který jsem musel upravit. Dále jsem musel migrovat moduly `AnsaEtherMAC`, `AnsaEthernetInterface` a `AnsaEthernetInterfaceWithVF` ze staré verze ANSAINETU za účelem umožnění přidání více IP adres na jedno fyzické rozhraní.

Modul se také stará o reakce na změny stavu zařízení, kde je implementována reakce na odpojení (případně připojení) linky po přijetí signálu `NF_INTERFACE_STATE_CHANGED`.

6.3 Formát HSRP zprávy

HSRP zasílá zprávy jednoho formátu. V souboru `HSRPMessage.msg` jsem definoval strukturu zpráv následovně:

```
packet HSRPMessage {
    unsigned char version = 0;
    unsigned char op_code;
    unsigned char state;
    unsigned char hellotime;
    unsigned char holdtime;
    unsigned char priority;
    unsigned char group;
    IPv4Address address;
}
```

Kde jednotlivá pole paketu odpovídají specifikaci.

6.4 Konfigurace

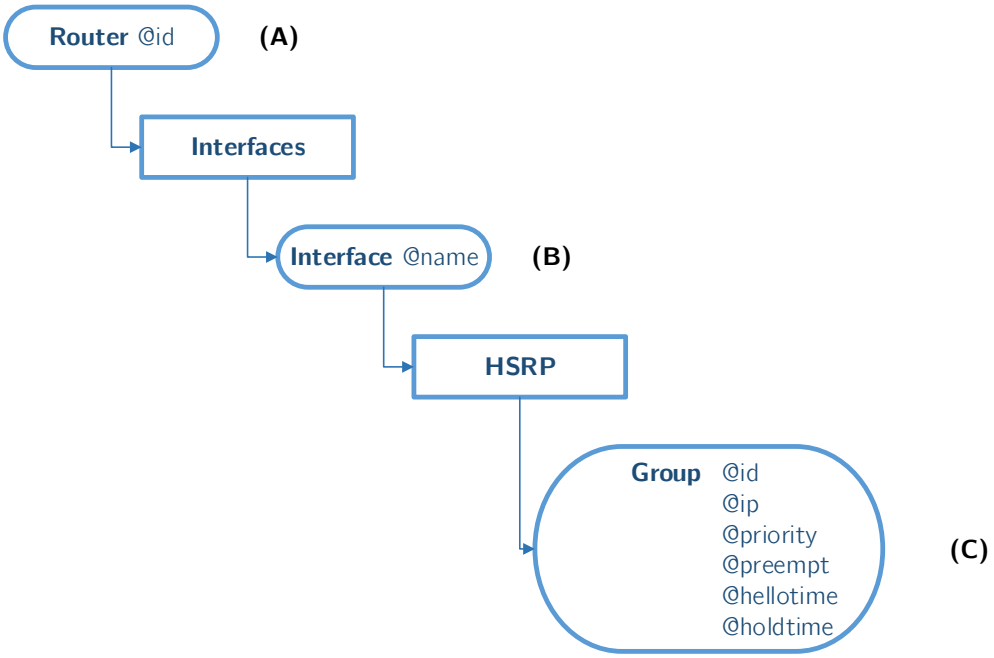
Konfigurační soubor je zapsán ve značkovacím jazyce XML. Tento formát zpracovávám v modulu `HSRP` v metodě `parseConfig(cXMLElement)`. Metodě je předán parametr `configData` odkazující na část souboru, která obsahuje konfiguraci jednotlivých rozhraní daného směrovače.

Strukturu konfiguračního souboru popisuje obrázek 6.2. Zde uvádím vysvětlení k obrázku:

(A) Identifikace sekce směrovače, pro který jsou určeny následující vnořené elementy. Parametr `id` určuje název směrovače.

(B) Identifikace konkrétního rozhraní. Parametr `name` udává název rozhraní.

(C) Specifikace přidávané skupiny se všemi implementovanými parametry. Možné hodnoty těchto parametrů odpovídají specifikaci v kapitole 2.6. Parametr `preempt` může mít hodnotu `true` nebo `false`.



Obrázek 6.2: Struktura konfigurace HSRP v XML.

Kapitola 7

Návrh a implementace GLBP

V této kapitole se zabývám návrhem a implementací protokolu GLBP s podporou pro IPv4. Popisuji zde moduly nutné pro běh protokolu GLBP, upravenou třídu zastupující *virtual forwarder*, formát zprávy GLBP, odlišnosti oproti Cisco implementaci a konfigurační soubor.

7.1 Modul GLBP

Jedná se o jednoduchý modul `GLBP`, který je součástí složeného modulu `ANSA_Router`. Modul komunikuje s modulem `ANSA_MultiNetworkLayer` přes modul `IUDP`. Modul se stará o načítání konfiguračních souborů týkajících se GLBP a dynamicky vytváří jednotlivé moduly `GLBPVirtualRouter`, které se starají o samotnou logiku GLBP protokolu.

Podobně jako u HSRP jsem si vytvořil vlastní modul `GLBPRouter` rozšiřující `ANSA_Router`, který pouze aktivuje GLBP modul. S tímto modulem směrovače pracuji v simulacích a jeho vnitřní struktura je znázorněna na obrázku 7.1.

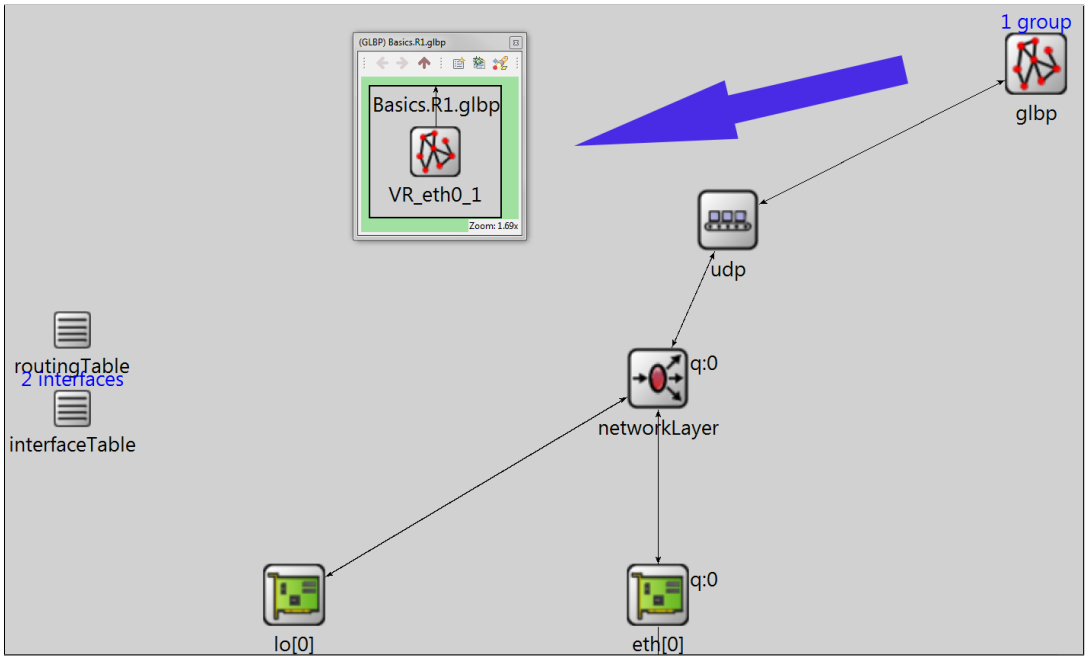
`GLBP` modul je propojen s `IUDP` modulem přes brány `udpIn` a `udpOut` na straně modulu `GLBP` a přes brány `appIn` a `appOut` na straně modulu `IUDP`. Je zde vytvořen UDP socket naslouchající na portu číslo 3222. Zprávy přicházející tímto portem jsou odesílány do příslušných `GLBPVirtualRouter` modulů podle čísla GLBP skupiny.

7.2 Modul GLBPVirtualRouter

Tento modul zastupuje jednu GLBP skupinu a zajišťuje funkcionalitu GLBP protokolu podle konečných automatů popsaných v kapitole 4.4. Je zde rovněž implementována metoda pro výběr virtuální MAC adresy *Round – robin*. Je zde i implementace reakce na změny stavu zařízení, kde je implementována reakce na odpojení (případně připojení) linky po přijetí signálu `NF_INTERFACE_STATE_CHANGED`.

Modul komunikuje s následujícími moduly, které jsem musel upravit:

- `ARP` - za účelem úpravy odpovědí na ARP dotazy klientů na výchozí bránu. Když `ARP` modulu přijde *ARP Request*, tak modul ověří, zda je směrovač ve stavu AVG a v kladném případě zašle signál `recvReqSignal`. `GLBPVirtualRouter` tento signál zpracuje a nastaví MAC adresu pomocí algoritmu *Round Robin*, která bude odeslána `ARP` modulem v odpovědi *ARP Reply*.



Obrázek 7.1: Ukázka umístění modulu `GLBP` s jednou dynamicky vytvořenou GLBP skupinou `VR_eth0_1`.

- `AnsaEtherMACFullDuplex` - modul linkové vrstvy jsem musel upravit za účelem nastavení virtuální MAC adresy pro ethernetový rámec v případě GLBP komunikace. Tento modul ověří, zda odesílaný paket patří protokolu GLBP a zda obsahuje informace týkající se AVF. Jestliže ano, tak nastaví virtuální MAC adresu obsaženou v *Request – response* TLV jako zdrojovou MAC adresu ethernetového rámce.

7.3 Třída zastupující VF

Třída `GLBPVirtualForwarder` je zděděná z třídy `VirtualForwarder` a jsou přidány pouze parametry nezbytné pro rozhodovací proces volby směrovačů *primary virtual forwarder* a *secondary virtual forwarder*. Také je zde příznak `AVG` určující, zda se jedná o AVG směrovač.

