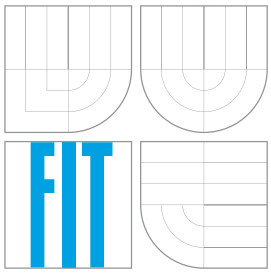
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MODELOVÁNÍ PROTOKOLŮ PRO MANAGEMENT NA ÚROVNI L2

MODELLING OF L2 MANAGEMENT PROTOCOLS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. TOMÁŠ RAJCA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. VLADIMÍR VESELÝ

BRNO 2016

Abstrakt

Tato práce se zabývá modelováním a simulací protokolů pro správu sítě na linkové vrstvě v nástroji OMNeT++. Jedná se o protokoly CDP, LLDP a ODR směrování. V první polovině jsou tyto protokoly popsány a v druhé polovině je naznačen jejich návrh a implementace v rámci projektu ANSA. Korektnost implementace je ověřena porovnáním simulace a reálné sítě na příkladech. V rámci práce bylo provedeno odstranění závislosti na modulu DeviceConfiguratoru používaného v knihovně ANSAINET.

Abstract

This thesis deals with modelling and simulation of management protocols on the data-link layer in OMNeT++ tool. Namely protocol CDP, LLDP and ODR routing. These protocols are described in the first thesis' half and in the second half is described their design and implementation in ANSA project. Correctness of implementation is verified by comparison between simulated and real network examples. Also dependencies on module DeviceConfiguratoru were removed from ANSAINET library.

Klíčová slova

OMNeT++, simulace sítí, INET, ANSA, CDP, ODR, LLDP.

Keywords

OMNeT++, network simulation, INET, ANSA, CDP, ODR, LLDP.

Citace

RAJCA, Tomáš. *Modelování protokolů pro management na úrovni L2.* Brno, 2016. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Veselý Vladimír.

Modelování protokolů pro management na úrovni L2

Prohlášení

.....
Tomáš Rajca
24. května 2016

Poděkování

Na tomto místě bych chtěl poděkovat Vladimírovi za odbornou pomoc a poskytnuté rady při psaní této práce. Obrovské díky rovněž patří všem kolegům, kteří mě v průběhu studia doprovázeli a pomáhali mi jej zvládat. Obzvláště pak Honzu Holušovi, jehož zdravý duch a společně vytvořený tým, pomohl překonat nejednu překážku.

Samozřejmě největší díky patří mé rodině za podporu během celého studia, které se již doufám blíží ke konci. Speciální díky rovněž patří mé švagrové Daniele za trpělivost při gramatické korektuře, která musela být celkem intenzivní.

Jako vyjádření díku bych chtěl věnovat recept na nejzákladnější pokrm, který lidstvo zná, a to chléb. Konkrétně Barbari (Perský sezamový chléb).Potřebujeme 1 lžičku tekutého medu, 325 ml vody, 2 lžičky sušeného droždí, 500 g hladké pšeničné mouky, 1,5 lžičky soli, 2 lžíce olivového oleje plus navíc na potření, 2 lžíce sezamového semínka.

Troubu rozpalte na 220 stupňů Celsia. Do misky se 175 ml vody nejprve vmíchejte med a pak přisypte droždí. Nechte 5 minut stát. Pak promíchejte, aby se droždí rozpustilo. Počkejte, až vzejde kvásek. Ve velké míse smíchejte mouku a sůl. Do středu udělejte důlek, přidejte vodu s droždím a přilévejte pomalu zbytek vody a olej. Vymíchejte pevné, vlhké těsto. Těsto vyklopte na lehce pomoučený vál a hněťte, až je hladké, lesklé a pružné (cca 10 minut). Vložte do olejem vymazané mísy, otočte v míse, aby se obalilo olejem a zakryjte utěrkou. Nechte kynout asi 1,5 hodiny, až zdvojnásobí objem. Poté rozdělte na 4 stejné díly, každý vytvarujte do placky o průměru 12 cm a tloušťce 2,5 cm. Zakryjte utěrkou a nechte dokynout asi 45 minut. Dva plechy posypte moukou a dejte na 15 minut ohřát do trouby. Do každé placky udělejte prstem devět důlků asi 2 cm hlubokých. Placky pomažte olivovým olejem a posypte sezamovým semínkem. Placky položte na horké plechy a pečte asi 20 minut dozlatova. Upečené placky musí znít při poklepu na spodní stranu dutě. Nechte vychladnout na mřížce.

© Tomáš Rajca, 2016.
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Kapitola 1

Úvod

Každým dnem se složitost síťových technologií zvětšuje a zároveň rapidně přibývá počet zařízení, do kterých jsou implementovány. Při zavádění nových technologií je tedy potřeba mít jistotu, že budou pracovat dle předpokladu. Sebemenší odchylka od předpokládané funkčnosti by mohla vést ke zkolabování celé sítě, čemuž se chce každý správce sítě vyhnout.

Před zavedením nové technologie do provozu je ji tedy vhodné vyzkoušet, aniž by došlo k ohrožení funkcionality sítě. Ideální je testovat na jiných fyzických zařízeních, což ale vyžaduje vlastnit nadbytečná zařízení. Tento způsob rovněž není použitelný pro větší sítě. Vhodnou alternativou je využít nějaký simulační nástroj.

Simulační nástroj umožňuje namodelovat aktuální nebo zamýšlenou topologii sítě. Takto namodelovanou síť pak lze podrobit různým testům. Některé simulační nástroje obsahují i grafické uživatelské rozhraní, kterým lze vizuálně sledovat chování sítě.

Pro simulaci sítě lze využít simulační nástroj OMNeT++ [16], který díky knihovně INET [11] umí simulovat i počítačové sítě. Knihovna ale ještě zdaleka neobsahuje všechny používané protokoly počítačových sítí. Zatím chybí jakýkoli protokol pro získávání informací o sousedních zařízeních na linkové vrstvě. Na naší fakultě v rámci projektu ANSA [5] dochází k rozšiřování této knihovny.

Tato práce se tedy zabývá modelováním existujících protokolů pro správu sítě na linkové vrstvě v prostředí OMNeT++, jmenovitě CDP (*Cisco Discovery Protocol*), LLDP (*Link Layer Discovery Protocol*) a ODR (*On – Demand Routing*). Součástí práce je i odstranění závislosti na modulu *DeviceConfigurator* používaného v ANSA projektu

Kapitola 2 obsahuje stručný popis CDP protokolu. Nejprve je popsán princip funkce protokolu, následně popis rozšíření ODR a na konci příkazy k nastavení a ověření funkčnosti na Cisco zařízeních. V podobném duchu se následující 3. kapitola zabývá LLDP. Nejprve popisuje protokol, následně LLDP-MED rozšíření a příkazy k nastavení a ověření funkčnosti na zařízeních firmy Cisco. V kapitole 4 je stručně popsáno, odstranění závislosti na modulu *DeviceConfigurator*. V následujících dvou kapitolách 5 a 6 je popsán návrh a implementace těchto protokolů. Popis ověřování korektnosti implementace je v kapitole 7. V závěrečné 8. kapitole jsou shrnuty výsledky práce a možné směry pokračování.

Kapitola 2

CDP

Cisco Discovery Protocol (CDP) je proprietární protokol navržený firmou Cisco Systems[1] pro svá zařízení. Jedná se o protokol druhé vrstvy síťového modelu OSI, který síťová zařízení využívají ke zjištění informací o sousedních (přímo propojených) zařízeních. Funguje na všech přenosových médiích, které podporují *Subnetwork Access Protocol* (SNAP), což je standard používaný pro přenos IP datagramů přes IEEE 802[2] sítě. Mezi ně patří například LAN[3], Frame Relay[4], ATM[5], PPP[6] fyzické média. Je zároveň i kompatibilní mezi dvěma zařízeními, jež pracují na jiném síťovém protokolu (IP, AppleTalk, IPX, atd...). Funguje na všech směrovačích, síťových mostech, přístupových serverech a přepínačích firmy Cisco. CDP rámce dokáží rozpoznat i některá zařízení firmy HP. Firmy jako Dell nebo Netgear používají vlastní implementaci CDP pod názvem ISDP[7].

Nejnovější verzí protokolu je verze 2 (CDPv2). Její výhodou je, že poskytuje inteligentnější funkce pro sledování zařízení. Jednou z nich je mechanismus ohlašování chyb, který umožňuje zásadně snížit dobu nečinnosti sítě. K ohlašovaným chybám patří nekonzistentní nativní VLAN ID (IEEE 802.1Q) na propojených portech a nekonzistentní port-duplex stav mezi sousedními zařízeními. Zprávy o chybách mohou být zasílány do konzole nebo do logovacího serveru.

Z CDP lze zjistit i adresu SNMP[8] agenta sousedního zařízení. Tato funkce umožňuje posílat SNMP dotazy sousedním zařízením.

Informace v této kapitole jsou čerpány z následujících zdrojů: [2], [4], [7], [8], [14] a [15].

2.1 Princip funkce

Zařízení periodicky posílají svá CDP oznámení na všechna připojená rozhraní podporující SNAP hlavičky. Ve výchozím nastavení jsou posílány v 60 sekundových intervalech. Jako cílovou adresu mají nastavenou veřejnou multicastovou MAC adresu 01-00-0C-CC-CC-CC. Informace z přijatých zpráv jsou uloženy v tabulce sousedů. Každé oznámení obsahuje informaci *Time To Live* (TTL), jež určuje, po jaké době ji má přijímající zařízení vymazat ze

[1] http://www.cisco.com
[2] https://cs.wikipedia.org/wiki/IEEE_802
[3] LAN - Local Area Network, https://en.wikipedia.org/wiki/Local_area_network
[4] Frame Relay, https://tools.ietf.org/html/rfc2427
[5] ATM - Asynchronous Transfer Mode, https://tools.ietf.org/html/rfc4454
[6] PPP - Point-to-Point Protocol, https://tools.ietf.org/html/rfc1661
[7] ISDP - Industry Standard Discovery Protoco, http://se.labri.fr/data/sysadmin/user-guide-switch.pdf
[8] SNMP - Simple Network Management Protocol, https://tools.ietf.org/html/rfc3413

své tabulky sousedů. Informace v tabulce jsou aktualizované po každém přijatém oznámení a smazány, pokud vyprší *holdtime* časovač záznamu. Tento časovač je nastaven na hodnotu TTL posledního oznámení. Ke smazání záznamu v tabulce může také dojít, pokud pole TTL příchozího oznámení obsahuje hodnotu 0.

Když dojde k zapnutí rozhraní, zapne se režim rychlého startu. V tomto režimu dojde k odesílání oznámení v intervalu jedné sekundy po dobu tří sekund.

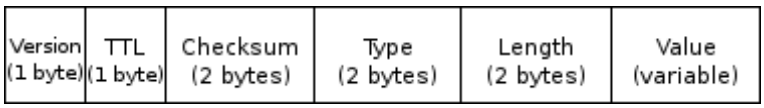
Informace obsažené v CDP oznámeních jsou závislá na typu zařízení a nainstalované verzi operačního systému. Všechna oznámení se skládají z hlavičky a libovolně dlouhé posloupnosti prvků *Type Length Value*[9] (TLV). Hlavička oznámení obsahuje tři části:

- *Version* - verze CDP protokolu.
- *TTL* - čas, po který má být informace uchována přijímacím zařízením (v sekundách). Doporučeno, aby se jednalo o trojnásobek periody posílání oznámení (ve výchozím nastavení $3 \times 60\,\mathrm{s} = 180\,\mathrm{s}$).
- *Checksum* - kontrolní součet CDP rámce. Spočte se stejným způsobem jak v IP hlavičce[10].

Každá TLV položka v oznámení obsahuje rovněž tři části:

- *Type* - určuje typ TLV položky.
- *Length* - délka TLV položky (v bajtech).
- *Value* - obsahuje samotnou informaci.

Příklad takového CDP rámce je zobrazen níže na obrázku 2.1.



Obrázek 2.1: Příklad CDP rámce s jednou TLV položkou.

V případě, že na zařízení je zapnuté CDP verze jedna, které přijme rámec verze dva, dojde k zahození pro něho neznámých TLV a zbytek rámce se dále zpracuje.

2.2 Typy TLV

Jak již bylo zmíněno výše, TLV jsou bloky dat vnořené v CDP oznámeních. Díky TLV formátu rámce, lze oznámení rozšířit o další položky, pokud je to potřeba. Typy TLV, které mohou být přenášeny, jsou přesně specifikovány. Některé z těchto typů jsou vypsany níže:

- *Address TLV* (0x0002) - obsahuje síťovou adresu jak přijímacího tak odesílacího zařízení.
- *Capabilities TLV* (0x0004) - identifikuje typ zařízení, jako například přepínač.
- *Device-ID TLV* (0x0001) - identifikuje název zařízení ve formě ASCII řetězce.