7.4 Formát GLBP zprávy

Formát jsem zvolil tak, aby věrohodně kopíroval skutečnou předlohu odchycenou programem Wireshark. V souboru `GLBPMessage.msg` je definovaná struktura zprávy následovně:

```
packet GLBPMessage {
    @customize(true);

    short version = 1;
    uint16_t group;
    MACAddress ownerId;

    TLVOptions TLV;
}
```

Kde proměnná `TLV` umožňuje přiřazovat libovolné množství `GLBPHello`, nebo `GLBPRequestResponse` struktur. To umožňuje dosáhnout požadované struktury GLBP paketu. V příloze E.2 uvádím příklad zobrazení GLBP paketu v prostředí OMNeT++. `GLBPHello` a `GLBPRequestResponse` struktury jsou definovány následovně:

```
class GLBPHello extends GLBPOption
{
    type = HELLO;
    length = GLBP_HELLO_BYTES;

    short vgState;
    short priority;
    uint32_t helloint;
    uint32_t holdint;
    uint16_t redirect;
    uint16_t timeout;
    short addressType = IPv4;
    short addresLength = 4;
    IPv4Address address;
}

class GLBPRequestResponse extends GLBPOption{
    type = REQRESP;
    length = GLBP_REQRESP_BYTES;

    short forwarder;
    short vfState;
    short priority;
    short weight;
    MACAddress macAddress;
}
```

7.5 Odlišnosti oproti Cisco implementaci

Během zkoumání protokolu GLBP jsem narazil na odlišnosti v Cisco implementaci protokolu na mnou testovaných verzích Cisco IOS 12.4(16) a Cisco IOS 15.2(4)S5. Kde první zmíněná odpovídala Cisco specifikaci dostupné online [2], ale obsahovala bohužel méně kvalitní ladicí výpisy, se kterými se mi nepodařilo sestavit stavový automat. Novější verze obsahovala kvalitní ladicí výpisy, ale obsahovala důležitou implementační odlišnost v rozporu s Cisco specifikací. Směrovače ve stavu *listen* odesílaly *Hello* zprávy, a tak se směrovač s nejvyšší prioritou dostal dříve do stavu *active*.

Má implementace je tedy založena na informacích v Cisco specifikaci a částečně na vypozorovaném chování v implementaci Cisco IOS 15.2(4)S5, na které je provedeno testování.

7.6 Konfigurace

Konfigurační soubor je zapsán ve značkovacím jazyce XML. Tento formát je zpracován v modulu `GLBP` v metodě `parseConfig(cXMLElement)`. Metodě je předán parametr `configData`

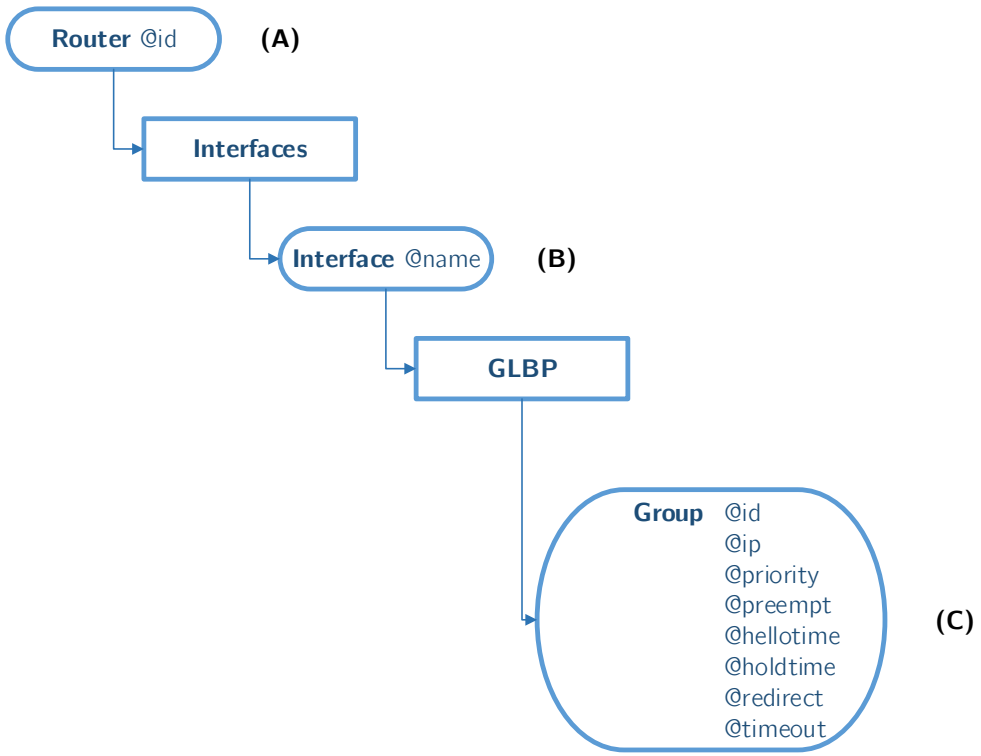
odkazující na část souboru, která obsahuje konfiguraci jednotlivých rozhraní daného směrovače.

Strukturu konfiguračního souboru popisuje obrázek 7.2. Zde uvádím vysvětlení k obrázku:

(A) Identifikace sekce směrovače, pro který jsou určeny následující vnořené elementy. Parametr `id` určuje název směrovače.

(B) Výběr konkrétního rozhraní. Parametr `name` udává název konkrétního rozhraní.

(C) Specifikace skupiny se všemi implementovanými parametry. Jednotlivé parametry odpovídají specifikaci v kapitole 4.5. Parametr `preempt` může mít hodnotu `true` nebo `false`.

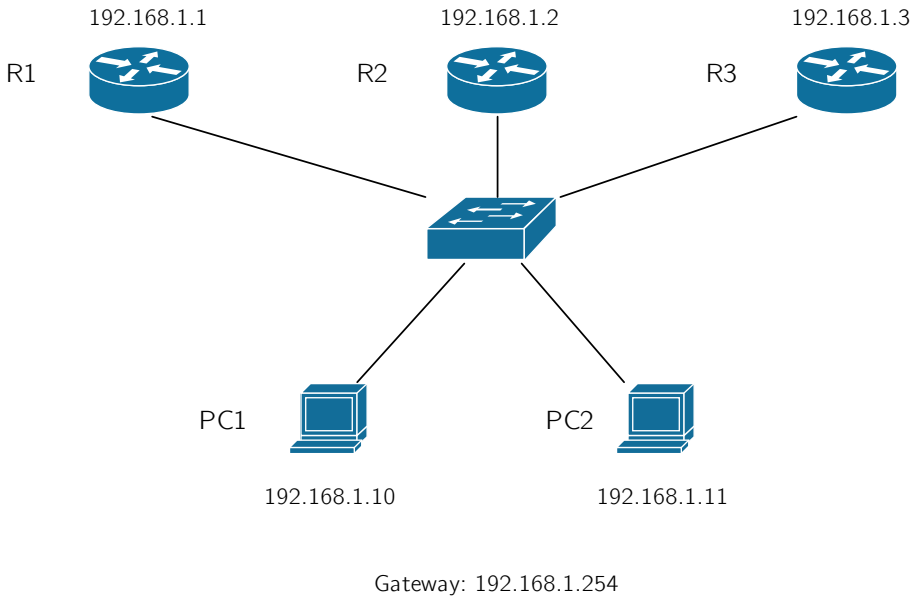


Obrázek 7.2: Struktura konfigurace GLBP v XML.

Kapitola 8

Porovnání simulace s reálným prostředím

Kapitola se zabývá porovnáváním průběhu simulace v OMNeT++ s reálnou sítí. Za účelem dostatečné názornosti a zachování přehlednosti jsem zvolil modelovou situaci se třemi směrovači zobrazenou na obrázku 8.1. Všechny směrovače patří v obou případech (u HSRP i GLBP) do jedné skupiny číslo 0 s IP adresou skupiny 192.168.1.254. Koncové stanice `PC1` a `PC2` spadají do stejné sítě 192.168.1.0/24 a mají jako výchozí bránu nastavenou adresu virtuální skupiny.



Obrázek 8.1: Topologie pro testování se znázorněním IP adres připojených rozhraní a s výchozí bránou koncových zařízení.

Jako počáteční čas v reálném prostředí pokládám spuštění nakonfigurovaného rozhraní

na směrovači. Testované směrovače jsou zařízení s operačním systémem `Cisco IOS Software` odpovídající verzi `7200 Software (C7200-ADVIPSERVICESK9-M), Version 15.2-(4)S5, RELEASE SOFTWARE (fc1)`.

Na Cisco zařízeních jsou jednotlivé časovače opožďovány o náhodnou hodnotu, která může být u HSRP až o 20 % větší, než zadaná hodnota [6]. U GLBP Cisco velikost této odchylky ve specifikacích neuvádí. Tímto mechanismem se zabraňuje nadměrnému zatížení sítě v jeden okamžik v podobě množství HSRP, či GLBP zpráv. Toto chování není v simulátoru implementováno, a tak se časy a pořadí některých událostí mohou lišit.

Na této topologii byly provedeny následující testovací scénáře zobrazené v tabulkách 8.1 a 8.2 s umístěním simulace v OMNeT++. Všechny scénáře se nacházejí ve složce `examples/`.

#	Název simulace	Kapitola	Simulace v OMNeT++
1	Volba virtuální brány	8.1	`ansa/hsrp/testActiveChoose`
2	Výpadek rozhraní	8.2	`ansa/hsrp/testActiveDown`
3	Obnovení rozhraní po výpadku	8.3	`ansa/hsrp/testActiveUp`

Tabulka 8.1: Seznam provedených testů na HSRP.