[9] TLV - Type-Length-Value, https://cs.wikipedia.org/wiki/Type-length-value
[10] https://tools.ietf.org/html/rfc1071

- *Full or Half Duplex TLV* (0x000B) - umožňuje zařízením rozpoznat, zda spojení je plně duplexní nebo poloduplexní. Tato informace je využívaná administrátory pro diagnostiku problému konektivity mezi sousedními síťovými zařízeními.
- *IP síťový prefix TLV* (0x0007) - toto TLV může obsahovat jednu ze dvou typů informací:
  - IP prefixy - jednotlivé IP prefixy označují přímo připojené segmenty sítě. Každý IP prefix se skládá ze 4 bytů IP adresy sítě a 1 bytu masky. Tato maska může být v rozsahu 0 až 32, kdy dané číslo udává počet bitů masky, nastavených na hodnotu 1.
  - výchozí cestu - v případě, že se jedná o ODR hub (viz. 2.3), TLV obsahuje výchozí cestu. Výchozí cesta se skládá z 4 bytů IP adresy. Maska v tomto případě není potřeba.
- *Location TLV* (0x000C) - doručí informace o poloze koncovému zařízení přes přístupové zařízení (směrovač nebo přepínač).
- *Native VLAN TLV* (0x000A) - oznamuje ISL[11] číslo nativní VLAN rozhraní pro netagované pakety. Tato položka je implementována pouze u rozhraní, která podporují IEEE 802.1Q protokol.
- *Platform TLV* (0x0006) - popisuje hardwarovou platformu zařízení. Například *Cisco 4500*.
- *Port-ID TLV* (0x0003) - identifikuje port, ze kterého byl CDP rámec odeslán.
- *Version TLV* (0x0005) - obsahuje informaci o verzi softwaru spuštěného na zařízení.
- *VTP Management Domain TLV* (0x0009) - specifikuje nastavený název domény protokolu VTP[12]. Tento název je používán administrátory k ověření domény VTP konfigurace sousedních uzlů.

2.3 ODR

On – Demand Routing (ODR) je rozšíření protokolu CDP, jež slouží k objevování dalších Cisco zařízení na broadcastovém nebo nebroadcastovém médiu. Slouží k dynamickému šíření IP prefixů připojených směrovačů skrze druhou vrstvu OSI modelu. Nejedná se ale o směrovací protokol. Do CDP byl přidán v Cisco IOS[13] verze 12.0 [9]. Tato vlastnost zabere pět bytů pro každou síť (čtyři byty pro IP adresu a jeden byt pro šíření masky podsítě). ODR umí přenášet i adresu s proměnnou délkou masky podsítě (VLSM[14]). Velikou výhodou této technologie je, že minimálně zatěžuje CPU a zároveň zbytečně nezabírá velkou šířku pásma.

ODR je ideálním řešením pro hvězdicovou topologii zapojení směrovačů. Topologie je znázorněna na obrázku 2.2. ODR se v takovém případě spouští na centrálním směrovači,

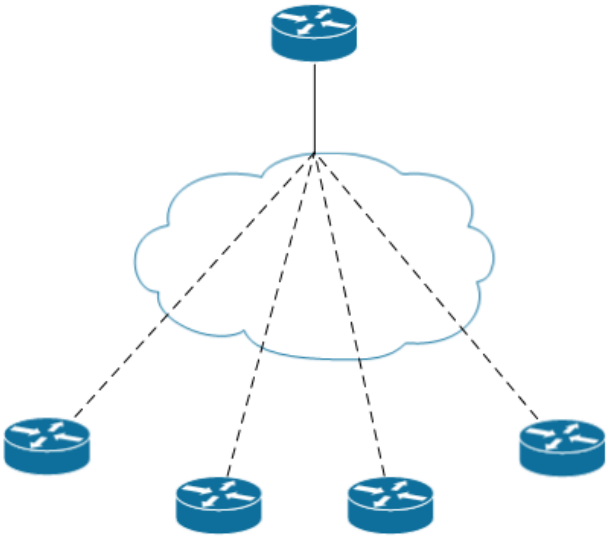
[11] ISL - Inter-Switch Link, http://www.cisco.com/c/en/us/support/docs/lan-switching/8021q/17056-741-4.html

[12] VTP - VLAN Trunking Protocol, http://www.cisco.com/c/en/us/support/docs/lan-switching/vtp/10558-21.html

[13] Operační systém používaný na směrovačích a přepínačích firmy Cisco Systems.

[14] VLSM - Variable-Length Subnet Mask, www.ciscopress.com/articles/article.asp?p=2169292&seqNum=2

který jako jediný může mít spuštěné i další směrovací protokoly (jako například RIP[15], OSPF[16], EIGRP[17]). Informace získané přes ODR se tak mohou šířit redistribucí i do dalších částí sítě, a to pomocí jiných směrovacích protokolů.



Obrázek 2.2: Hvězdicová topologie zapojení směrovačů.

Na zařízeních s IOS verze 12.0.5T a novějším, není potřeba manuálně nastavovat statickou výchozí cestu. Centrální zařízení totiž každému okrajovému zařízení automaticky posílá výchozí cestu.

ODR očekává, že bude dostávat periodické CDP oznámení obsahující IP prefixy. V případě, že oznámení nějaký čas neobdrží, označí cestu za neplatnou, ale stále přeposílá touto cestou pakety. Po dalším časovém úseku ji označí jako nedostupnou a přestane ji používat pro přeposílání (pokusí se najít jinou cestu k cíli). Za nějaký čas tuto cestu smaže ze směrovací tabulky. Výchozí nastavení těchto časovačů je založené na výchozí hodnotě posílaní CDP oznámení a verzi IOS. Na IOS verze 15.4(2)T4 je ve výchozím nastavení nastaveno zneplatnění a označení cesty jako nedostupné ve stejný čas, a to v 180. sekundě. Ke smazání cesty pak dojde za 240 sekund od příjmu posledního oznámení obsahujícího tuto cestu. Jestliže se změní CDP časovače, je dobré podle této změny upravit i ODR časovače.

2.4 Konfigurace

Ve výchozím nastavení je CDP zapnuté na všech podporovaných zařízeních. Jestliže nechceme mít na zařízení zapnuto CDP, lze ho vypnout nebo zapnout těmito příkazy:

```
Router(config)# no cdp run
Router(config)# cdp run
```

[15]RIP, https://tools.ietf.org/html/rfc4822
[16]OSPF, https://tools.ietf.org/html/rfc7474
[17]EIGRP, http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/16406-eigrp-toc.html

Rovněž je možné vypnout nebo zapnout CDP pouze na určitém rozhraní zařízení. Vypnout CDP pouze na určitém rozhraní může být za určitých okolností užitečné. V případě, že je k rozhraní připojen počítač uživatele, je to vyloženě doporučeno. Počítač neumí rozpoznávat CDP rámce, a tak se jedná jen o zbytečnou komunikaci. A v případě, že by byl u počítače útočník, získal by informace o síti, které by mohl zneužít. Příkazy pro zapnutí a vypnutí CDP na určitém rozhraní jsou:

Router(config-if)# cdp enable
Router(config-if)# no cdp enable

Na všech zařízeních s IOS verze 12.0 a vyšším je ve výchozím nastavení implicitně nastaveno šíření rámců CDPv2. Lze je zakázat nebo znovu povolit následujícími příkazy:

Router(config)# no cdp advertise-v2
Router(config)# cdp advertise-v2

Časový interval odesílání oznámení *timer* a čas, po který si má zařízení pamatovat přijaté informace před jejich odstraněním *holdtime*, lze upravit příkazy uvedenými níže. Je doporučeno, aby interval odesílání oznámení byl alespoň třikrát menší než interval odstraňování informací. Časovač *timer* lze nastavovat v rozsahu 5 až 254 a časovač *holdtime* v rozsahu 10 až 255.

Router(config)# cdp timer *seconds*
Router(config)# cdp holdtime *seconds*

ODR se nastavuje pouze na centrálním směrovači. Tento směrovač pak umí přijaté IP prefixy sousedních směrovačů zapisovat do své směrovací tabulky. K zapnutí ODR podpory slouží příkaz:

Router(config)# router odr

Tímto příkazem je zároveň umožněn vstup do konfiguračního módu ODR routingu. V tomto módu lze pomocí ACL omezit IP adresy, které se bude směrovač dynamicky učit skrze ODR. K tomuto slouží příkaz:

Router(config-router)# distribute-list *access-list-number* | *access-list-name* | `prefix` *list-name* {`in` | `out`} [*interface-type interface-number*]

V konfiguračním módu ODR routingu lze ještě upravit nastavení časovačů.

Router(config-router)# timers basic *update invalid holddown flush* [*sleeptime*]

Parametr *invalid* určuje, po jaké době od přijetí poslední aktualizace je cesta zneplatněna, ale pořád je využívaná k přeposílání paketů. Je doporučeno, aby hodnota *invalid* byla alespoň třikrát větší než hodnota *update*. Po uplynutí *holdtime* časovače jsou cesty z jiných zdrojů přijaty a cesta s vypršeným *holdtime* časovačem označená jako nedostupná. Ze směrovací tabulky se vymaže po uplynutí času *flush* od poslední aktualizace, který by měl být alespoň tak velký jako součet *invalid* a *holdtime*. Volitelným parametrem je *sleeptime*, kterým lze nastavit čas v milisekundách odkládání posílání směrovacích informací.

2.5 Ověření konfigurace

Pomocí příkazů `show` lze zobrazit různé podrobnosti ohledně konfigurace nebo běhu CDP. V této podkapitole bude pro názornou ukázku ke každému příkazu `show` zobrazen i příklad výpisu směrovače, a následně popsáno, co vše jde z výpisu vyčíst.

Pro zjištění podrobností týkajících se nastavení CDP na zařízení lze použít příkaz:

```
Router# show cdp
Global CDP information:
    Sending CDP packets every 60 seconds
    Sending a holdtime value of 180 seconds
    Sending CDPv2 advertisements is enabled
```

Jak lze vidět výše, zařízení vypíše nastavený interval odesílání oznámení (60 s), časový interval, po jehož dobu je oznámení na daném portu validní (180 s) a verzi oznámení (CDPv2). Pokud se za tento příkaz napíše ještě `interface` a identifikátor rozhraní, kromě výše zmíněných informací se vypíše i pomocí kterého protokolu je rámec na daném rozhraní zapouzdřen (například HDLC[18]).

Informace ohledně sousedních zařízení lze vypsat příkazem:

```
Router# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater
Device ID        Local Intrfce  Holdtme  Capability  Platform   Port ID
C2950-1          Fas 0/0        148      S I         WS-C2950   Fas 0/15
RX-SWV.cisco.com Fas 0/1        167      T S         WS-C3524   Fas 0/13
```

Z příkladu lze vyčíst, že k zařízení jsou připojena dvě jiná zařízení, která mají zapnuté CDP. Zařízení s názvem *C2950-1* je připojeno k rozhraní *FastEthernet 0/0*. Pokud od tohoto zařízení nepřijde do 148 sekund žádný CDP rámec, záznam o zařízení se vymaže z tabulky. Ze sloupce *Capability* lze vyčíst o jaký typ zařízení se jedná a některé funkce, které podporuje. V případě prvního záznamu se tedy jedná o přepínač s podporou IGMP[19]. Poslední dva sloupce uvádějí platformu připojeného zařízení a kterým portem je připojen.

Pokud se za příkaz *show cdp neighbors* přidá klíčové slovo `detail`, kromě výše zmíněných věcí jde z výpisu vyčíst i ID nativní VLANy, duplex mód a VTP název domény připojeného zařízení. Zobrazí se i podrobnější informace o platformě.

[18] HDLC - High-Level Data Link Control, https://tools.ietf.org/html/rfc1662
[19] IGMP - Internet Group Management Protocol, https://tools.ietf.org/html/rfc4604

Podrobnější informace o sousedním zařízení lze vypsat příkazem:

```
Router# show cdp entry device.cisco.com
    Device ID: device.cisco.com
    Entry address(es):
    IP address: 10.1.17.24
    IPv6 address: FE80::203:E3FF:FE6A:BF81 (link-local)
    IPv6 address: 4000::BC:0:0:C0A8:BC06 (global unicast)
    CLNS address: 490001.1111.1111.1111.00
    Platform: cisco 3640, Capabilities: Router
    Interface: Ethernet0/1, Port ID (outgoing port): Ethernet0/1
    Holdtime : 160 sec
    Version :
    Cisco Internetwork Operating System Software
    IOS (tm) 3600 Software (C3640-A2IS-M), Experimental Version 12.2
    Copyright (c) 1986-2001 by cisco Systems, Inc.
    Compiled Wed 08-Aug-01 12:39 by joeuser
```

Nahrazením názvu zařízení znakem `*`, se zobrazí podrobné informace o všech sousedních zařízeních. Pokud se na konec příkazu přidá slovo `version`, výpis se omezí pouze na informace ohledně verze softwaru sousedního zařízení. Stejně tak lze omezit výpis pouze na informace o protokolu spuštěném na sousedním zařízení příkazem `protocol`.

Zobrazit informace o rozhraních, na kterých je CDP spuštěné, lze zobrazit následujícím příkazem:

```
Router#show cdp interface
FastEthernet0/1 is up, line protocol is up
    Sending CDP packets every 60 seconds
    Holdtime is 180 seconds
FastEthernet0/2 is up, line protocol is up
    Sending CDP packets every 60 seconds
    Holdtime is 180 seconds
```

Rovněž výpis tohoto příkazu lze omezit pouze na určité rozhraní dopsáním typu rozhraní a portu za příkaz.

Pro zobrazení stavu čítačů slouží příkaz:

```
Router# show cdp traffic
CDP counters :
    Total packets output: 81684, Input: 81790
    Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
    No memory: 0, Invalid packet: 0, Fragmented: 0
    CDP version 1 advertisements output: 0, Input: 0
    CDP version 2 advertisements output: 81684, Input: 81790
```

Příkaz vypíše počet všech jak odeslaných, tak i přijatých CDP rámců verze 1 i 2. Kromě toho lze vyčíst i počet zahozených rámců, a to například kvůli špatnému kontrolnímu součtu nebo nedostatku paměti. Tyto čítače lze vynulovat příkazem:

```
Router# clear cdp counters
```

Pro vymazání celé tabulky o sousedních zařízeních slouží příkaz:

```
Router# clear cdp table
```

Kapitola 3

LLDP

LLDP (*Link Layer Discovery Protocol*) stejně jako CDP pracuje na linkové vrstvě, a to konkrétně na jeho horní podvrstvě LLC[1], díky čemuž může fungovat i mezi dvěma zařízeními s jinými síťovými protokoly. LLC podvrstva zapouzdřuje LLDP zprávy do výstupních rámců.

Protokol je standardizovaný v dokumentu IEEE 802.1AB-2009[3], takže může být využit více výrobci na svých zařízeních. Cílem při vytváření bylo právě nahradit a vylepšit již existující a nekompatibilní proprietární protokoly jako CDP, EDP a další. Existují i implementace pro Linux/Unix a Windows, pomocí kterých lze získat informace i o koncových zařízeních. Pro zjišťování informací o koncových zařízeních slouží LLDP-MED (viz. 3.4) rozšíření.

V zařízení může být spuštěn jeden nebo více LLDP agentů, kdy každý agent je spjat s konkrétním portem. Agenti odesílají informace o aktuálním stavu systému a portu a zároveň přijímají zprávy odeslané sousedy.

Jedná se o jednosměrný protokol, takže si nemůže vyžádat žádné informace od sousedů a zároveň ani nedochází k potvrzování informací. V zařízení se nacházejí dvě *management information base* (MIB) databáze, a to místní a vzdálená. Do vzdálené MIB databáze se ukládají informace přijaté od sousedů. Místní MIB databáze uchovává aktuální informace o stavu zařízení a jeho agentech. Závislost mezi LLC entitou, LLDP agentem a MIB databází je zobrazen na obrázku 3.1. Z těchto hodnot jsou pak sestavovány LLDP zprávy. K těmto MIB databázím může *network management system* (NMS) přistupovat použitím nějakého protokolu pro správu zařízení, jako například SNMP. V případě získání všech údajů z MIB databází všech zařízení lze pomocí algoritmů pro procházení stavového prostoru získat kompletní topologii sítě.

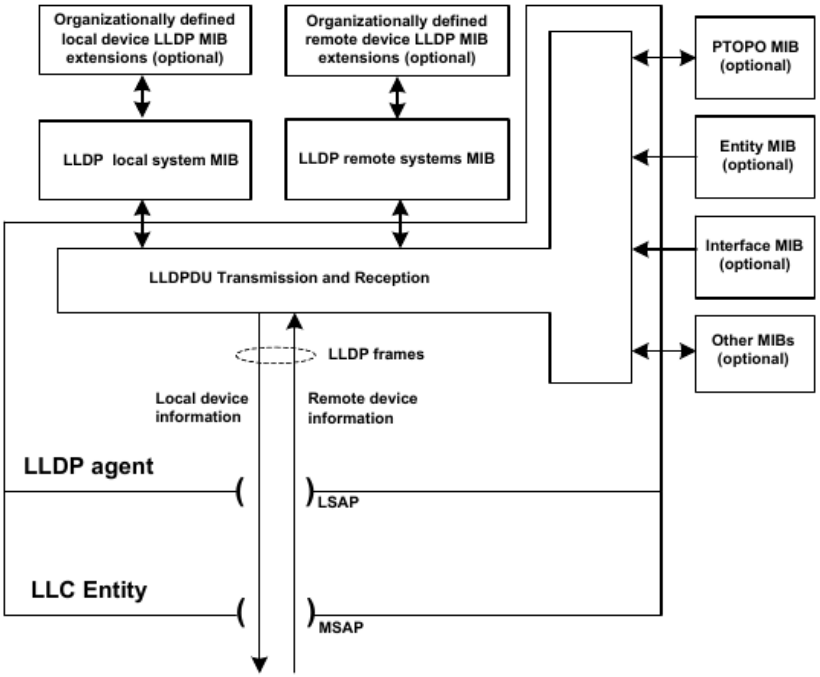
Tato kapitola čerpá z následujících zdrojů: [1], [3], [6], [10], [12] a [13].

3.1 Struktura rámců

Protokol je kompatibilní s normou IEEE 802 a k odesílaní rámců využívá služeb linkové vrstvy. Hlavička IEEE 802.3 rámce je nastavena následovně:

- Cílová adresa - podle toho, které zařízení má rámec zachytit, se používá jedna ze tří rezervovaných multicastových adres v rozsahu IEEE 802.1D[2]. Multicastové adresy

[1] LLC - Logical Link Control, https://standards.ieee.org/about/get/802/802.2.html
[2] https://standards.ieee.org/findstds/standard/802.1D-2004.html



Obrázek 3.1: Závislost mezi LLC entitou, LLDP agentem a MIB databází. Převzato z [3].

jsou vypsány v tabulce 3.1. Kromě toho lze využít i libovolnou skupinovou nebo individuální MAC adresu.

- Zdrojová adresa - jedná se o MAC adresu vysílací stanice nebo portu.
- Ethertype - pole využívané k identifikaci LLDP protokolu, obsahující hodnotu 88-CC (hexadecimálně).

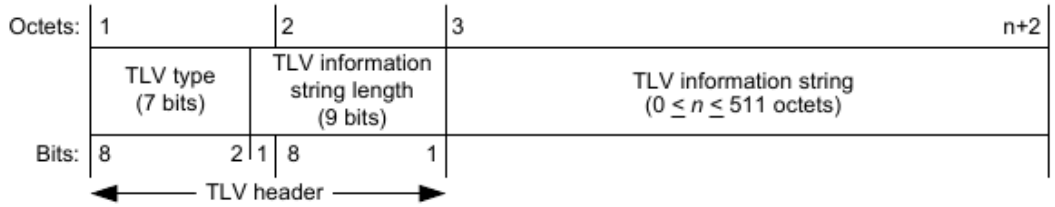
Zařízení si můžou vyměňovat LLDP zprávy i skrze jiné než ethernetové sítě. Například *Frame Relay*[3] sítě využívají SNAP hlavičky k identifikaci LLDP zpráv.

Název	Hodnota	Účel
Nejbližší most[4]	01-80-C2-00-00-0E	Propagace omezená na jedinou fyzickou linku. Zastavená všemi druhy mostů.
Nejbližší ne-TPMR most	01-80-C2-00-00-03	Propagace omezená všemi mosty jinými než TPMR[5]. Určené pro použití v rámci přemostěných sítí poskytovatele.
Nejbližší zákaznický most	01-80-C2-00-00-00	Propagace omezená uživatelskými mosty.

Tabulka 3.1: MAC adresy používané v LLDP.

[3]Frame Relay - https://cs.wikipedia.org/wiki/Frame_Relay

V každém LLDP rámci jsou informační pole obsažená v *Link Discovery Protocol Data Unit* (LLDPDU) jako sekvence různě dlouhých informačních elementů, kde každý obsahuje typ, délku a hodnotu (TLV). Příklad takového rámce je na obrázku 2.1. Každý LLDPDU rámec obsahuje tři povinné TLV následované libovolným počtem volitelných TLV. Obecný formát TLV v LLDP rámci je znázorněn na obrázku 3.2. Jednotlivé typy TLV jsou vypsány v následující tabulce 3.2 a blíže popsány v podkapitole 3.3.



Obrázek 3.2: Obecný formát TLV.

TLV typ	TLV název	Použití v LLDPDU
0	*End Of LLDPDU*	Volitelné
1	*Chassis ID*	Povinné
2	*Port ID*	Povinné
3	*Time To Live*	Povinné
4	*Port Description*	Volitelné
5	*System name*	Volitelné
6	*System description*	Volitelné
7	*System capabilities*	Volitelné
8	*Management Address*	Volitelné
9-126	Rezervace pro budoucí standardizaci	–
127	Organizačně specifické TLV	Volitelné

Tabulka 3.2: Typy TLV

3.2 Odesílání a přijímání LLDPDU

Každý LLDP agent může pracovat v jednom ze tří režimů:

- **pouze vysílání** - všechny příchozí LLDP rámce se ignorují. Pracuje pouze s místní MIB databází. Typicky se jedná o koncová zařízení, jako například IP telefony.
- **pouze příjem** - všechny přijaté rámce ukládá do vzdálené MIB databáze, ale žádné rámce neodesílá.
- **příjem i vysílání** - zařízení obsahuje jak místní, tak vzdálenou MIB databázi. Převážně všechny směrovače a přepínače pracují v tomto režimu.

[4]Myšleno zařízení spojující dvě části sítě na druhé OSI vrstvě
[5]TPMR - Two-Port MAC Relay, https://standards.ieee.org/findstds/standard/802.1AX-2014.html

3.2.1 Odesílání LLDPDU

K odeslání zprávy může dojít z jednoho z následujících tří důvodů:

(a) Pravidelný přenos na pozadí proběhne, jakmile vyprší časovač pro odesílání. Doporučený interval odesílání je 30 sekund, ale může nabývat hodnot v rozmezí 5 až 32768 sekund. Časování je řízené systémovým *tiky*, kdy každý *tik* je roven jedné sekundě. Nicméně na sdílených LAN médiích, z důvodu vyvarování se shlukování přenosů, interval mezi *tiky* obsahuje náhodnou rozprostřovací složku, která zajistí, že průměrný interval mezi *tiky* bude 1 sekunda, ale interval mezi dvojicemi sousedních *tiků* se bude lišit.

(b) Jestliže je rozpoznán nový soused (přijetím LLDP rámce na portu), v krátkých intervalech se odešle několik LLDPDU, aby bylo zajištěno, že soused je rychle informován aktuálními informacemi o svých sousedech. Jedná se o režim rychlého startu. Tento způsob je potlačen, pokud vzdálená MIB databáze není schopná zpracovat nové sousedovy informace bez smazání stále validních informací týkajících se staršího souseda. Může dojít k odeslání 1 až 8 rámců v intervalech nastaveném v rozmezí 1 až 3600 sekund. Je doporučeno odeslání tří rámců s rozestupem jedné sekundy.

(c) Pokud se změní stav nebo hodnota jednoho nebo více objektů v místní MIB databázi, je okamžitě odeslán jeden LLDPDU. Tímto je zajištěná okamžitá propagace lokálních změn sousedním uzlům.

Odesílání funguje na principu přidělování kreditů. Aby agent mohl odeslat LLDP rámec, potřebuje k tomu jeden kredit. Při každém systémovém *tik* se přidá každému agentu jeden kredit, pokud již počet jeho kreditů nedosáhl maximálního počtu. Maximální počet kreditů lze nastavit v rozmezí jednoho až desíti kreditů. Je doporučeno nastavit pět kreditů. Jakmile je kredit LLDP agenta vyčerpán, rychlost odesílání je omezená na odesílání maximálně jednoho rámce za sekundu. Toto umožňuje odeslání velkého množství LLDP rámců v krátkém čase, aniž by došlo k zahlcení linky, i když se vyskytne velké množství lokálních změn.

Pro sestavení rámce LLDP agent využívá údaje ze své místní MIB databáze. Z této databáze agent sestaví LLDPDU začínající povinnými položkami, následovanými volitelnými položkami. Položka TTL udává životnost odesílané informace. Pokud vzdálená strana po tuto dobu neobdrží aktualizaci informace, smaže ji ze své vzdálené MIB databáze. Doporučuje se nastavit tuto hodnotu na čtyřnásobek intervalu pravidelného odesílání LLDP zpráv, což dle doporučení odpovídá 120 sekundám. Může se ji ale nastavit jako 1 až 100 násobek pravidelného odesálání.

Pokud bylo LLDP na zařízení vypnuto a následně znova zapnuto, existuje časovač, který určuje, za jak dlouho může dojít k prvnímu přenosu LLDP rámce. Tento časovač může nabývat hodnot v rozmezí $1-10$ sekund, kdy ve výchozím nastavení jsou doporučeny dvě sekundy.

Jsou definovány dva typy LLDPDU:

(a) ***Normal*** - poskytuje informace o lokální stanici sousedním stanicím, s určitou dobou platnosti.

(b) ***Shutdown*** - označuje, že informace o lokální stanici spravována ve vzdálené MIB databázi souseda je již neplatná a musí být odstraněna. Tento LLDPDU se označí nastavením TTL hodnoty na hodnotu 0.

3.2.2 Příjem LLDPDU

Příjem LLDPDU se skládá ze tří fází:

- *rozpoznání rámce* - pomocí cílové adresy a hodnoty *Ethertype* se určí, zda je rámec určen pro LLDP agenta.
- *validace rámce* - rámec se rozdělí na jednotlivá TLV a ověří se, zda je korektně vytvořen a zda obsahuje správnou množinu povinných TLV ve správném tvaru a pořadí. Rovněž se v této fázi zkontroluje, zda jsou všechny volitelné TLV v korektním tvaru a zda některé TLV není obsaženo v rámci vícekrát, než je povoleno. Pokud se při validaci vyskytne chyba, podle typu chyby se pak zahodí buď celý rámec, nebo samotné chybné TLV.
- *aktualizace vzdálené MIB databáze* - rámec, který prošel validační fází, je použit spolu s informací, který agent rámec přijal, k aktualizaci údajů ve vzdálené MIB databázi. V databázi se objekt identifikuje pomocí *MAC service access point* (MSAP) identifikátoru, který se složí z přijatého LLDPDU jako konkatenace hodnoty TLV *chassis ID* a *port ID*.