#	Název simulace	Kapitola	Simulace v OMNeT++
1	Volba virtuální brány	8.4	`ansa/glbp/testVgChoose`
2	ARP odpověď	8.5	`ansa/glbp/testArp`
3	Výpadek rozhraní	8.6	`ansa/glbp/testVgDown`
4	Obnovení rozhraní po výpadku	8.7	`ansa/glbp/testVgUp`

Tabulka 8.2: Seznam provedených testů na GLBP.

8.1 Volba virtuální brány HSRP

V testu porovnávám změny stavů jednotlivých směrovačů od počátku simulace. Výpisy z logu modulu ze simulace jsou porovnány s ladícími výpisy na Cisco zařízení zobrazenými pomocí příkazu `debug standby`. Při porovnávání se zaměřuji zejména na správné pořadí přechodů a čas jednotlivých přechodů.

Zobrazuji zde pouze tabulky s časy přechodů. Tabulka 8.3 zobrazuje porovnání časů přechodů směrovače `R1`, tabulka 8.4 zobrazuje porovnání časů přechodů směrovače `R2` a tabulka 8.5 zobrazuje porovnání časů přechodů směrovače `R3`. U Cisco směrovačů jsem časy převedl do času 0 pro lepší názornost. Všechny časy jsem zaokrouhlil. Neupravené časy je možné si prohlédnout v příloze F.2.

8.1.1 Zhodnocení

U směrovače `R1` (viz tabulka 8.3) je v simulaci větší počet zakolísání mezi stavy. To je způsobeno zpožďováním časovačů, které je popsané v úvodu této kapitoly. Jinak je v porovnání časů z tabulek u směrovačů `R1`, `R2` a `R3` vidět odchylky do 4 s. Chování simulace tedy odpovídá Cisco směrovačům.

Přechod	**Cisco t [s]**	**Simulace t [s]**
Init → Listen	0	0
Listen → Speak	11	10
Speak → Listen	14	10
Listen → Speak	27	20
Speak → Listen	30	20
Listen → Speak	-	30
Speak → Listen	-	30

Tabulka 8.3: Porovnání přechodů na `R1`.

Přechod	**Cisco t [s]**	**Simulace t [s]**
Init → Listen	0	0
Listen → Speak	11	10
Speak → Listen	12	13
Listen → Speak	32	30
Speak → Standby	43	40

Tabulka 8.4: Porovnání přechodů na `R2`.

Přechod	**Cisco t [s]**	**Simulace t [s]**
Init → Listen	0	0
Listen → Speak	10	10
Speak → Standby	20	20
Standby → Active	23	20

Tabulka 8.5: Porovnání přechodů na `R3`.

8.2 Výpadek rozhraní u HSRP

V tomto testu se zaměřuji na správnou změnu stavů u směrovačů `R1` a `R2` při vypojení linky vedoucí k směrovači `R3`, který je ve stavu *active*.

Linka je vypojena v momentě, kdy jsou ustáleny stavy a je tedy znám *active* i *standby* router. Na Cisco zařízení je sledováno rozhraní vedoucí k směrovači `R2`. V simulaci jsou sledovány pakety odeslané jednotlivými směrovači.

Průběh komunikace na Cisco zařízení je zobrazen na obrázku 8.2. Průběh komunikace v simulaci je na obrázku 8.3, kde byly irelevantní informace ořezány z důvodu zmenšení velikosti výpisu. Ve výřezu je zvýrazněn poslední paket od směrovače `R3`. Změna stavu směrovače `R2` do *active* a odeslání *ARP Gratuitous*. Poslední zvýrazněný paket je přechod stavu `R1` do *standby*.

Na obrázcích 8.4 je zobrazeno porovnání paketů ze směrovače `R2`. Ostatní HSRP pakety se liší pouze hodnotou stavu, a tak není třeba je porovnávat.

No.	Time	Source	Destination	Protocol	Length	Info
52	50.653905000	192.168.1.3	224.0.0.2	HSRP	62	Hello (state Active)
53	51.426932000	192.168.1.2	224.0.0.2	HSRP	62	Hello (state Standby)
54	53.521007000	192.168.1.2	224.0.0.2	HSRP	60	Advertise (state Passive)
55	54.298035000	192.168.1.2	224.0.0.2	HSRP	62	Hello (state Standby)
57	56.532114000	192.168.1.1	224.0.0.2	HSRP	60	Advertise (state Passive)
58	57.148136000	192.168.1.2	224.0.0.2	HSRP	62	Hello (state Standby)
60	59.791234000	192.168.1.2	224.0.0.2	HSRP	62	Hello (state Standby)
61	60.749270000	192.168.1.2	224.0.0.2	HSRP	60	Advertise (state Active)
62	60.751270000	192.168.1.2	224.0.0.2	HSRP	62	Hello (state Active)
63	60.771271000	All-HSRP-routers_00	Broadcast	ARP	60	Gratuitous ARP for 192.168.1.254 (Reply)
64	60.771271000	All-HSRP-routers_00	STP-UplinkFast	ARP	60	Gratuitous ARP for 192.168.1.254 (Reply)
66	61.717307000	192.168.1.1	224.0.0.2	HSRP	62	Hello (state Speak)
67	63.579377000	192.168.1.2	224.0.0.2	HSRP	62	Hello (state Active)
68	63.589378000	192.168.1.1	224.0.0.2	HSRP	60	Advertise (state Passive)
69	63.756384000	All-HSRP-routers_00	Broadcast	ARP	60	Gratuitous ARP for 192.168.1.254 (Reply)
70	64.227402000	192.168.1.1	224.0.0.2	HSRP	62	Hello (state Speak)
71	66.450484000	192.168.1.2	224.0.0.2	HSRP	62	Hello (state Active)
72	67.201513000	192.168.1.1	224.0.0.2	HSRP	62	Hello (state Speak)
74	69.138580000	192.168.1.2	224.0.0.2	HSRP	62	Hello (state Active)
75	69.831606000	192.168.1.1	224.0.0.2	HSRP	62	Hello (state Speak)
76	70.162617000	192.168.1.1	224.0.0.2	HSRP	62	Hello (state Standby)
77	71.812681000	192.168.1.2	224.0.0.2	HSRP	62	Hello (state Active)

Obrázek 8.2: Výřez komunikace HSRP zachycený ve Wiresharku.

Event#	Time	Src/Dest	Name
#2421	41.000000959999	R3 --> SW1	HSRPHello (Active)
#2473	43.000020259997	R2 --> SW1	HSRPHello (Standby)
#2527	46.000020259997	R2 --> SW1	HSRPHello (Standby)
#2569	49.000020259997	R2 --> SW1	HSRPHello (Standby)
#2625	51.000013539998	R2 --> SW1	arpGrt
#2626	51.000013539998	R1 --> SW1	HSRPHello (Speak)
#2643	51.000020259997	R2 --> SW1	HSRPHello (Active)
#2739	54.000013539998	R1 --> SW1	HSRPHello (Speak)
#2740	54.000013539998	R2 --> SW1	HSRPHello (Active)
#2823	57.000013539998	R1 --> SW1	HSRPHello (Speak)
#2824	57.000013539998	R2 --> SW1	HSRPHello (Active)
#2907	60.000013539998	R1 --> SW1	HSRPHello (Speak)
#2908	60.000013539998	R2 --> SW1	HSRPHello (Active)
#2981	61.000013539998	R1 --> SW1	HSRPHello (Standby)
#3023	63.000013539998	R2 --> SW1	HSRPHello (Active)

Obrázek 8.3: Výřez komunikace HSRP zachycený v OMNeT++.

```
Cisco Hot Standby Router Protocol
  Version: 0
  Op Code: Hello (0)
  State: Active (16)
  Hellotime: Default (3)
  Holdtime: Default (10)
  Priority: 100
  Group: 0
  Reserved: 0
  Authentication Data: Default (cisco)
  Virtual IP Address: 192.168.1.254 (192.168.1.254)
```

(a) Wireshark

```
encapsulatedPacket = (inet::HSRPMessage) HSRPHello (Active) (cPacket)
controlInfo = NULL (cObject)
encapsulatedPacket = NULL (cPacket)
version = 0 [...] (unsigned char)
op_code = 0 [...] (unsigned char)
state = 16 [...] (unsigned char)
hellotime = 3 [...] (unsigned char)
holdtime = 10 [...] (unsigned char)
priority = 100 [...] (unsigned char)
group = 0 [...] (unsigned char)
address = 192.168.1.254 (IPv4Address)
```

(b) OMNeT++

Obrázek 8.4: Porovnání prvního *active* paketu od směrovače R2.

8.2.1 Zhodnocení

Při porovnání výstupů z výřezu komunikace na obrázcích 8.2 a 8.3 je vidět, že směrovač R2 odeslal v případě Cisco směrovačů o jednu *Hello* zprávu ve stavu *standby* více, než v případě simulace. To je způsobeno opožďováním časovačů v případě Cisco směrovačů.

Dále jsou vidět ve výstupu z Wiresharku zprávy typu *Advertise*, které nejsou uvedeny ve specifikaci, a které jsem tedy neimplementoval.

Porovnání zaokrouhlených časů přechodů ukazuji v následující tabulce 8.6. Odlišnost časových rozestupů u Cisco implementace a simulace je nižší než jedna sekunda.

Událost	**Cisco t [s]**	**Simulace t [s]**
R3: odeslal Active Hello	51	41
R2: Standby → Active	61	51
R2: odeslal ARP Gratuitous	61	51
R1: Listen → Speak	61	51
R1: Speak → Standby	70	61

Tabulka 8.6: Porovnání časů jednotlivých událostí HSRP.

8.3 Obnovení rozhraní po výpadku u HSRP

Test navazuje na předchozí testování. Zaměřuje se na sledování přechodů mezi stavy na všech směrovačích po obnovení linky k směrovači `R3`. Linka je obnovena v čase $t = 63\,s$, kdy už jsou stavy po výpadku linky k `R3` ustáleny. Na Cisco je linka obnovena v čase $t = 40\,s$. V tomto testu jsou použity ladící výpisy pomocí příkazu `debug standby` na Cisco zařízeních a jsou porovnány s výpisy z logu modulu ze simulace.