Pokud se jedná o normální LLDPDU a záznam ještě neexistuje ve vzdálené MIB databázi, vytvoří se nový. Jestliže již existuje, informace obsažené v LLDPDU se využijí k aktualizaci údajů. Pokud objekt obsahoval údaje, které nebyly obsažené v LLDPDU, jsou tyto údaje smazány.

V případě *shutdown* LLDPDU rámce se záznam existující ve vzdálené MIB databázi začne považovat za neplatný a odstraní se.

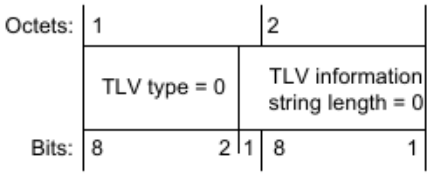
3.3 Formát TLV

Pokud u jednotlivých typů TLV není uvedeno jinak, nesmí se vyskytovat více než jednou v LLDPDU rámci.

3.3.1 End Of LLDPDU

End of LLDPDU je TLV o velikosti dvou oktetů, označující konec TLV sekvence v LLDPDU. Znázorněn je na obrázku 3.3. Obě části jsou vždy nastavené na hodnotu 0.

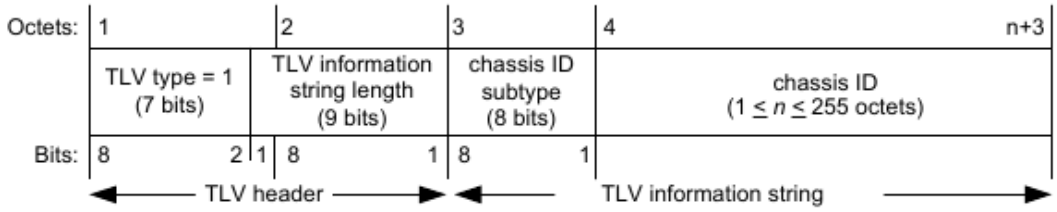
Některé IEEE 802 MAC požadují, aby velikost dat v rámci byla minimálně určitý počet oktetů. V případě že by rámec byl menší, dle IEEE 802 MAC by musel být doplněn nespecifikovanými daty do minimální velikosti. Nespecifikovaná data by ale mohla být na druhé straně rozpoznána jako určitý typ TLV. Pro tento účel slouží tento typ TLV, který jednoznačně říká, kde končí TLV posloupnost.



Obrázek 3.3: Formát *End Of LLDPDU* TLV.

3.3.2 Chassis ID

Jedná se o povinné TLV, které jednoznačně identifikuje šasi obsahující IEEE 802 LAN stanici, spjatou s vysílacím LLDP agentem. Formát TLV je na obrázku 3.4. Každé LLDPDU musí obsahovat právě jednu *Chassis ID* TLV, která je první v TLV posloupnosti.



Obrázek 3.4: Formát *chassis ID* TLV.

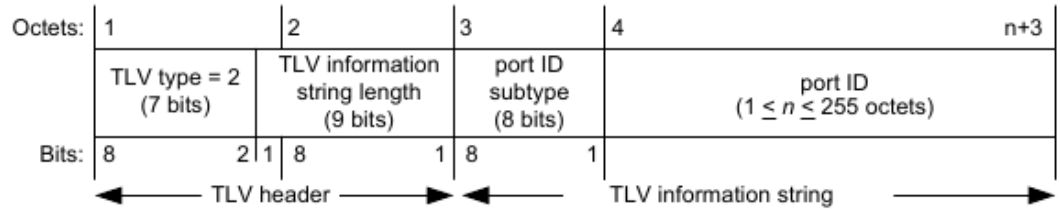
Položka *length* obsahuje velikost v oktetech následujících dvou položek (*chassis ID subtype* a *chassis ID*). Existuje několik způsobů identifikace šasi, k čemuž slouží pole *subtype*. Jednotlivé způsoby jsou vypsané v tabulce 3.3.

ID podtypu	Popis
0	Rezervované
1	Komponenta šasi
2	Alias rozhraní
3	Komponenta portu
4	MAC adresa
5	Síťová adresa
6	Název rozhraní
7	Lokálně přiřazené
8-255	Rezervované

Tabulka 3.3: Výčet podtypů *Chassis ID* TLV.

3.3.3 Port ID

Druhou položkou v TLV posloupnosti musí být *Port ID*, identifikující port zařízení, spjatý s vysílacím LLDP agentem. Identifikace musí být jednoznačná (například MAC adresa). Nikde jinde v posloupnosti se nesmí vyskytovat. Obrázek 3.5 ukazuje formát.



Obrázek 3.5: Formát *port ID* TLV.

Položka *length* musí obsahovat délku následujících dvou položek v oktetech. Stejně jak u *chassis ID*, existuje několik způsobů identifikace portu. V *port ID subtype* je číselně

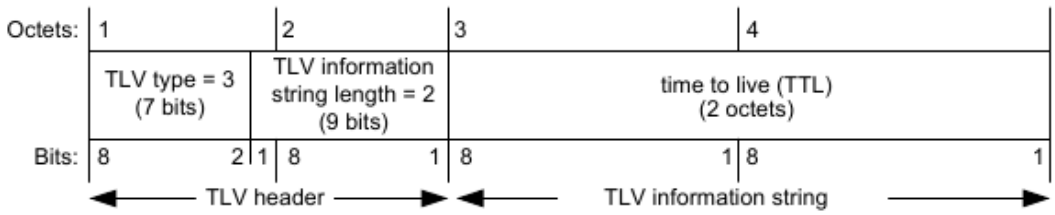
uveden způsob identifikace dle tabulky 3.4. Samotná identifikace portu je uvedena v *port ID* pomocí alfanumerické posloupnosti.

ID podtypu	Popis
0	Rezervované LLDPDU
1	Alias rozhraní
2	Komponenta portu
3	MAC adresa
4	Síťová adresa
5	Název rozhraní
6	*Agent circuit ID*
7	Lokálně přiřazené
8-255	Rezervované

Tabulka 3.4: Výčet podtypů *port ID* TLV.

3.3.4 Time To Live TLV

TTL TLV indikuje počet sekund, po které je přijímající LLDP agent povinen udržovat přijatou informaci jako validní. Pokud se jedná o nenulovou hodnotu, LLDP agent nahradí všechny informace spojené s daným MSAP identifikátorem, informacemi přijatými v LLDPDU. Když se *TTL* hodnota rovná 0, jedná se o *shutdown* LLDPDU rámec informující, že všechny údaje spojené s daným MSAP mají být odstraněny. Jedná se o třetí a poslední povinnou položku LLDPDU rámce s formátem zobrazeným na obrázku 3.6. V rámci se musí vyskytovat právě jednou.



Obrázek 3.6: Formát *Time To Live* TLV.

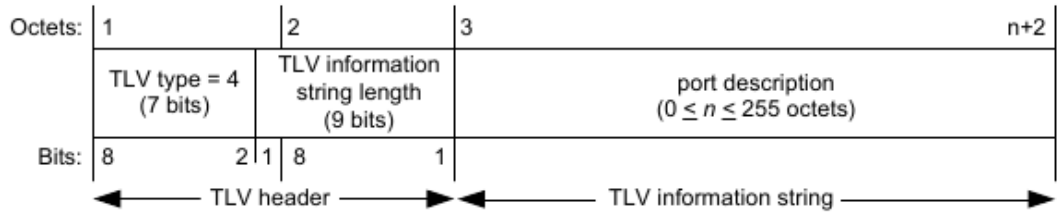
Počet sekund uvedených v poli *time to live* musí být v rozsahu 0 až 65535.

3.3.5 Port Description

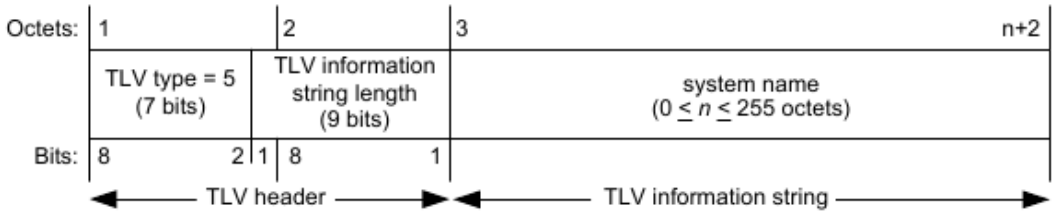
Díky tomuhle TLV lze oznamovat IEEE 802 LAN popis portu stanice. Neměl by se v rámci vyskytovat vícekrát než jednou. TLV formát je zobrazen na obrázku 3.7.

3.3.6 System name

V tomto TLV je uveden název systému, který je nastaven administrátorem. Formát je znázorněn na obrázku 3.8.



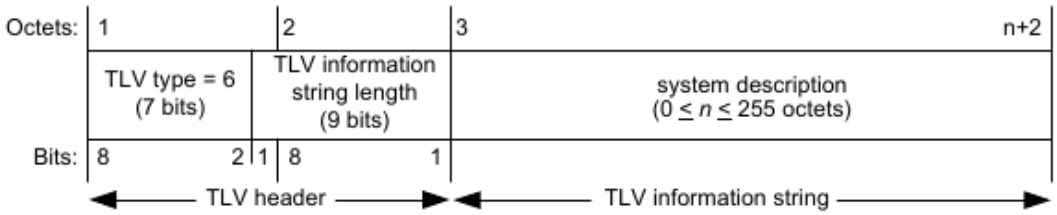
Obrázek 3.7: Formát *port description* TLV.



Obrázek 3.8: Formát *system name* TLV.

3.3.7 System description

Je v něm uveden alfanumerický popis síťové entity. Popis by měl obsahovat celý název a verzi hardwaru, operačního systému a síťového softwaru. Na obrázku 3.9 je zobrazen formát.



Obrázek 3.9: Formát *system description* TLV.

3.3.8 System capabilities

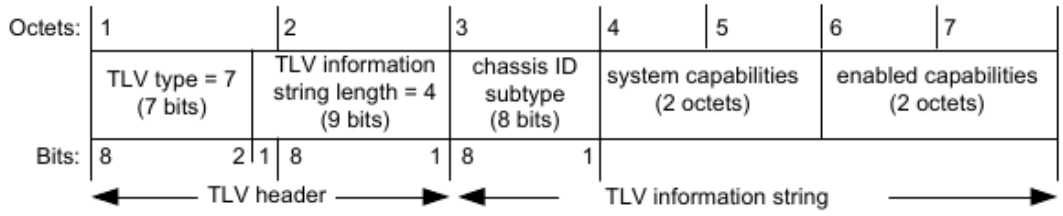
Jedná se o bitovou mapu schopností zařízení, které definují hlavní funkce systému. Rovněž definuje, které funkce jsou zapnuté. Jednotlivé položky jsou zobrazeny na obrázku 3.10.

Bitová mapa schopností zařízení je uložena v položce *system capabilities*. Poloha bitu pro každou jednotlivou funkci je znázorněna v tabulce 3.5. Tato poloha ale není garantována. Stejný princip platí i pro zapnuté schopnosti systému *enabled capabilities*. Pokud je v položce *enabled capabilities* označená nějaká schopnost systému jako zapnutá, ale není v položce *system capabilities*, TLV se interpretuje jako chybné.

3.3.9 Management address

Management address identifikuje adresu spjatou s lokálním LLDP agentem, která může být využitá pro přístup k entitám vyšších vrstev. TLV rovněž obsahuje prostor pro číslo

[6]Customer-VLAN a Service-VLAN se používá v 802.1Q tunelování. Jedná se o VLAN tagy využívané uživatelem a poskytovatelem služeb na svých zařízeních.

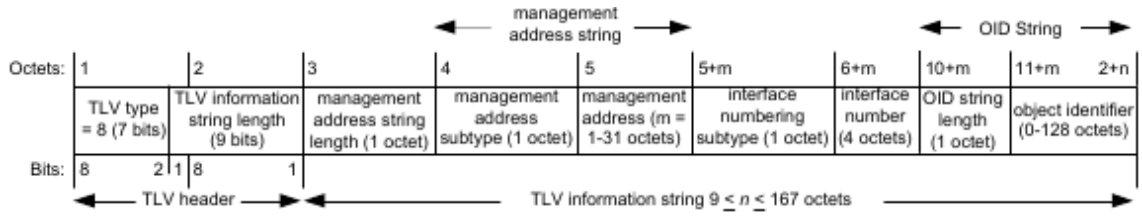


Obrázek 3.10: Formát *system capabilities* TLV.

Bit	Schopnosti
1	Jiné
2	Opakovač
3	MAC most
4	WLAN přístupový bod
5	Směrovač
6	Telefon
7	DOCSIS drátové zařízení
8	Pouze stanice
9	C-VLAN[6] komponenta VLAN mostu
10	S-VLAN[6] komponenta VLAN mostu
11	Two-port MAC Relay (TPMR)
12-16	Rezervované

Tabulka 3.5: Význam bitů bitové mapy v *system capabilities*.

rozhraní a identifikátor objektu (OID), který je spjat s management adresou, pokud je alespoň jeden z nich znám. Formát je znázorněn na obrázku 3.11.



Obrázek 3.11: Formát *management address* TLV.

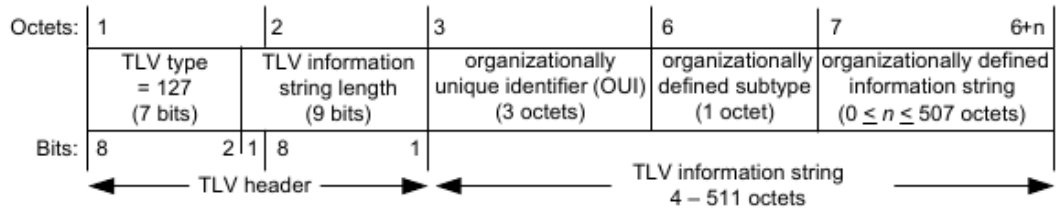
V položce *management address* by měla být adresa, která se nejvíce hodí pro správu, typicky adresa 3. vrstvy, jako například IPv4 adresa. Pokud není žádná taková adresa dostupná, měla by být vložena alespoň MAC adresa zařízení nebo portu.

Toto TLV se může vyskytovat v rámci vícekrát. Je doporučeno, aby se v každém rámci vyskytovalo alespoň jednou.

3.3.10 Formát speciálních TLV

Tato kategorie TLV byla vytvořena z důvodu, aby různé organizace jako IEEE nebo IETF, ale i různý software a zařízení výrobců, mohly definovat své vlastní TLV. Musí se ale držet již definovaných pravidel pro základní množinu TLV. Mohou být použity pro jednosměrnou

distribuci informací ostatním LAN stanicím, které musí být nezávislé na informacích přijatých od jiných vzdálených agentů. Informace šířené přes specifické TLV nesmí přesahovat skrz několik TLV a nesmí být na vyžádání přeposílány na další porty přijímající stanice. Základní formát specifického TLV je na obrázku 3.12.



Obrázek 3.12: Formát speciálních TLV.

Organizace IEEE v 802.1 definovala několik specifických TLV, sloužících hlavně pro práci s VLAN sítěmi. Jedná se mezi jinými o:

- **Port VLAN ID** - přenáší VLAN ID portu.
- **VLAN Name** - umožňuje zařízení šířit textový název kterékoli VLAN, kterou je konfigurované.
- **Protocol Identity** - přenáší typy podporovaných protokolů.

3.4 LLDP-MED

Media Endpoint Discovery (MED) je rozšíření k základnímu LLDP protokolu, které poskytuje podporu pro VoIP[7]. Rozšíření je definováno standardem vyvinutým TIA[8] a publikovaným v ANSI/TIA-1057. LLDP-MED zjednodušuje nasazování VoIP zařízení do IEEE 802 LAN prostředí. Jelikož se jedná o publikovaný standard, VoIP zařízení různých výrobců spolu umí komunikovat.

LLDP-MED poskytuje následující výhody:

- automatické detekování síťových politik (VLAN, 802.1Q, DSCP)
- rozšířená a automatická správa napájení PoE[9] koncových zařízení
- poskytuje administrátorům možnost sledovat síťová zařízení a zjišťovat jejich charakteristiky (název výrobce, verzi softwaru a hardwaru, sériové číslo)
- zjišťování umístění zařízení z důvodu vytvoření lokalizační databáze, a v případě VoIP služby tísňového volání

3.5 Konfigurace

Konfigurační příkazy a případné výstupy těchto příkazů v této a následující kapitole, jsou z Cisco směrovače s IOS verze 15.4(2)T4.

[7] VoIP - Voice over Internet Protocol, https://cs.wikipedia.org/wiki/Voice_over_Internet_Protocol
[8] TIA - Telecommunications Industry Association, http://www.tiaonline.org
[9] PoE - Power over Ethernet, https://cs.wikipedia.org/wiki/Power_over_Ethernet

Pro globální zapnutí LLDP slouží příkaz:

```
Router(config)# lldp run
```

Vysílání a přijímání rámců na specifickém rozhraní se zapíná příkazy:

```
Router(config-if)# lldp transmit
Router(config-if)# lldp recieve
```

Vypnutí LLDP lze provést ekvivalentně, pouze přidáním slovíčka **no** před příkaz:

```
Router(config)# no lldp run
Router(config-if)# no lldp transmit
Router(config-if)# no lldp recieve
```

Nastavení doby, jak dlouho bude zařízení uchovávat informaci před smazáním, inicializační zpoždění a frekvenci odesílání aktualizací, lze nastavit následujícími příkazy:

```
Router(config)# lldp holdtime seconds
Router(config)# lldp reinit seconds
Router(config)# lldp timer seconds
```

3.6 Ověření konfigurace

Vypsat globální informace, jako frekvence odesílání, *holdtime* a inicializační zpoždění, lze příkazem:

```
Router# show lldp
Global LLDP Information:
    Status: ACTIVE
    LLDP advertisements are sent every 30 seconds
    LLDP hold time advertised is 120 seconds
    LLDP interface reinitialisation delay is 2 seconds
```

Příkladem níže se zobrazují informace o konkrétním sousedovi, kdy místo znaku * lze zadat název konkrétního souseda (název získaný ze *system name TLV*), jehož informace se chce vypsat:

Router# show lldp entry *
Capability codes:
(R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
(W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Chassis id: 0007.7deb.18e0
Port id: Gi0/1
Port Description: GigabitEthernet0/1
System Name: Router

System Description:
Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version 15.3(3)M6, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Tue 04-Aug-15 03:59 by prod_rel_team

Time remaining: 93 seconds
System Capabilities: B,R
Enabled Capabilities: R
Management Addresses:
IP: 10.0.0.2
Auto Negotiation - not supported
Physical media capabilities - not advertised
Media Attachment Unit type - not advertised
Vlan ID: - not advertised

Total entries displayed: 1

Zobrazit informace o sousedech lze příkazem:

Router# show lldp neighbors
Capability Codes:
(R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
(W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID	Local Intrfce	Hold-time	Capability	Port ID
Nortel IP Phone	Gi1/0/1	180	T	0019.e1e7.018d
Polycom SoundPoint	Gi1/0/19	180	T	0004.f22f.88b7
Baseline Switch 1	Gi1/0/18	180	P,B	Ethernet0/26

Total entries displayed: 4

Výpis sousedů lze omezit pouze na sousedy na konkrétním rozhraní. Vypsané informace lze rozšířit specifikováním parametru *detail*.

Zobrazit informace o LLDP provozu, obsahující počet přijatých a odeslaných rámců, počet zahozených rámců a počet nerozpoznaných TLV, lze příkazem:

```
Router# show lldp traffic
LLDP traffic statistics:
    Total frames out: 560
    Total entries aged: 0
    Total frames in: 211
    Total frames received in error: 0
    Total frames discarded: 0
    Total TLVs discarded: 208
    Total TLVs unrecognized: 208
```

Zobrazit stav počítadel LLDP chyb lze příkazem:

```
Router# show lldp errors
LLDP errors/overflows:
    Total memory allocation failures: 0
    Total encapsulation failures: 0
    Total input queue overflows: 0
    Total table overflows: 0
```

Ke smazání LLDP tabulek o sousedech slouží příkaz:

```
Router# clear lldp table
```

Zresetovat statistiky o provozu lze příkazem:

```
Router# clear lldp counters
```

Kapitola 4

Odstranění závislosti na modulu DeviceConfigurator

Jedním z cílů této práce bylo odstranění závislosti na modulu `DeviceConfigurator` [18], čemuž se věnuje tato kapitola. Jedná se o modul, který byl vytvořen v rámci projektu ANSA [5]. Projekt ANSA je rozšířením frameworku INET [11], který má veškerou přidanou funkcionalitu ve složce `src/ansa/`.

Modul byl primárně vytvořen ze dvou důvodů. Prvním důvodem bylo, že u každého zařízení existuje sada obecných příkazů, které nelze jednoznačně přiřadit konkrétnímu modulu. Jedná se především o parametry platné pro celé zařízení (*id*, *hostname*), ale také základní nastavení rozhraní a např. statického směrování. Druhým důvodem vytvoření tohoto modulu bylo mít možnost externě měnit parametry ostatních modulů.

Tento modul tedy umožňuje měnit výchozí nastavení všech modulů v zařízení, a to bez nutnosti zasahovat a měnit jejich kód. Moduly musí pouze poskytovat potřebné rozhraní. Je to tedy obzvlášť užitečné v případě modulů importovaných z INET, jejichž úpravy nejsou žádoucí. Když se modul spojí s knihovnou `xmlParser`, vznikne tak jednoduchý nástroj, s nímž lze konfigurovat libovolné zařízení XML souborem.

Jak se ale ANSA projekt rozšiřoval víc a víc, začaly být neúnosné překlady knihovny. Jelikož velká část modulů obsahovala závislost na tomto modulu, při sebemenší změně bylo potřeba překládat celý projekt ANSA. Bylo tedy rozhodnuto, že se odstraní závislosti na tomto modulu.

Cílem bylo jej úplně odstranit z ANSA projektu. V každého modulu, který na něj obsahoval závislost, vytvořit novou třídu, jež bude obsahovat stejnou funkcionalitu. Tato třída bude mít v sobě i funkčnost z knihovny `xmlParser`.

Jako referenční verze ANSA byla použita verze 2.2[1]. Všechny vytvořené změny byly nahrány do nové větve `del-deviceConf`[2] na serveru GitHub.

V každém modulu, který obsahoval závislosti na tomto modulu, byla vytvořená nová třída. Vždycky byla snaha o zachování stejné sémantiky pojmenovávání souborů, jaká byla zvolena tvůrcem daného modulu. Například tedy v modulu `Babel` byla nová třída pojmenována `BabelDeviceConfigurator.cc|h|ned`.

Jelikož modul sloužil pro konfiguraci mnoha modulů, obsahoval metody pro konfiguraci všech těchto modulů. Při rozkopírovávání do všech modulů bylo tedy potřeba v každé nově vytvořené třídě konfigurátoru odstranit všechny metody, které daný modul nevyužíval a

[1] https://github.com/kvetak/ANSA/tree/ansainet-2.2
[2] https://github.com/kvetak/ANSA/tree/ansainet-del_deviceConf

nebyly pro něj relevantní.

Rovněž z takto vytvořených tříd byly odstraněny závislosti na jiných modulech, které daný modul nepotřeboval, ale potřeboval jen závislosti jiného modulu. Jednalo se například o odstranění závislosti z modulu `Babel` na modulu `RIPngRouting`. Závislost byla vytvořena jen z důvodu, že modul `RIPngRouting` obsahoval závislosti na knihovně `ANSARoutingTable6Access` a `algorithm`.

Kapitola 5

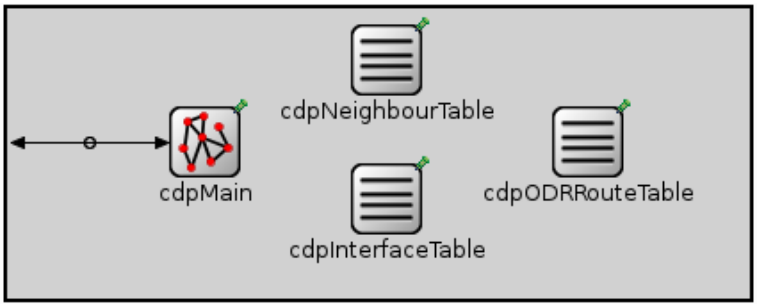
Návrh a implementace CDP

V této kapitole je popsán návrh a implementace CDP podle teoretických základů z kapitoly 2. Návrh byl vcelku přímočarý, proto tato kapitola slučuje návrh s implementací. Jsou zde popsány vytvořené struktury a jejich funkčnost. Diagram tříd se nachází v příloze C.3.

Protokol je implementován v knihovně INET pro simulační nástroj OMNeT++[17]. Programovacím jazykem je C++. V knihovně INET bylo potřeba provést několik úprav. Do tříd `IPv4Route` a `IRoute` byly doplněny informace, aby zdroj síťové cesty mohl být nastaven na ODR. Poslední změnou v knihovně byla oprava případu, kdy dojde k přerušení spojení mezi zařízeními. Když pak na takové rozhraní přišel paket, simulace se ukončila chybou. Chyba již byla v knihovně zaznamenána a popsána její případná oprava. Nacházela se ve třídě `MACBase`, a tak byla opravena.

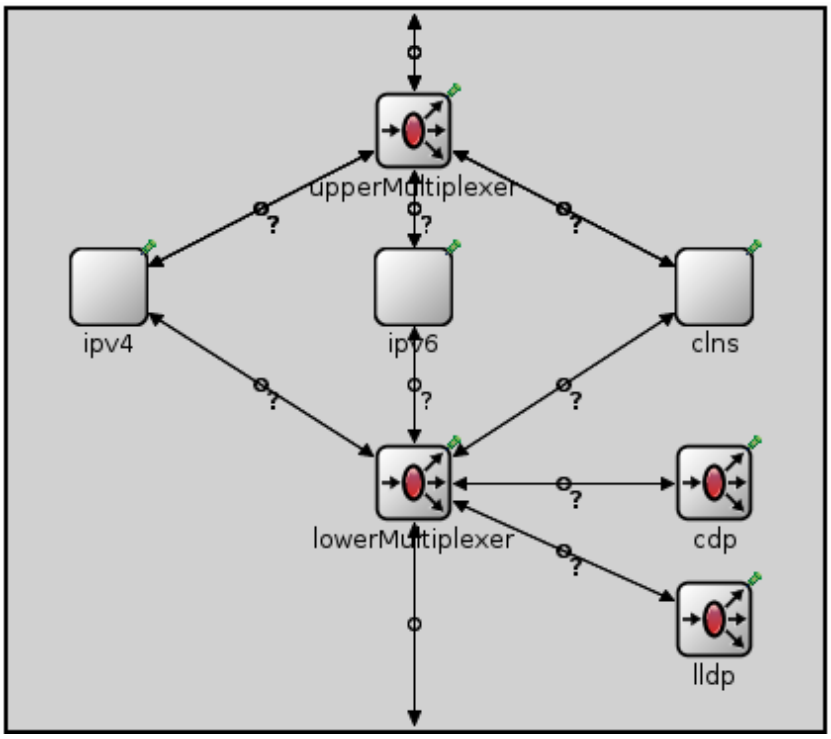
5.1 Modul protokolu

Jedná se o složený modul. Obrázek 5.1 zobrazuje jeho vnitřní strukturu. O celou funkcionalitu se stará podmodul `CDPMain`. Pro uchování dat pak slouží tři tabulky.