V tabulce 8.7 jsou zobrazeny časy přechodů směrovačů `R1` a `R3`. Směrovač `R2` zůstane během této komunikace ve stavu *active*. Výstupy ze simulace a z pomocných výpisů na Cisco zařízení jsou zobrazeny v příloze F.1.

`R1` **přechody**	**Cisco t [s]**	**Simulace t [s]**
Standby → Listen	43	64
Listen → Speak	-	74
Speak → Listen	-	74

`R3` **přechody**	**Cisco t [s]**	**Simulace t [s]**
Init → Listen	41	63
Listen → Speak	43	64
Speak → Standby	55	74

Tabulka 8.7: Porovnání přechodů u směrovačů `R1` a `R3`.

8.3.1 Zhodnocení

U směrovače `R1` je vidět v tabulce 8.7 zakolísání stavů v případě simulace. To je způsobeno tím, že směrovači vypršel *standby timer* a v ten samý okamžik obdržel *Hello* zprávu od směrovače `R3` ve stavu *standby*. Tato situace nenastala na Cisco směrovači z důvodu opožďování časovačů.

Ostatní chování simulace odpovídá až na malé časové odchylky způsobené opožďováním časovačů v implementaci Cisco.

8.4 Volba virtuální brány GLBP

Test se zaměřuje na sledování přechodů mezi jednotlivými stavy a na časový okamžik přechodu. Je porovnáno správné ustanovení virtuální brány a zároveň i ustanovení jednotlivých VF. Pro tento test jsou použity ladící výpisy týkající se přechodů stavů na Cisco zařízení pomocí příkazu `debug glbp` a jsou porovnány s výpisy z logu modulu ze simulace.

Na obrázku 8.5 uvádím výsledky z Cisco zařízení, na obrázku 8.6 uvádím výsledky ze simulace. Vzhledem k velikosti výpisu jsem zde uvedl pouze výpis směrovače, který skončil ve stavu AVG po ustálení konfigurace. Celý výpis se nachází v příloze F.3.

```
R3
18:22:43.483: GLBP: Fa0/0 Interface up
18:22:43.487: GLBP: Fa0/0 0 Init -> Listen
18:22:53.407: GLBP: Fa0/0 0 Listen -> Speak
18:22:53.831: GLBP: Fa0/0 0 Speak -> Active
18:22:53.839: GLBP: Fa0/0 0.1 Disabled -> Listen
18:23:00.355: GLBP: Fa0/0 0.2 Disabled -> Listen
18:23:00.723: GLBP: Fa0/0 0.3 Disabled -> Listen
18:23:04.767: GLBP: Fa0/0 0.1 Listen -> Active
```

Obrázek 8.5: Výřez z ladících výstupů na Cisco směrovači R3.

```
R3
t=0 Grp 0 Init -> Listen
t=10 Grp 0 Listen -> Speak
t=20 Grp 0 Speak -> Active
t=20 Fwd 1 Grp 0 Disabled -> Listen
t=30 Fwd 1 Grp 0 Listen -> Active
t=30.000096759992 Fwd 2 Grp 0 Disabled -> Listen
t=30.000105239991 Fwd 3 Grp 0 Disabled -> Listen
```

Obrázek 8.6: Výřez z konzolových výstupů ze simulace na R3 směrovači.

8.4.1 Zhodnocení

V tabulce 8.8 jsou pro názornost časy přechodů při volbě VG převedeny do nulového času a zaokrouhleny na sekundy. V simulaci je patrné opoždění oproti implementaci Cisco. Je to zapříčiněno odlišností Cisco implementace oproti specifikaci, kterou zmiňuji v kapitole 7.5. Od této odlišnosti se odvíjí i opoždění při přidělování rolí VF ostatním směrovačům. Kdy se jednotlivé Cisco směrovače dozvědí dříve o existenci AVG, a tak jsou i jednotliví VF vytvořeni na AVG a přiřazení jednotlivým PVF a SVF dříve.

Přechod	Cisco t[s]	Simulace t[s]
Init → Listen	0	0
Listen → Speak	10	10
Speak → Active	10	20

Tabulka 8.8: Zaokrouhlené časy změn přechodů na směrovači R3 při volbě AVG.

8.5 ARP odpověď u GLBP

Tento test se zaměřuje na správné přidělování výchozí brány jednotlivým koncovým stanicím `PC1` a `PC2` v ARP odpovědích. Na stanici `PC1` je spuštěn příkaz `ping 192.168.1.254` v čase $t = 35\,s$, kdy už jsou stavy bezpečně ustanoveny. Na stanici `PC2` je stejný příkaz spuštěn v čase $t = 36\,s$.

Výstupy ze simulace jsou porovnány s obsahy ARP paketů mezi jednotlivými Cisco směrovači zachycenými programem Wireshark. Jelikož sleduji zejména ARP odpovědi, tak je Wireshark spuštěn na lince vedoucí od přepínače k směrovači R3, který je ve stavu *active*.

Na obrázku 8.7 uvádím pakety zachycené v reálné síti. Na obrázku 8.8 jsou zvýrazněny sledované ARP pakety v simulátoru OMNeT++.

No.	Time	Source	Destination	Protocol	Length	Info
90	78.156817000	ca:04:21:b0:00:00	Broadcast	ARP	60	Who has 192.168.1.254? Tell 192.168.1.10
91	78.208817000	ca:03:32:78:00:00	ca:04:21:b0:00:00	ARP	60	192.168.1.254 is at 00:07:b4:00:00:01
112	88.287836000	ca:05:01:78:00:00	Broadcast	ARP	60	Who has 192.168.1.254? Tell 192.168.1.11
113	88.297837000	ca:03:32:78:00:00	ca:05:01:78:00:00	ARP	60	192.168.1.254 is at 00:07:b4:00:00:02

Obrázek 8.7: Pakety *ARP request* a *ARP reply* zachycené ve Wiresharku.

Event#	Time	Src/Dest	Name	Info
#3599	35	PC1 --> SW1	arpREQ	ARP req: 192.168.1.254=? (s=192.168.1.10(0A-AA-00-00-00-04))
#3602	35.000000959999	R3 --> SW1	GLBPHello,Req/Resp	inet::GLBPMessage:60 bytes UDP: 192.168.1.3.3222 > 224.0.
#3607	35.00000581	SW1 --> R1	arpREQ	ARP req: 192.168.1.254=? (s=192.168.1.10(0A-AA-00-00-00-04))
#3608	35.00000581	SW1 --> R2	arpREQ	ARP req: 192.168.1.254=? (s=192.168.1.10(0A-AA-00-00-00-04))
#3609	35.00000581	SW1 --> R3	arpREQ	ARP req: 192.168.1.254=? (s=192.168.1.10(0A-AA-00-00-00-04))
#3610	35.00000581	SW1 --> PC2	arpREQ	ARP req: 192.168.1.254=? (s=192.168.1.10(0A-AA-00-00-00-04))
#3618	35.000010129999	SW1 --> PC1	GLBPHello,Req/Resp	inet::GLBPMessage:60 bytes UDP: 192.168.1.3.3222 > 224.0.
#3648	35.000012529999	SW1 --> R1	GLBPHello,Req/Resp	inet::GLBPMessage:60 bytes UDP: 192.168.1.3.3222 > 224.0.
#3649	35.000012529999	SW1 --> R2	GLBPHello,Req/Resp	inet::GLBPMessage:60 bytes UDP: 192.168.1.3.3222 > 224.0.
#3651	35.000012529999	SW1 --> PC2	GLBPHello,Req/Resp	inet::GLBPMessage:60 bytes UDP: 192.168.1.3.3222 > 224.0.
#3652	35.000012579999	R3 --> SW1	arpREPLY	ARP reply: 192.168.1.254=00-07-B4-00-00-01 (d=192.168.1.10(0
#3664	35.000018019997	R1 --> SW1	GLBPHello,Req/Resp	inet::GLBPMessage:60 bytes UDP: 192.168.1.1.3222 > 224.0.
#3676	35.000020209998	SW1 --> PC1	arpREPLY	ARP reply: 192.168.1.254=00-07-B4-00-00-01 (d=192.168.1.10(0
#3711	35.000026019998	PC1 --> SW1	ping0	PING req 192.168.1.10 to 192.168.1.254 (60 bytes) id=5853 se
#3717	35.000027189997	SW1 --> R2	GLBPHello,Req/Resp	inet::GLBPMessage:60 bytes UDP: 192.168.1.1.3222 > 224.0.
#3718	35.000027189997	SW1 --> R3	GLBPHello,Req/Resp	inet::GLBPMessage:60 bytes UDP: 192.168.1.1.3222 > 224.0.
#3719	35.000027189997	SW1 --> PC1	GLBPHello,Req/Resp	inet::GLBPMessage:60 bytes UDP: 192.168.1.1.3222 > 224.0.
#3720	35.000027189997	SW1 --> PC2	GLBPHello,Req/Resp	inet::GLBPMessage:60 bytes UDP: 192.168.1.1.3222 > 224.0.
#3756	35.000037269996	SW1 --> R3	ping0	PING req 192.168.1.10 to 192.168.1.254 (60 bytes) id=5919 se
#3770	35.000047079995	R3 --> SW1	ping0-reply	PING reply 192.168.1.254 to 192.168.1.10 (60 bytes) id=5939
#3775	35.000055929995	SW1 --> PC1	ping0-reply	PING reply 192.168.1.254 to 192.168.1.10 (60 bytes) id=5946
#3808	36	PC1 --> SW1	ping1	PING req 192.168.1.10 to 192.168.1.254 (60 bytes) id=5997 se
#3809	36	PC2 --> SW1	arpREQ	ARP req: 192.168.1.254=? (s=192.168.1.11(0A-AA-00-00-00-05))
#3813	36.000000959999	R2 --> SW1	GLBPHello,Req/Resp	inet::GLBPMessage:60 bytes UDP: 192.168.1.2.3222 > 224.0.
#3818	36.00000581	SW1 --> R1	arpREQ	ARP req: 192.168.1.254=? (s=192.168.1.11(0A-AA-00-00-00-05))
#3819	36.00000581	SW1 --> R2	arpREQ	ARP req: 192.168.1.254=? (s=192.168.1.11(0A-AA-00-00-00-05))
#3820	36.00000581	SW1 --> R3	arpREQ	ARP req: 192.168.1.254=? (s=192.168.1.11(0A-AA-00-00-00-05))
#3821	36.00000581	SW1 --> PC1	arpREQ	ARP req: 192.168.1.254=? (s=192.168.1.11(0A-AA-00-00-00-05))
#3835	36.000010129999	SW1 --> PC2	GLBPHello,Req/Resp	inet::GLBPMessage:60 bytes UDP: 192.168.1.2.3222 > 224.0.
#3864	36.000012529999	SW1 --> R1	GLBPHello,Req/Resp	inet::GLBPMessage:60 bytes UDP: 192.168.1.2.3222 > 224.0.
#3866	36.000012529999	SW1 --> R3	ping1	PING req 192.168.1.10 to 192.168.1.254 (60 bytes) id=6083 se
#3867	36.000012529999	SW1 --> PC1	GLBPHello,Req/Resp	inet::GLBPMessage:60 bytes UDP: 192.168.1.2.3222 > 224.0.
#3868	36.000012579999	R3 --> SW1	arpREPLY	ARP reply: 192.168.1.254=00-07-B4-00-00-02 (d=192.168.1.11(0

Obrázek 8.8: Průběh komunikace v OMNeT++ se zvýrazněnými sledovanými ARP pakety.

8.5.1 Zhodnocení

Na uvedených výstupech je znázorněna komunikace `PC1` a `PC2` s AVG. V reálné síti i v simulaci zašle AVG na první ARP dotaz odpověď obsahující vMAC patřící VF 1. Na druhý ARP dotaz zašle ARP odpověď obsahující vMAC od VF 2. Chování v simulaci tedy odpovídá chování na Cisco zařízení.

8.6 Výpadek rozhraní u GLBP

Test je zaměřen na porovnání chování v případě výpadku linky vedoucí k AVG. Sleduji, zda *standby* směrovač přejde správně do stavu *active* a zároveň převezme roli VF za odstaveného AVG. V čase $t = 35\,s$ je v simulaci vypnuta linka vedoucí k AVG.

Výpisy ze simulace jsou porovnány s výpisem odchycených paketů v programu Wireshark. Wireshark je spuštěn na lince mezi přepínačem a směrovačem `R2`, který je v čase výpadku linky ve stavu *standby*. Zaměřuji se zde zejména na správné pořadí jednotlivých zpráv.

8.6.1 Sledování převzetí role VF

Na obrázcích 8.9 a 8.10 jsou zvýrazněny porovnávané pakety.

No.	Time	Source	Destination	Protocol	Length	Info
70	41.282064000	192.168.1.2	224.0.0.102	GLBP	102	G: 0, Hello, IPv4, Request/Response?
71	43.459067000	192.168.1.2	224.0.0.102	GLBP	102	G: 0, Hello, IPv4, Request/Response?
72	43.889068000	192.168.1.1	224.0.0.102	GLBP	102	G: 0, Hello, IPv4, Request/Response?
73	44.049068000	192.168.1.1	224.0.0.102	GLBP	74	G: 0, Request/Response?
74	44.079068000	192.168.1.2	224.0.0.102	GLBP	74	G: 0, Request/Response?
75	44.189068000	192.168.1.1	224.0.0.102	GLBP	102	G: 0, Hello, IPv4, Request/Response?
77	45.976071000	192.168.1.2	224.0.0.102	GLBP	122	G: 0, Hello, IPv4, Request/Response?, Request/Response?
78	47.103073000	192.168.1.1	224.0.0.102	GLBP	102	G: 0, Hello, IPv4, Request/Response?
79	48.607076000	192.168.1.2	224.0.0.102	GLBP	122	G: 0, Hello, IPv4, Request/Response?, Request/Response?
80	49.637077000	192.168.1.1	224.0.0.102	GLBP	102	G: 0, Hello, IPv4, Request/Response?
81	51.089079000	192.168.1.2	224.0.0.102	GLBP	122	G: 0, Hello, IPv4, Request/Response?, Request/Response?
83	52.534083000	192.168.1.1	224.0.0.102	GLBP	102	G: 0, Hello, IPv4, Request/Response?
84	53.894085000	192.168.1.2	224.0.0.102	GLBP	122	G: 0, Hello, IPv4, Request/Response?, Request/Response?
85	54.204086000	192.168.1.1	224.0.0.102	GLBP	102	G: 0, Hello, IPv4, Request/Response?

Obrázek 8.9: Průběh komunikace zachycené ve Wiresharku se zvýrazněnými sledovanými GLBP pakety.

Event#	Time	Src/Dest	Name
#3877	42.000020259998	R1 --> SW1	GLBPRequest/Response
#3878	42.000020259998	R2 --> SW1	GLBPRequest/Response
#3889	42.000027829998	SW1 --> PC1	GLBPRequest/Response
#3890	42.000027829998	SW1 --> PC2	GLBPRequest/Response
#3895	42.000028739997	R1 --> SW1	GLBPHello,Req/Resp
#3896	42.000028739997	R2 --> SW1	GLBPHello,Req/Resp
#3897	42.000028789997	SW1 --> R1	GLBPRequest/Response
#3898	42.000028789997	SW1 --> R2	GLBPRequest/Response
#3913	42.000036309997	SW1 --> PC1	GLBPRequest/Response
#3914	42.000036309997	SW1 --> PC2	GLBPRequest/Response
#3965	42.000040419996	R1 --> SW1	GLBPHello,Req/Resp
#3966	42.000040419996	R2 --> SW1	GLBPHello,Req/Resp

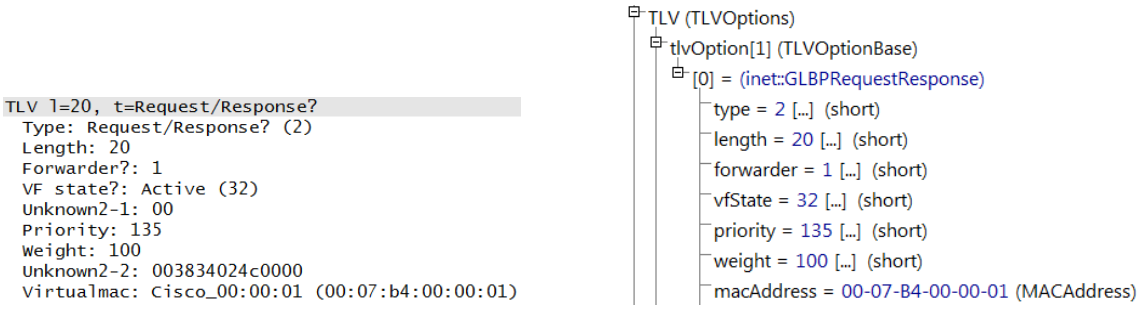
Obrázek 8.10: Průběh komunikace v OMNeT++ se zvýrazněnými sledovanými GLBP pakety.

První zvýrazněné pakety, označené jako `Request/Response?` ve Wiresharku (respektive `GLBPRequest/Response` v OMNeT++), jsou odeslané zbylými routery po vypršení *active timeru* nefunkčního VF. Porovnání jejich obsahu je na obrázcích 8.11. Oba směrovače zasílají stejné GLBP pakety, proto zobrazují pouze paket zaslaný směrovačem `R1`.

Dále je v simulaci modře zvýrazněno vypršení *hello timeru*, které na Cisco směrovačích nenastalo.

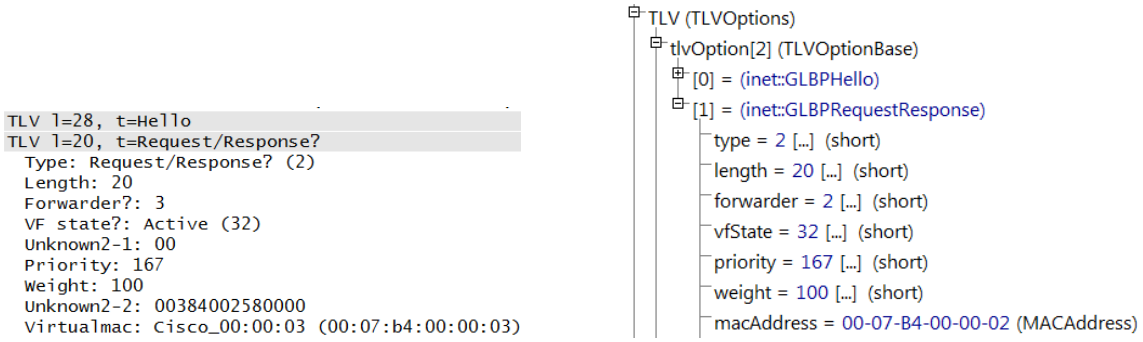
Další paket je zpráva od `R1`, což je směrovač s nižší prioritou, který se vzdal VF 1 a přenechal stav *active* pro VF 1 směrovači `R2`. Porovnání této zprávy je na obrázku 8.12, kde je zobrazený paket nesoucí informace o stavu VG (`Hello` TLV) a o VF jehož je `R1` PVF.

Poslední paket je od směrovače `R2`, který funguje jako *active* pro VF 1 a svou skupinu VF. Jeho porovnání je na obrázku 8.13. Paket nese informace o VG stavu v `Hello` TLV a o dvou VF v jednotlivých `Request/Response` TLV.