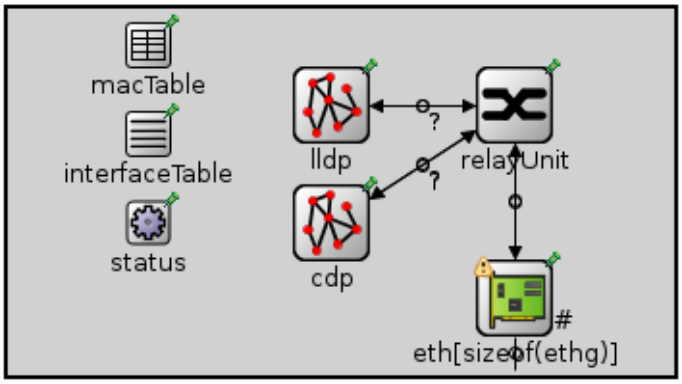
Obrázek 5.1: Vnitřní struktura CDP modulu.

Aby byla zachována podpora všech ANSA rozšíření, jako referenční směrovač byl použit `ANSA_Router` a vytvořen `ANSA_LLDPCDPRouter`. Od referenčního se liší pouze přidáním několika parametrů. Modul CDP je vložen do `ANSA_MultiNetworkLayer`. Přesné umístění je na obrázku 5.2. Modul byl poupraven tak, aby uměl rozpoznat CDP a LLDP rámce a přeposlat je CDP nebo LLDP modulu. Propojení s tímto modulem je skrze brány *ifIn* a *ifOut*.



Obrázek 5.2: Modul ANSA_MultiNetworkLayer.

V rámci práce byl vytvořen i přepínač s názvem `ANSA_EtherSwitch`. Pro jeho vytvoření byl použit přepínač z knihovny INET `EtherSwitch`, ze kterého byl smazán `STP` modul. Umístění modulu je ukázáno na obrázku 5.3. Modul `relayUnit` byl vytvořen zkopírováním `Ieee8021dRelay` a rozšířen o schopnost rozpoznání CDP a LLDP rámců. Rozpoznávání probíhá dle cílové MAC adresy, která je pro CDP a LLDP jedinečná.



Obrázek 5.3: Modul ANSA_CDPEtherSwitch.

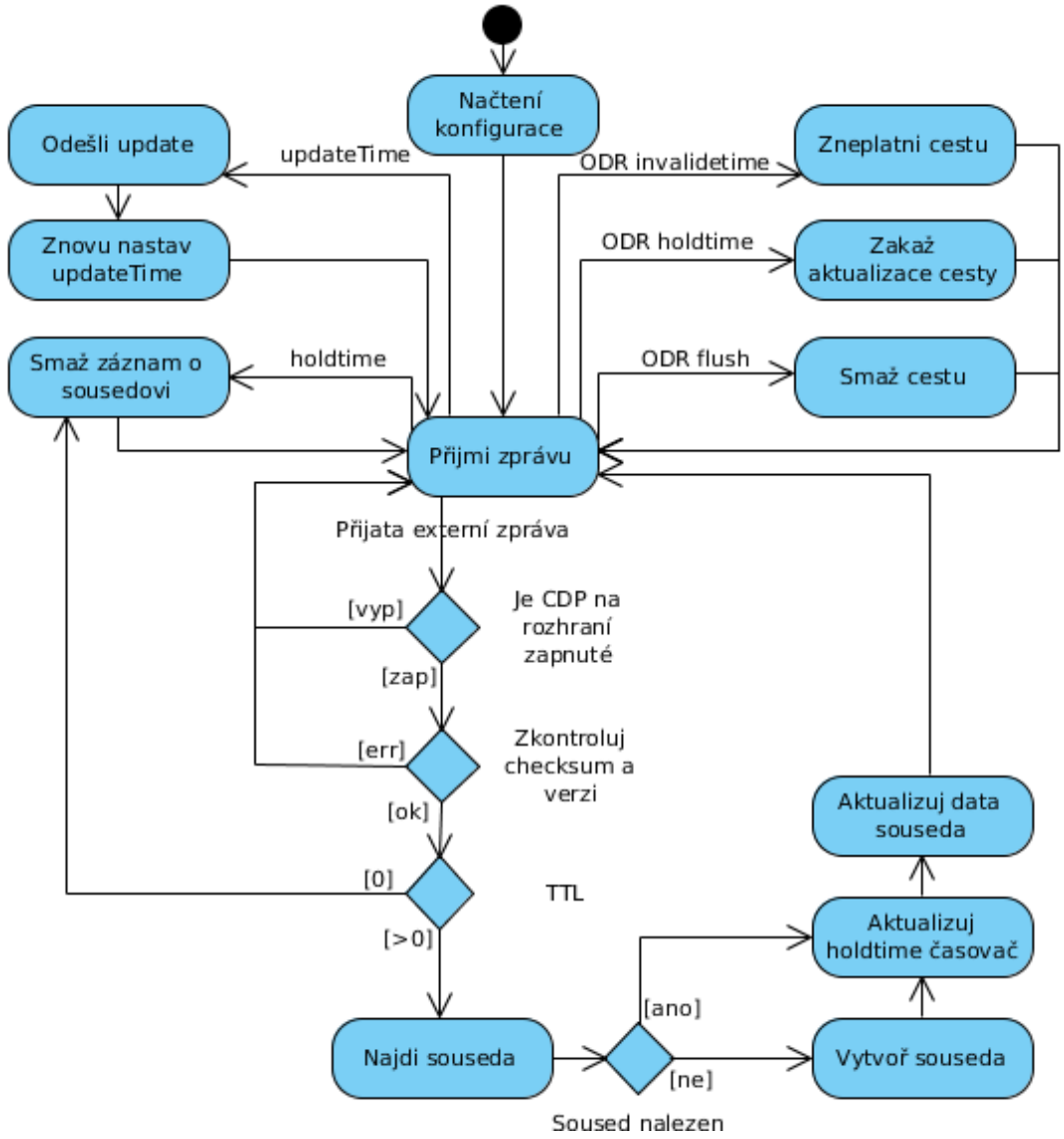
5.2 Modul CDPMain

Tento modul má na starosti základní funkcionalitu. Jeho činnost po spuštění je zobrazena na diagramu 5.4. Tento diagram zachycuje pouze zásadní chování modulu.

Po spuštění simulace nejprve načte parametry definované v *CDP.ned* souboru a ověří

jejich hodnoty. Rovněž získá odkazy na tabulky dat `CDPNeighbourTable`, `CDPInterfaceTable`, `CDPODRRouteTable`, `IInterfaceTable` a případně (pokud se jedná o zařízení pracující na síťové vrstvě) `IRoutingTable`.

V dalším kroku inicializace se modul zaregistruje pro odběr notifikací informujících o změně stavu rozhraní, vytvoření rozhraní, smazání rozhraní a smazání cesty ze směrovací tabulky. Pro zpracování notifikací slouží povinná metoda `receiveChangeNotification()`. V tomto kroku rovněž načte konfigurace ze souboru XML, k čemuž slouží třída `CDPDeviceConfigurator` (popsána v kapitole 5.5). Na každém rozhraní, na kterém má CDP odesílat a přijímat oznámení, spustí časovače *updateTime* pro jejich periodické zasílání.



Obrázek 5.4: Diagram chování modulu CDPMain.

Následně modul očekává příjem zprávy. Každá zpráva je přijatá metodou `handleMessage()`, která zjistí, zda se jedná o vlastní zprávu (časovač), nebo o externí zprávu, která přišla z jiného zařízení. O zpracování vlastních zpráv se stará metoda `handleTimer()` a o zprávy metoda `handleUpdate()`. Struktura zpráv je blíže popsaná v kapitole 5.4.

Metoda `handleTimer()` zjistí, k vypršení kterého časovače došlo. Existuje pět následujících časovačů:

- *CDPUpdateTime* - časovač informující, že má dojít k zaslání CDP oznámení na konkrétní rozhraní. Při posílání oznámení se zavolá metoda `sendUpdate()` a znovu nastaví časovač, pro další odeslání oznámení. Tato metoda sestrojí celý CDP rámec a odešle ho na rozhraní, které má specifikované v parametru.
- *CDPHoldtime* - došlo k expiraci údaje v tabulce sousedů `CDPNeighbourTable`. Příslušný záznam o sousedovi se tedy smaže.
- *CDPODRInvalideTime* - vypršela platnost cesty v ODR směrovací tabulce `CDPODRRouteTable`. Cesta se smaže z hlavní směrovací tabulky zařízení, označí jako neplatná a zakáže její případné aktualizace.
- *CDPODRHolddown* - ODR cesta se znovu označí jako aktualizovatelná.
- *CDPODRFlush* - platnost cesty v tabulce `CDPODRRouteTable` vypršela, a tak dojde k jejímu smazání.

K odeslání aktualizace může ještě dojít i v případě, kdy rozhraní nebo zařízení zjistí, že bude vypnuté (modul zachytí jednu z notifikací, k jejichž odběru je přihlášený). V tomto případě se odešle aktualizace, která má jako *TTL* nastavenou hodnotu 0. K odeslání slouží metoda `sendUpdate()`, která takovou zprávu zkonstruuje a odešle. Žel, tato vlastnost ještě nefunguje, jelikož v INETu se rámec zahodí na výstupním rozhraní, které je už v době obdržení rámce z vyšších vrstev vypnuté.

Metoda `handleUpdate()` má na starosti zpracování zpráv, které přišly z vnějšku. Celý tento proces je znázorněn na sekvenčním diagramu v příloze C.1. Ověří, zda zpráva obsahuje alespoň nějaké TLV, zkontroluje zda je CDP na příchozím rozhraní zapnuté, spočte a ověří kontrolní součet a ověří verzi CDP. Jestliže všechno odpovídá, zpráva se předá metodě `neighbourUpdate()`, jinak se zahodí.

Aktualizace údajů souseda probíhá v metodě `neighbourUpdate()`. Jestliže *TTL* zprávy je 0, příslušný soused se smaže z tabulky sousedů, v opačném případě se jej vyhledá v tabulce. Pokud takový soused neexistuje, vytvoří se a přidá do tabulky. Záznam o sousedovi se pak aktualizuje podle informací získaných ze zprávy.

Pokud je ODR zapnuté a při aktualizaci se zjistí, že do stejné sítě existuje více než N cest, je cesta zahozena. N je ve výchozím nastavení nastaveno na hodnotu 4. Tuto hodnotu lze změnit pomocí proměnné `maxDestinationPaths` v `CDP` modulu.

Jakmile dojde k ukončení simulace, modul vypíše statistiky ohledně posílání CDP paketů. Jedná se například o statistiky, kolik paketů které verze přijal nebo kolik přijal paketů s chybnou hlavičkou.

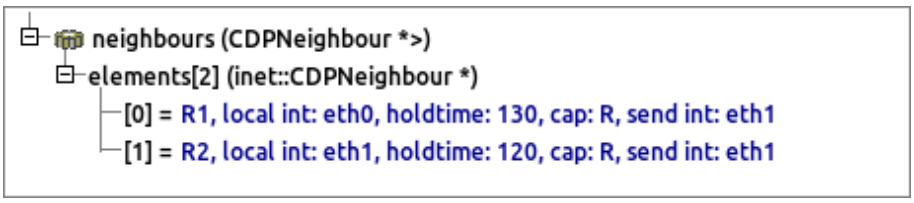
5.3 Třídy pro uchování dat

V modulu `CDP` se nachází tři tabulky, které slouží pro uchování dat naučených od sousedů nebo dat načtených během konfigurace.

5.3.1 Tabulka sousedů CDPNeighbourTable

Tato tabulka uchovává informace o přímo připojených zařízeních se zapnutým CDP. Jednotlivé položky tabulky jsou tvořeny třídou `CDPNeighbour`. Tato třída obsahuje informace o sousedovi a také nastavený časovač *holdtime*.

Nad tabulkou lze volat základní metody pro přidání, smazání a vyhledání souseda. Kromě toho existuje i metoda pro spočtení počtu naučených sousedů z konkrétního portu. Obsah této tabulky lze zobrazit během simulace, viz obr. 5.5.