(a) Wireshark (b) OMNeT++

Obrázek 8.11: Porovnání prvního zvýrazněného paketu od směrovače `R1`.

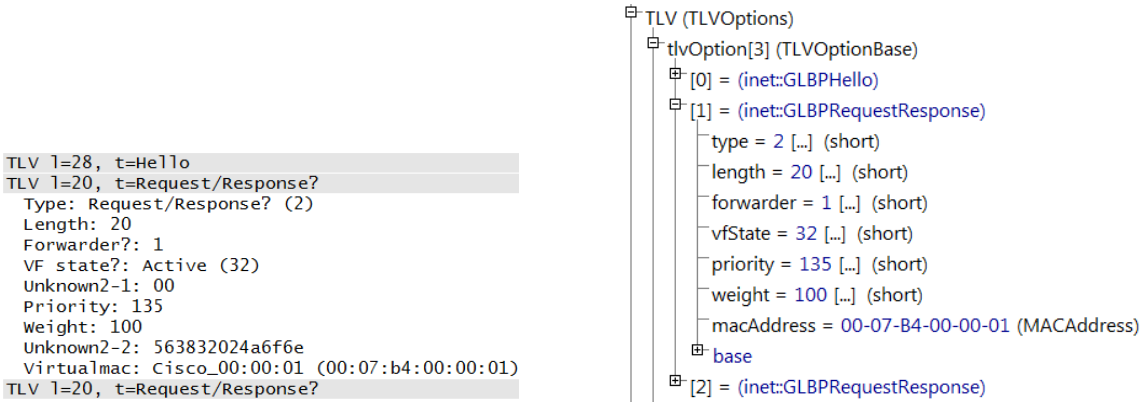


(a) Wireshark (b) OMNeT++

Obrázek 8.12: Porovnání paketu od směrovače `R1`.

8.6.2 Sledování převzetí role VG

Jelikož obrázky zabírají spoustu místa a jsou podobné jako při převzetí role VF, tak zde uvádím pouze tabulku 8.9 s časy.



(a) Wireshark (b) OMNeT++

Obrázek 8.13: Porovnání paketu `R2`.

t [s]	**Událost Cisco**	t [s]	**Událost Simulace**
36	Přijetí Hello od Active R3	32	Přijetí Hello od Active R3
37	Vypnutí linky k R3	35	Vypnutí linky k R3
47	R2: Standby → Active	42	R2: Standby → Active
47	R1: Listen → Speak	52	R1: Listen → Speak
57	R1: Speak → Standby	62	R1: Speak → Standby

Tabulka 8.9: Porovnání přechodů směrovačů do stavu Active VG a Standby VG při výpadku AVG.

8.6.3 Zhodnocení

U převzetí role VF je vidět v komunikaci modře zvýrazněná odlišnost (viz obrázek 8.10), která je způsobena jak opožďováním časovačů, tak i odlišným chováním v případě posílání zpráv ve stavu *listen*, které je popsáno v kapitole 7.5.

Zobrazené porovnání paketů se v prvním (obrázek 8.11) a třetím (obrázek 8.13) případě shodují. Avšak paket odeslaný směrovačem R1 (obrázek 8.12) obsahuje odlišné číslo VF (položka `forwarder`). To je způsobeno tím, že v simulaci se spouštějí směrovače prakticky v jeden okamžik, ale u Cisco zařízení se mi toto nepodařilo, a tak směrovač R2 naběhl rychleji, než R1. R1 tak obdržel až třetí číslo pro VF.

U převzetí role VG (viz tabulka 8.9) je opět patrné opoždění v simulaci oproti Cisco zařízení. Zejména při přechodu směrovače `R1` ze stavu *listen* do stavu *speak*. Důvodem je odlišnost v Cisco implementaci popsaná výše.

8.7 Obnovení rozhraní po výpadku u GLBP

Tento test navazuje na předchozí testování výpadku. Pozoruji zde změny stavů VF a VG obnoveného směrovače. Směrovač obnovím v čase $t = 63\,s$.

Jelikož už jsem popsal možné odlišnosti v obsahu paketů, tak uvádím pouze tabulku 8.10 popisující důležité události.

t [s]	Událost Cisco
<65	R1: VG: Standby VF 2: Active R2: VG: Active VF 1: Active VF 3: Active
65	R3: Zapnutí rozhraní
67	R3: VG: Listen → Speak VF 1: Init → Listen
69	R1: Standby → Listen
77	R3: Speak → Standby
97	R3: VF 1: Listen → Active

t [s]	Událost Simulace
<63	R1: VG: Standby VF 2: Active R2: VG: Active VF 1: Active VF 3: Active
63	R3: Zapnutí rozhraní
63	R3: VF 1: Listen → Active
63	R2: VF 1: Active → Listen
65	R3: Listen → Speak
65	R1: Standby → Listen
75	R3: Speak → Standby

Tabulka 8.10: Porovnání přechodů na Cisco zařízení a v simulaci po zapnutí rozhraní směrovače `R3`.

8.7.1 Zhodnocení

V tomto testu je viditelná jedna odlišnost mé implementace oproti Cisco implementaci. Cisco směrovače využívají zpoždění, které zabraňuje obnovenému směrovači získat okamžitě zpět stav *active* pokud je PVF.

Má implementace toto zpoždění nevyužívá, a tak v momentě, kdy obnovený směrovač přijme zprávu obsahující VF nižší priority, převezme nad tímto VF kontrolu. Tato situace nastala v simulaci přesně po zapnutí rozhraní (viz tabulka 8.10, čas $t = 63\,s$), kde směrovač `R3` okamžitě přešel do stavu *active*.

Kapitola 9

Závěr

V této diplomové práci jsem se zabýval problematikou současně používaných protokolů pro redundanci síťové brány. Jednotlivé protokoly jsem nastudoval a popsal včetně jejich použití na zařízeních Cisco, kde Cisco podporuje HSRP, VRRP i GLBP.

Taktéž jsem stručně představil prostředí OMNeT++ a knihovnu ANSAINET. Knihovna obsahovala pouze implementaci protokolu VRRP verze 2 nad protokolem IPv4 vytvořenou Petrem Vítkem v roce 2013 [18] jako součást jeho diplomové práce. Toto řešení jsem prozkoumal a navázal na něj implementací protokolů HSRP a GLBP. Aktuálně se jedná o jedinou veřejně dostupnou *open–source* realizaci těchto dvou protokolů.

Při implementaci protokolu HSRP jsem vycházel zejména z RFC, popřípadě z Cisco specifikace. Dokumentace je dobře zpracovaná, a tak jsem zjišťoval specifické detaily na Cisco zařízeních pouze výjimečně.

Jelikož protokol GLBP nedisponuje RFC specifikací, využíval jsem pouze Cisco specifikaci a doplňující informace jsem získával sledováním zasílaných paketů mezi jednotlivými Cisco směrovači pomocí nástroje Wireshark. Z těchto informací jsem sestrojil stavový automat mapující změny stavů směrovačů a z něj jsem pak vytvořil samostatnou implementaci.

Protokoly jsem následně otestoval v porovnání s Cisco zařízeními ve školní laboratoři. Protokol HSRP odpovídá Cisco implementaci, až na jemné časové odchylky způsobené zpožďováním časovačů. Protokol GLBP v simulaci vykazuje větší odchylky oproti řešení u Cisco zařízení. Důvody těchto odchylek a odlišnosti mé implementace oproti Cisco implementace uvádím v textu.

Jednotlivé testy jsou součástí řešení spolu s komplexním příkladem, který slouží případným zájemcům k urychlení pochopení problematiky.

Další možná rozšíření nad protokoly FHRP spatřuji zejména v implementaci podpory pro IPv6, jelikož knihovna INET tuto podporu obsahuje. Dále je možné rozšíření v podobě reakce na různé změny stavů zařízení, kdy jednotlivé protokoly vykazují různé chování při různých reakcích. Popřípadě u GLBP implementace různých *load balancing* algoritmů.

Literatura

[1] Cisco Systems, Inc. *Hot Standby Router Protocol Features and Functionality* [online]. 2006 [cit. 14.9.2015]. Dostupné na: <http://www.cisco.com/c/en/us/support/docs/ip/hot-standby-router-protocol-hsrp/9234-hsrpguidetoc.html>.

[2] Cisco Systems, Inc. *GLBP - Gateway Load Balancing Protocol* [online]. 2009 [cit. 25.11.2015]. Dostupné na: <http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ft_glbp.html>.

[3] Cisco Systems, Inc. *Understanding and Troubleshooting HSRP Problems in Catalyst Switch Networks* [online]. 2009 [cit. 10.10.2015]. Dostupné na: <http://www.cisco.com/c/en/us/support/docs/ip/hot-standby-router-protocol-hsrp/10583-62.html>.

[4] Cisco Systems, Inc. *Cisco IOS Debug Command Reference - Commands S through Z* [online]. 2013 [cit. 20.4.2016]. Dostupné na: <http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/debug/command/s1/db-s1-cr-book/db-s1-cr-book_CLT_chapter.html>.

[5] Cisco Systems, Inc. *Cisco IOS Debug Command Reference - Commands E through H* [online]. 2014 [cit. 20.4.2016]. Dostupné na: <http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/debug/command/e1/db-e1-cr-book/db-e1.html>.

[6] Cisco Systems, Inc. *First Hop Redundancy Protocols Configuration Guide, Cisco IOS XE Release 3S* [online]. 2014 [cit. 20.5.2016]. Dostupné na: <http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipapp_fhrp/configuration/xe-3s/fhp-xe-3s-book/fhp-hsrp.html>.

[7] Cisco Systems, Inc. *Cisco IOS First Hop Redundancy Protocols Command Reference* [online]. 2015 [cit. 15.10.2015]. Dostupné na: <http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipapp_fhrp/command/fhp-cr-book.html>.

[8] Cisco Systems, Inc. *Cisco IOS Software Integrity Assurance* [online]. 2015 [cit. 20.5.2016]. Dostupné na: <http://www.cisco.com/c/en/us/about/security-center/integrity-assurance.html>.

[9] Faculty of Information Technology, Brno University of Technology. *Project ANSA* [online]. 2012 [cit. 5.5.2015]. Dostupné na: <https://nes.fit.vutbr.cz/ansa/>.

[10] Hinden, E. *RFC 3768: Virtual Router Redundancy Protocol (VRRP).* [b.m.]: RFC, IETF, duben 2004.

[11] Hucaby, D. *CCNP Switch 642-813 Official Certification Guide.* Indianapolis: Cisco Press, 2010. 268 – 289 s. ISBN 1-58720-243-3.

[12] Li, T., Cole, B., Morton, P. et al. *RFC 2281: Cisco hot standby router protocol (HSRP).* [b.m.]: RFC, IETF, březen 1998.

[13] Nadas, E. *RFC 5798: Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6.* [b.m.]: RFC, IETF, březen 2010.

[14] Natarajan, S. *Method and system for managing a network having an HSRP group.* červenec 2012. US Patent 8,213,439. Dostupné na: <http://www.google.com/patents/US8213439>.

[15] Nosella, T. a Wilson, I. *Gateway load balancing protocol.* únor 2011. US Patent 7,881,208. Dostupné na: <http://www.google.com/patents/US7881208>.

[16] Varga, A. *OMNeT++ Simulation Manual* [online]. 2015 [cit. 5.5.2016]. Dostupné na: <https://omnetpp.org/doc/omnetpp/manual/>.

[17] Varga, A., Bojthe, Z., Meszaros, L. et al. *INET Framework* [online]. 2016 [cit. 15.4.2016]. Dostupné na: <https://inet.omnetpp.org/>.

[18] Vítek, P. *Modelování protokolů pro redundanci brány.* Brno: FIT VUT v Brně, 2013. Diplomová práce.

Příloha A

Obsah CD

/ANSA/src/ansa/networklayer/hsrp/*	Zdrojové kódy k protokolu HSRP
/ANSA/src/ansa/networklayer/glbp/*	Zdrojové kódy k protokolu GLBP
/ANSA/examples/ansa/hsrp/*	Simulační scénáře a konfigurační soubory k protokolu HSRP
/ANSA/examples/ansa/glbp/*	Simulační scénáře a konfigurační soubory k protokolu GLBP
/tex	Zdrojové soubory této práce
/vsdx/*	Zdrojové soubory obrázků a diagramů
/tests/*	Kompletní výstupy testů uvedených v příloze
/InstallGuide.pdf	Návod k instalaci OMNeT++ (anglicky)
/projekt.pdf	PDF verze práce
/readme.txt	Obsah CD

Příloha B

Seznam zkratek

ARP	Address Resolution Protocol
AVF	Active Virtual Forwarder
AVG	Active Virtual Gateway
BGP	Border Gateway Protocol
BSD	Berkeley Software Distribution
CARP	Common Address Redundancy Protocol
FHRP	First Hop Redundancy Protocol
GLBP	Gateway Load Balancing Protocol
HSRP	Hot Standby Routing Protocol
IEEE	Institute of Electrical and Electronics Engineers
IOS	Internetwork Operating System
IP	Internet Protocol
IPvX	Souhrnné označení pro IP verze 4 nebo IP verze 6
KA	Konečný automat
MAC	Media Access Control
MD5	Message-Digest algorithm
OSPF	Open Shortest Path First
PPP	Point-to-Point Protocol
PVF	Primary Virtual Forwarder
SVF	Secondary Virtual Forwarder
TLV	Type length value
TTL	Time to live
UDP	User Datagram Protocol
VF	Virtual Forwarder
VG	Virtual Gateway
VHID	Virtual Host ID
vMAC	Virtual MAC
VRID	Virtual Router ID
VRRP	Virtual Router Redundancy Protocol
XML	Extensible Markup Language

Příloha C

Přechodová tabulka KA HSRP

	1 Initial	2 Learn	3 Listen	4 Speak	5 Standby	6 Active
Event	States					
a	AB/2|3+					
b		CD/1	CD/1	CD/1	CD/1	CDH/1
c			AB/4		CDFI/6	
d			B/4	D/5		
e				F	F	F
f				B/3	B/3	
g		EAB/3	EA	EA	EA	AB/4
h		EAB/3	A|BGFI/6*|	A|BGFI/6*|	A|BGFI/6*|	G
i			AB/4	A	CFI/6	
j						ABH/4
k			B	B/3	B/3	B
l			B/4	D/5		B

Tabulka je převzatá z RFC 2281 [12].

\+ Jestliže je IP adresa nakonfigurována, přejde do stavu *listen*. V opačném případě přejde do stavu *learn*. V obou případech vykoná patřičné akce.

* Při nastavené preempci je proveden přechod do stavu *active* s vykonáním patřičných akcí. Při vypnuté preempci se pouze aktualizuje *active timer*.

Příloha D

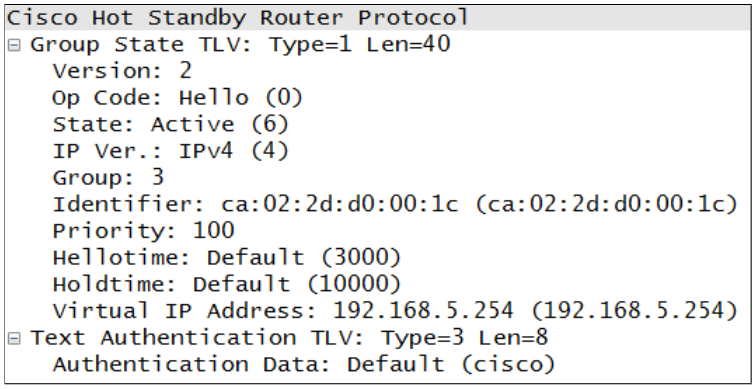
Ukázky k HSRPv2

```
R2(config-if)# do show standby

FastEthernet0/0 - Group 0
State is Active
2 state changes, last state change 00:01:22
Virtual IP address is 192.168.1.254
Active virtual MAC address is 0000.0c07.ac00 (MAC In Use)
Local virtual MAC address is 0000.0c07.ac00 (v1 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 2.624 secs
Preemption disabled
Active router is local
Standby router is 192.168.1.1, priority 100 (expires in 9.440 sec)
Priority 100 (default 100)
Group name is "hsrp-Fa0/0-0"(default)

FastEthernet1/0 - Group 3 (version 2)
State is Active
2 state changes, last state change 00:02:02
Virtual IP address is 192.168.5.254
Active virtual MAC address is 0000.0c9f.f003 (MAC In Use)
Local virtual MAC address is 0000.0c9f.f003 (v2 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 1.456 secs
Preemption disabled
Active router is local
Standby router is 192.168.5.1, priority 100 (expires in 8.512 sec)
Priority 100 (default 100)
Group name is "hsrp-Fa1/0-3"(default)
```

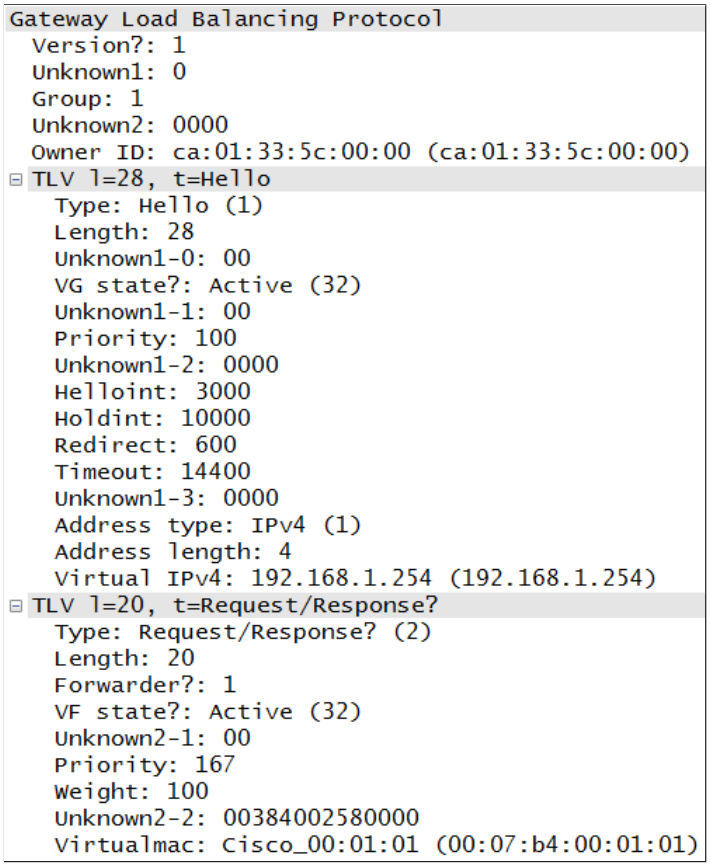
Obrázek D.1: Ukázka spuštění HSRP verze 1 a HSRP verze 2 na různých rozhraních souběžně.



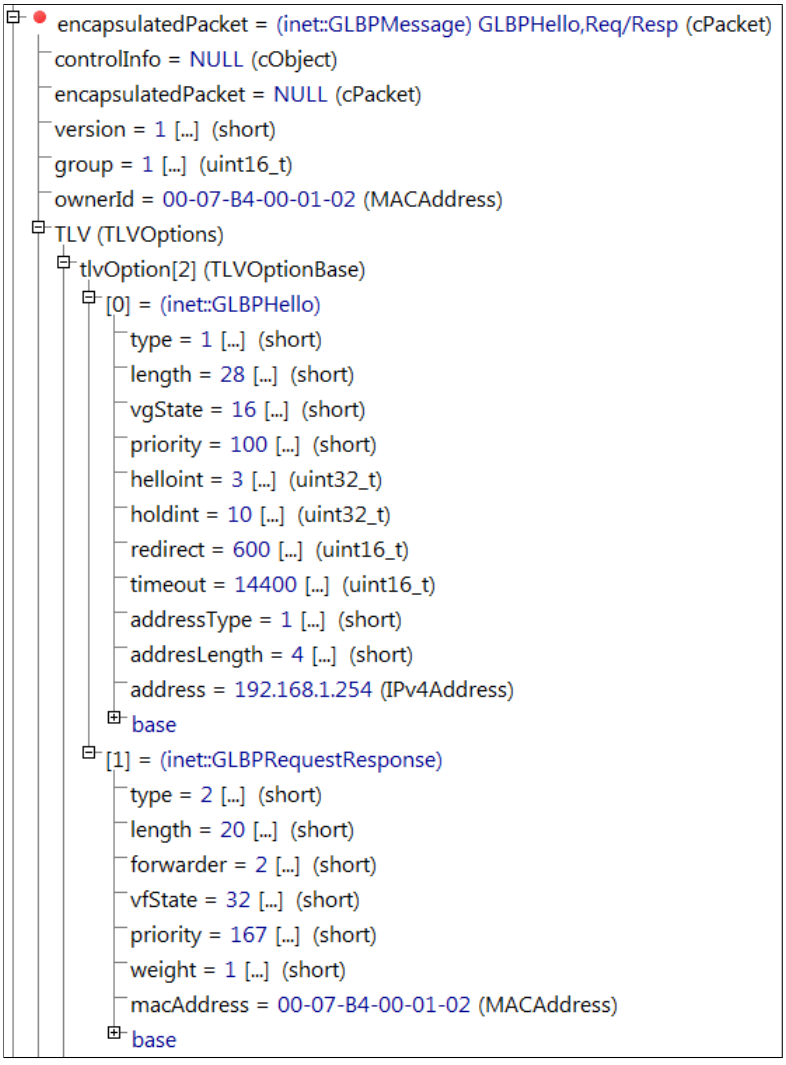
Obrázek D.2: Ukázka paketu HSRP verze 2 zachyceného ve Wiresharku.

Příloha E

Ukázka GLBP paketu



Obrázek E.1: Ukázka paketu zachyceného Wiresharkem. Paket zaslal směrovač ve stavu *active* VG a *active* pro VF 1.



Obrázek E.2: Ukázka paketu z OMNeT++ zaslaného směrovačem ve stavu *standby* VG a *active* pro VF 2.

Příloha F

Výstupy z testů

```
CISCO:
R1
09:51:43.371: HSRP: Fa0/0 Grp 0 Standby -> Listen

R3
09:51:39.811: HSRP: Fa0/0 Interface UP
09:51:40.815: HSRP: Fa0/0 Grp 0 Init -> Listen
09:51:43.343: HSRP: Fa0/0 Grp 0 Listen -> Speak
09:51:55.291: HSRP: Fa0/0 Grp 0 Speak -> Standby
```

```
SIMULACE:
R1
t=64.000037739997 eth0 Grp 0 Standby -> Listen
t=74.000037739997 eth0 Grp 0 Listen -> Speak
t=74.000037739997 eth0 Grp 0 Speak -> Listen

R3
t=63 eth0 Interface up
t=63 eth0 Grp 0 Init -> Listen
t=64.000025159998 eth0 Grp 0 Listen -> Speak
t=74.000025159998 eth0 Grp 0 Speak -> Standby
```

Obrázek F.1: Výstupy z testu obnovení rozhraní po výpadku u HSRP.

```
CISCO:
R1
11:09:24.915: HSRP: Fa0/0 Interface UP
11:09:25.883: HSRP: Fa0/0 Grp 0 Init -> Listen
11:09:37.723: HSRP: Fa0/0 Grp 0 Listen -> Speak
11:09:40.451: HSRP: Fa0/0 Grp 0 Speak -> Listen
11:09:53.871: HSRP: Fa0/0 Grp 0 Listen -> Speak
11:09:55.827: HSRP: Fa0/0 Grp 0 Speak -> Listen

R2
11:09:24.307: HSRP: Fa0/0 Interface UP
11:09:25.271: HSRP: Fa0/0 Grp 0 Init -> Listen
11:09:36.127: HSRP: Fa0/0 Grp 0 Listen -> Speak
11:09:37.519: HSRP: Fa0/0 Grp 0 Speak -> Listen
11:09:55.831: HSRP: Fa0/0 Grp 0 Listen -> Speak
11:10:07.439: HSRP: Fa0/0 Grp 0 Speak -> Standby

R3
11:09:23.763: HSRP: Fa0/0 Interface UP
11:09:24.723: HSRP: Fa0/0 Grp 0 Init -> Listen
11:09:34.799: HSRP: Fa0/0 Grp 0 Listen -> Speak
11:09:43.791: HSRP: Fa0/0 Grp 0 Speak -> Standby
11:09:46.283: HSRP: Fa0/0 Grp 0 Standby -> Active
```

```
SIMULACE:
R1
t=0 eth0 Grp 0 Disabled -> Init
t=0 eth0 Grp 0 Init -> Listen
t=10 eth0 Grp 0 Listen -> Speak
t=13.000013539998 eth0 Grp 0 Speak -> Listen
t=30.000019299998 eth0 Grp 0 Listen -> Speak
t=33.000032839996 eth0 Grp 0 Speak -> Listen

R2
t=0 eth0 Grp 0 Disabled -> Init
t=0 eth0 Grp 0 Init -> Listen
t=10 eth0 Grp 0 Listen -> Speak
t=13.000020259997 eth0 Grp 0 Speak -> Listen
t=30.000019299998 eth0 Grp 0 Listen -> Speak
t=40.000019299998 eth0 Grp 0 Speak -> Standby

R3
t=0 eth0 Grp 0 Disabled -> Init
t=0 eth0 Grp 0 Init -> Listen
t=10 eth0 Grp 0 Listen -> Speak
t=20 eth0 Grp 0 Speak -> Standby
t=20 eth0 Grp 0 Standby -> Active
```

Obrázek F.2: Výstupy z testu pro volbu virtuální brány u HSRP.

```
CISCO:
R1
18:22:44.099: GLBP: Fa0/0 Interface up
18:22:44.103: GLBP: Fa0/0 0 Init -> Listen
18:22:56.115: GLBP: Fa0/0 0.1 Disabled -> Listen
18:22:57.523: GLBP: Fa0/0 0.3 Disabled -> Listen
18:23:02.251: GLBP: Fa0/0 0.2 Disabled -> Listen
18:23:02.771: GLBP: Fa0/0 0 Listen -> Speak
18:23:02.787: GLBP: Fa0/0 0 Speak -> Listen
18:23:08.891: GLBP: Fa0/0 0.3 Listen -> Active
R2
18:22:43.715: GLBP: Fa0/0 Interface up
18:22:43.719: GLBP: Fa0/0 0 Init -> Listen
18:22:56.379: GLBP: Fa0/0 0.1 Disabled -> Listen
18:22:57.343: GLBP: Fa0/0 0.2 Disabled -> Listen
18:23:02.939: GLBP: Fa0/0 0.3 Disabled -> Listen
18:23:03.039: GLBP: Fa0/0 0 Listen -> Speak
18:23:09.099: GLBP: Fa0/0 0.2 Listen -> Active
18:23:13.067: GLBP: Fa0/0 0 Speak -> Standby
R3
18:22:43.483: GLBP: Fa0/0 Interface up
18:22:43.487: GLBP: Fa0/0 0 Init -> Listen
18:22:53.407: GLBP: Fa0/0 0 Listen -> Speak
18:22:53.831: GLBP: Fa0/0 0 Speak -> Active
18:22:53.839: GLBP: Fa0/0 0.1 Disabled -> Listen
18:23:00.355: GLBP: Fa0/0 0.2 Disabled -> Listen
18:23:00.723: GLBP: Fa0/0 0.3 Disabled -> Listen
18:23:04.767: GLBP: Fa0/0 0.1 Listen -> Active
```

```
SIMULACE:
R1
t=0 Grp 0 Init -> Listen
t=10 Grp 0 Listen -> Speak
t=10.000015779998 Grp 0 Speak -> Listen
t=20 Grp 0 Listen -> Speak
t=20.000015779998 Grp 0 Speak -> Listen
t=30.000023299998 Fwd 1 Grp 0 Disabled -> Listen
t=30.000080659993 Fwd 2 Grp 0 Listen -> Active
t=30.000104599991 Fwd 3 Grp 0 Disabled -> Listen
R2
t=0 Grp 0 Init -> Listen
t=10 Grp 0 Listen -> Speak
t=10.000023619997 Grp 0 Speak -> Listen
t=20 Grp 0 Listen -> Speak
t=20.000088499992 Fwd 3 Grp 0 Disabled -> Listen
t=30 Grp 0 Speak -> Standby
t=30.000016419998 Fwd 1 Grp 0 Disabled -> Listen
t=30.000088499992 Fwd 3 Grp 0 Listen -> Active
t=30.000096759992 Fwd 2 Grp 0 Disabled -> Listen
R3
t=0 Grp 0 Init -> Listen
t=10 Grp 0 Listen -> Speak
t=20 Grp 0 Speak -> Active
t=20 Fwd 1 Grp 0 Disabled -> Listen
t=30 Fwd 1 Grp 0 Listen -> Active
t=30.000096759992 Fwd 2 Grp 0 Disabled -> Listen
t=30.000105239991 Fwd 3 Grp 0 Disabled -> Listen
```

Obrázek F.3: Výstupy z testu volby virtuální brány u GLBP.