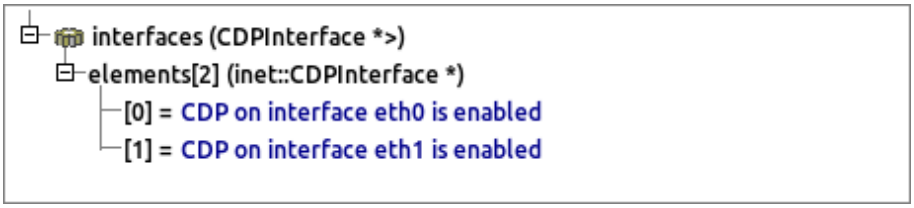


Obrázek 5.5: Textová podoba CDP tabulky sousedů.

5.3.2 Tabulka rozhraní CDPInterfaceTable

V této tabulce se nachází informace o všech rozhraních zařízení, na kterých může běžet CDP. Záznamy v tabulce jsou tvořeny třídou `CDPInterface`. Tato třída obsahuje odkaz na rozhraní, časovač *updateTime* zasílání oznámení, informaci o stavu CDP a čítač rychlého startu CDP. Stav CDP na rozhraní lze nastavit pouze pomocí `CDPDeviceConfigurator` popsaného v kapitole 5.5.

Stejně jak v minulém případě, tabulka obsahuje základní metody pro přidávání, mazání a vyhledávání rozhraní. Během simulace lze tabulku zobrazit, viz obr. 5.6.



Obrázek 5.6: Textová podoba CDP tabulky rozhraní.

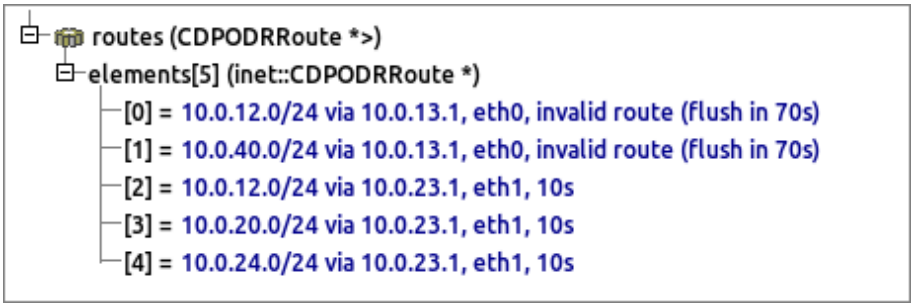
5.3.3 Tabulka rozhraní CDPODRRouteTable

Tabulka obsahuje síťové cesty, které se zařízení naučilo skrze ODR. Položky tabulky jsou tvořeny třídou `CDPODRRoute`, která obsahuje informace o cestě a tři základní časovače sloužící pro správu cesty.

Tabulka rovněž obsahuje metody pro přidávání, mazání a vyhledávání cest. K tomu ještě existuje metoda `countDestinationPaths()`, která spočte počet různých cest do určité sítě. Obsah tabulky během simulace lze vidět na obrázku 5.7.

5.4 CDP zprávy

Zprávy, které zpracovává modul `CDPMain`, se dělí na dva typy. Prvním typem jsou vlastní zprávy, což jsou zprávy časovačů. Druhým typem jsou externí zprávy.



Obrázek 5.7: Textová podoba ODR směrovací tabulky.

5.4.1 Vlastní zprávy

Jedná se o zprávy, které modul poslal sám sobě s určitým zpožděním. Tímto se v diskrétním časovači modulují časovače. Tyto zprávy jsou definovány v souboru *CDPTimer.msg*, z něhož OMNeT++ automaticky vygeneruje odpovídající třídu.

Zpráva obsahuje proměnnou *timerType* typu *char*, která může nabývat hodnot z *CDPTimerType* enumeration seznamu. Podle této hodnoty se zjistí, k vypršení kterého časovače došlo. Časovače *CDPHoldtime*, *CDPODRInvalideTime*, *CDPODRHolddown* a *CDPODRFlush* jsou vázané na konkrétní záznam v tabulce. Při doručení zprávy tak není potřeba procházet všechny záznamy tabulky, aby se zjistilo, ke kterému záznamu tento časovač patří.

5.4.2 Externí zprávy

Jedná se o CDP oznámení z jiného zařízení definované v souboru *CDPUpdate.msg*. Struktura těchto zpráv je následující:

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

  char        version = 2;
  unsigned char ttl;
  uint16_t     checksum;
  TLVOptions    options;
}
```

Na rozdíl od vlastních zpráv, vygenerování odpovídající třídy nemůže probíhat zcela automaticky, ale je dodefinováno v třídě `CDPUpdate`. Je to způsobené tím, že zpráva obsahuje různé typy TLV položek, kdy každý typ je jiná třída a všechny se nachází ve stejném poli. Pro tvorbu TLV pole, byla použitá třída definována v INET `TLVOptions`. Tato třída definuje hlavičku všech TLV a obsahuje základní metody pro práci s nimi. Všechny TLV položky pak dědí od společného rodiče `TLVOptionBase`. Příkladem takového TLV může být TLV pro přenos prefixů:

```
class prefixType
{
  uint32_t network;
  uint8_t  mask;
```

```
};

class CDPOptionPref extends TLVOptionBase
{
  type = CDPTLV_IP_PREF;
  prefixType  prefixes[];
}
```

Zapouzdření těchto zpráv se ale liší od zpráv reálné sítě. V reálné síti jsou zprávy zapouzdřeny do *Ethernet* rámců se SNAP hlavičkou. Tato implementace zapouzdřuje zprávy do *Ethernet II*[1] rámců. Toto řešení bylo zvolené z důvodu, že INET zařízení zapouzdřují zprávy do rámce, který byl specifikován při jeho inicializaci. Nelze tedy určit různé zapouzdřování pro různé zprávy. Jelikož všechny protokoly v INET knihovně používají zapouzdřování do *Ethernet II* rámců, z důvodu kompatibility takto zapouzdřuje i CDP. Až bude možné specifikovat typ zapouzdření pro konkrétní zprávu, bylo by dobré změnit na zapouzdřování se SNAP hlavičkou. Tato vlastnost nemá žádný vliv na funkčnost CDP protokolu.

5.5 Třída CDPDeviceConfigurator

Jak již bylo zmíněno, tato třída slouží k načtení konfigurace CDP a ODR ze souboru XML. Jako výchozí nastavení CDP a ODR se primárně bere hodnoty nastavené v souboru *CDP.ned*. Tato třída umí tyto hodnoty přepsat, nebo specifikovat hodnoty na jednotlivých rozhraních zařízení dle hodnot v XML souboru.

K získání nastavení ze souboru slouží metoda `loadCDPConfig()`. Metoda nejprve zavolá metodu `loadODRProcessConfig()`, která načte konfiguraci ODR. Konkrétně se jedná o možnost načtení konfigurace pro všechny tři časovače ODR. Následně se metodou `loadCDPInterfacesConfig()` načte CDP nastavení na jednotlivých rozhraních. Na rozhraní lze pouze nastavit, zda CDP na něm je zapnuté nebo vypnuté. Všechna výše zmíněná nastavení jsou nepovinná a nemusí být v XML souboru specifikovaná. Takový soubor může vypadat následovně:

```
<Router id="R3">
  <Interfaces>
    <Interface name="eth0">
      <CDP>
        <status>enabled</status>
      </CDP>
    </Interface>
  </Interfaces>
  <Routing>
    <ODR>
      <invalid>100</invalid>
      <holddown>30</holddown>
      <flush>200</flush>
    </ODR>
  </Routing>
</Router>
```

[1] Ethernet II - https://cs.wikipedia.org/wiki/Ethernet_II

Kapitola 6

Návrh a implementace LLDP

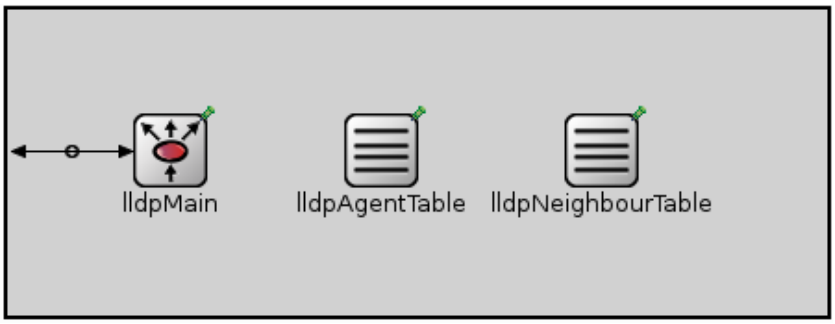
Kapitola obsahuje popis návrhu a implementace LLDP protokolu. Potřebné teoretické základy jsou čerpány z kapitoly 3. Jsou zde popsané hlavní vytvořené struktury a jejich funkčnost. Diagram tříd implementace se pak nachází v příloze D.3.

Protokol je implementován v jazyce C++ v knihovně INET. V této knihovně byla opravena jedna chyba. Konkrétně se jednalo o případ, kdy došlo k přerušení spojení mezi zařízeními, a na rozhraní přišel rámec z vyšších vrstev. Tato chyba již byla zaznamenána ve třídě `MACBase`, a tak ji stačilo opravit.

Jelikož protokol je specifikován standardem popisujícím jeho přesnou funkčnost, byla snaha zachovat názvy proměnných a metod. Do velké části tedy názvy proměnných a metod z této implementace a standardu odpovídají.

6.1 Modul protokolu

Protokol LLDP je implementován jako složený modul. Strukturu tohoto modulu lze vidět na obrázku 6.1. Modul se skládá ze tří jednoduchých modulů, a to konkrétně z modulu `LLDPMain` a dvou tabulek.

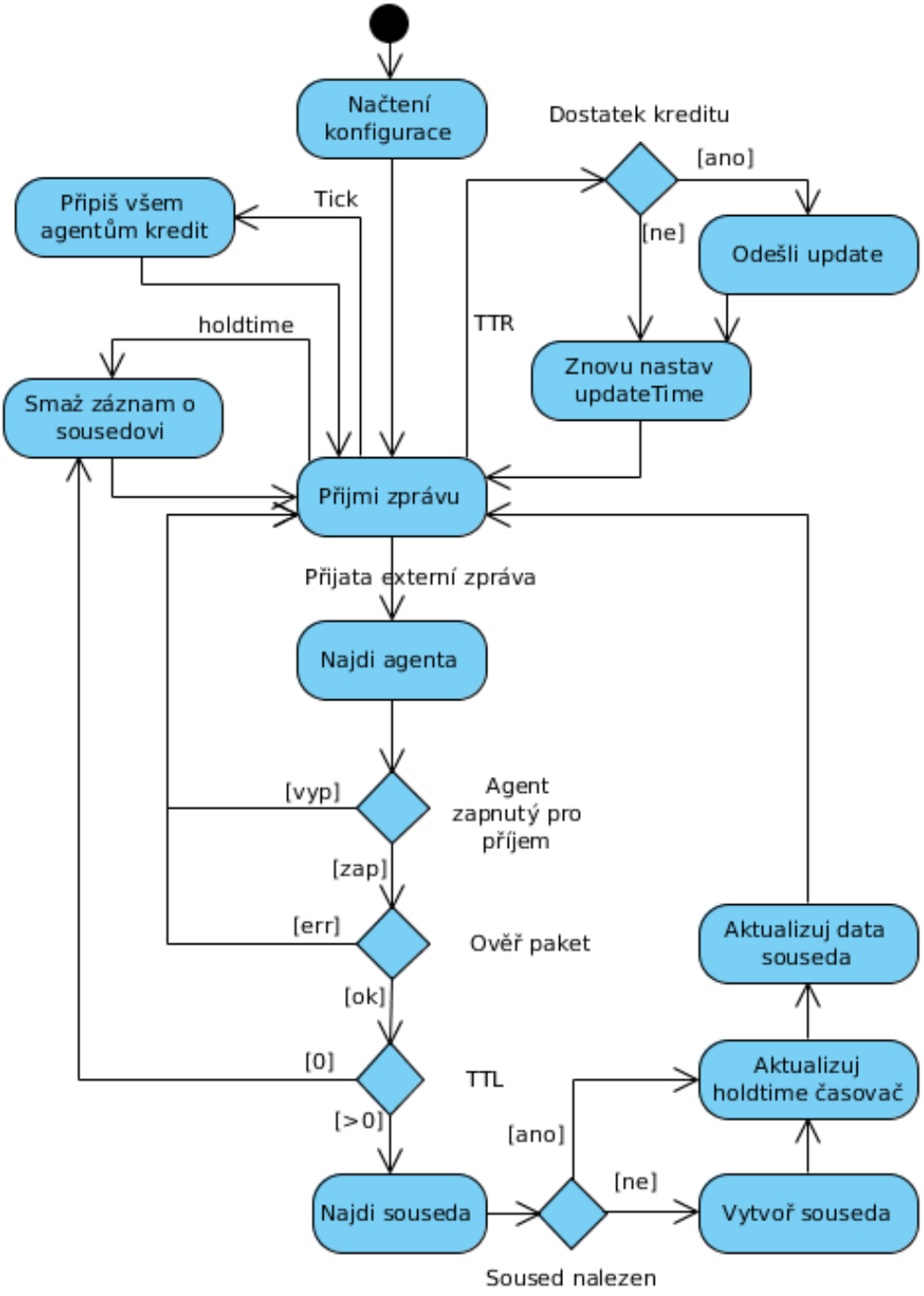


Obrázek 6.1: Vnitřní struktura LLDP modulu.

Tento modul byl vložen do vytvořeného směrovače `ANSA_LLDPCDPRouter` a přepínače `ANSA_EtherSwitch`, jež jsou popsány v kapitole 5.1.

6.2 Modul LLDPMain

Základní funkcionalitu obstarává právě tento modul. Chování modulu je zachyceno na obrázku 6.2



Obrázek 6.2: Diagram chování modulu LLDPMain.

Při zapnutí modul nejprve načte hodnoty proměnných ze souboru `LLDP.ned` a získá odkazy na tabulku `LLDPAgentTable`, `LLDPNeighbourTable` a `IInterfaceTable`. Následně zaregistruje pro odběr notifikací o změně stavu rozhraní, přidání rozhraní a smazání roz-

hraní. Rovněž vygeneruje své `chassis ID`. Jako `chassis ID` použije MAC adresu prvního rozhraní z tabulky `IInterfaceTable`, které ji má specifikovanou. Pokud nenajde žádné takové rozhraní, použije název zařízení, v němž se nachází. Následně pomocí třídy `LLDPDeviceConfigurator` (viz. 6.5) načte konfigurace jednotlivých rozhraní z XML souboru. V poslední části inicializace spustí všechny agenty.

Po inicializaci modul čeká na příchozí zprávy. O příjem všech zpráv se stará metoda `handleMessage()`. Pokud přišla vlastní zpráva, tak zavolá metodu `processTimer()`, a v případě příjmu externí zprávy `handleUpdate()`. Struktura zpráv je popsána v kapitole 6.4.

Metoda `processTimer()` rozpozná, k vypršení kterého časovače došlo. Existují tři typy časovačů:

- `Tick` - časovač informující o tom, že má být všem agentům přidělen jeden kredit.
- `TTR` - při vypršení časovače, dojde k odeslání pravidelné zprávy sousedovi. Instance tohoto časovače existuje v každém agentovi. Jakmile zpráva dorazí, zjistí se, od kterého agenta pochází a následně se nad tímto agentem zavolá metoda `txInfoFrame()`. Tato metoda sestrojí zprávu, odešle a znova naplánuje další přenos.
- `ShutdownWhile` - tento časovač se nastaví, když dojde k vypnutí agenta. Určuje, kdy může dojít k jeho další inicializaci.

Kromě výše zmíněného případu odeslání zprávy, modul odešle zprávu i v případě, když zjistí, že se některé rozhraní vypne. Před samotným vypnutím příslušný agent pomocí metody `txShutdownFrame()` odešle *shutdown* zprávu skrze toto rozhraní. Tato vlastnost ještě ale nefunguje, protože v INETu dojde k zahození zprávy na výstupním rozhraní, které je již tou dobou vypnuté.

Systém kreditů zatím v simulaci nemá žádné opodstatnění, jelikož nikdy nemůže dojít k více než jedné změně za jednu sekundu. Proto nikdy nedojde k vyčerpání kreditu. Toho půjde docílit pouze v případě, že bude možné sledovat proměnné, které se používají ke skládání LLDP zprávy. Když pak některá z těchto proměnných změní svojí hodnotu, dojde k požadavku na odeslání LLDP zprávy. Jelikož tento systém zatím pouze zdržoval simulaci, je ve výchozím nastavení vypnutý. Zapnout ho lze definováním makra `CREDIT` v souboru *LLDPMain.h*.

Pokud došlo k doručení externí zprávy, zkontroluje se její validita metodou `frameValidation()`. Metoda ověří, zda zpráva splňuje následující pravidla:

- obsahuje všechny povinné TLV;
- žádné TLV ve zprávě není obsažené vícekrát, než může být;
- délka uvedená v TLV je shodná se skutečnou délkou TLV.

Tato metoda rovněž smaže všechny TLV, které se nacházejí za TLV typu *End Of TLV*. Tato zpráva se pak dál předá příslušnému agentovi. Ten v prvním kroku ověří, zda se nejedná o *shutdown* zprávu, a jestliže ano, smaže příslušného souseda z tabulky sousedů. Pokud se jednalo o normální zprávu, tak ověří, zda soused ze zprávy existuje v tabulce sousedů. Pokud soused v tabulce není, vytvoří ho a přidá do tabulky. Informace o sousedovi pak aktualizuje dle informací získaných ze zprávy, a pokud soused obsahoval informace, které se ve zprávě nevyskytovaly, tak tyto informace smaže.

Jestliže dojde k objevení nového souseda, na příslušném agentovi dojde k zapnutí režimu rychlého startu.

Jakmile dojde k ukončení simulace, každý agent vypíše své statistiky ohledně odesílání a příjmu zpráv.

6.3 Třídy pro uchování dat

Modul LLDP obsahuje dva jednoduché moduly sloužící jako tabulky, jejichž účelem je uchovávat informace o agentech a naučených sousedech.

6.3.1 Tabulka agentů LLDPAgentTable

Tabulka osahuje informace o všech agentech ze zařízení. Položky tabulky jsou tvořeny třídou `LLDPAgent`.

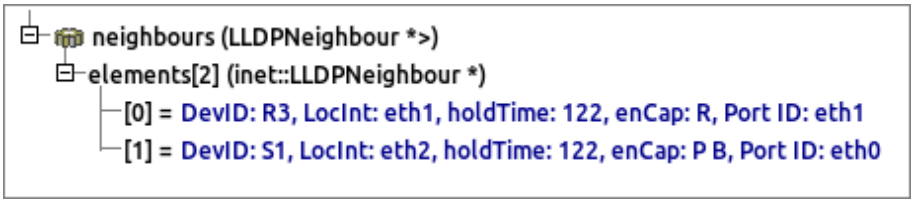
Tabulka obsahuje metody pro přidávání, mazání a vyhledávání agentů. Kromě toho obsahuje i metodu `startAgents()` pro zapnutí všech agentů a metodu `printStats()` pro tisk statistik všech agentů. Obsah tabulky lze vidět na obrázku 6.3.

Obrázek 6.3: Textová podoba LLDP tabulky agentů.

6.3.2 Tabulka sousedů LLDPNeighbourTable

V této tabulce jsou uchovány informace o naučených sousedech, kdy každý z nich je instanci třídy `LLDPNeighbour`. Tato třída obsahuje informace o sousedovi, odkaz na agenta skrze kterého byl naučen a časovač expirace *rxInfoTtl*.

Obdobně jak v minulé tabulce, tabulka obsahuje základní metody pro přidávání, mazání a vyhledávání. Kromě toho umí metodou `removeNeighboursByAgent()` smazat všechny sousedy, kteří byli naučení skrze určitého agenta. Metodou `restartRxInfoTtl()` pak umí restartovat časovač expirace určitého souseda. Textovou podobu tabulky lze vidět na obrázku 6.4.



Obrázek 6.4: Textová podoba LLDP tabulky sousedů.

6.4 LLDP zprávy

Všechny zprávy zpracovávané modulem `LLDPMain` lze rozdělit na dva typy. Jedná se o vlastní zprávy a o externí zprávy.

6.4.1 Vlastní zprávy

Tyto zprávy modul zaslal sám sobě. Simulují vypršení nějakého časovače. Jsou definovány v souboru `LLDPTimer.msg`, ze kterého OMNeT++ automaticky vygeneruje třídu těchto zpráv.

V této třídě je proměnná *timerType*, která určuje o který časovač se jedná. `LLDPMain` může přijmout tři typy časovačů. Prvním je časovač *Tick*, jež slouží k inkrementaci kreditů agentů a je volán každou sekundu. Druhým časovačem je *TTR* informující, že má dojít k pravidelnému odeslání zprávy sousedovi. Posledním časovačem je *ShutdownWhile* určující, kdy agent může znova přejít do zapnutého stavu.

6.4.2 Externí zprávy

Tyto zprávy jsou definovány v souboru `LLDPUpdate.msg`. Jejich struktura je následující:

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

  TLVOptions   options;
}
```

Stejně jak CDP externí zprávy, i tyto musí být dospecifikovány, a to ve třídě `LLDPUpdate`.

TLV pole bylo vytvořené pomocí třídy `TLVOptions` z knihovny INET. V této třídě je definována hlavička TLV a *value* část je pak definována pro každý TLV typ zvlášť. V následujícím příkladu lze vidět *chassis ID* TLV:

```
class LLDPOptionChassisId extends TLVOptionBase
{
   type = LLDPTLV_CHASSIS_ID;
   uint8_t  subtype;
   string  value;
}
```

6.5 Třída LLDPDeviceConfigurator

Jedná se o třídu, jež umí načíst konfiguraci LLDP z XML souboru. Jako výchozí konfigurace LLDP se bere nastavení v `LLDP.ned` souboru. Tímto nastavením ale nejde nastavit odlišné hodnoty jednotlivým LLDP agentům. S třídou `LLDPDeviceConfigurator` lze právě toho dosáhnout.

K získání konfigurace slouží metoda `loadLLDPConfig()`. Tato metoda rozparsuje XML soubor a nastaví hodnoty jednotlivým agentům. Lze nastavit následující hodnoty:

- `msgTxInterval` - interval odesílání pravidelných zpráv.
- `msgTxHold` - násobitel hodnoty *msgTxInterval* sloužící ke zjištění TTL hodnoty.

- `adminStatus` - určuje stav agenta. Zda je vypnutý (*disabled*), pouze vysílá (*enabledtxonly*), pouze přijímá (*enabledrxonly*) nebo odesílá i přijímá (*enabledrxtx*).
- `reinitDelay` - zpoždění od chvíle, kdy `adminStatus` byl nastaven na *disabled* a kdy může dojít k nové reinicializaci.
- `txFastInit` - kolik zpráv má být odesláno, když se spustí režim rychlého startu.
- `msgFastTx` - interval odesílání zpráv v režimu rychlého startu.
- `txCreditMax` - maximální počet kreditů.

Všechny tyto hodnoty jsou nepovinné a nemusí být uvedeny v konfiguračním souboru. Tento soubor může vypadat následovně:

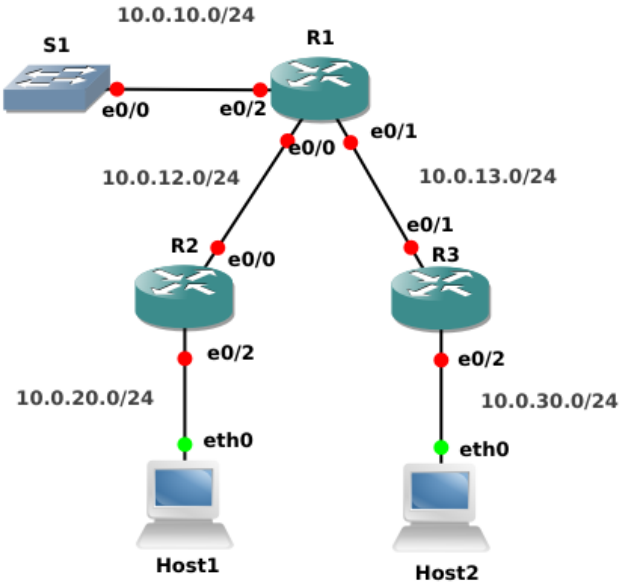
```
<Router id="R3">
  <Interfaces>
    <Interface name="eth0">
      <LLDP>
        <msgFastTx>1</msgFastTx>
        <msgTxHold>3</msgTxHold>
        <msgTxInterval>60</msgTxInterval>
        <reinitDelay>2</reinitDelay>
        <txCreditMax>5</txCreditMax>
        <txFastInit>8</txFastInit>
        <AdminStatus>enabledrxtx</AdminStatus>
      </LLDP>
    </Interface>
  </Interfaces>
</Router>
```

Kapitola 7

Porovnání simulace a reálné sítě

Tato kapitola se zabývá testováním a porovnáváním výsledků simulace v OMNeT++ s reálnou sítí. Jsou zde zapsané výsledky testování obou protokolů. Jako testovací zařízení reálné sítě byly použity směrovače Cisco s operačním systémem IOS 15.4(2)T4 (`I86BI_LINUX-ADVENTERPRISEK9-M`) a přepínač Cisco s IOS 15.2 (`I86BI_LINUXL2-ADVENTERPRISEK9-M`).

Všechny následující testy byly provedené na stejné topologii a liší se tedy jen scénářem. V OMNeT++ je simulace umístěná ve složkách `examples/ansa/cdp/mixedNetwork` a `examples/ansa/lldp/mixedNetwork`. Výsledky každého jednoho scénáře jsou popsány v následujících podkapitolách. K odchytávání paketu z reálné sítě byl použit program *Wireshark*. V každém testu jsou vypsané jen tabulky, které souvisí s daným testováním. Topologie sítě je na obrázku 7.1.



Obrázek 7.1: Topologie testované sítě.

Na rozhraních směrem k hostům jsou oba dva protokoly vypnuté. V protokolu LLDP byly pro lepší přehlednost změněny výchozí hodnoty časovačů. Interval pravidelného odesílání oznámení byl nastaven na 60 sekund a životnost sousedství byla nastavena na troj-

násobek tohoto intervalu.

Kvůli přehlednosti jsou z výpisu tabulek sousedů a směrovacích tabulek reálné sítě smazány hlavičky popisující významy zkratek.

V příloze C.2 se nachází porovnání CDP paketu simulační a reálné sítě. Jedná se o pakety odeslané z R1 přes rozhraní *e0/0*. V příloze D.2 se pak nachází porovnání paketů LLDP protokolu.

Časy přenosu paketů v simulaci a v reálné síti se budou vždycky trochu lišit. Je to způsobené tím, že simulace probíhá v diskrétním simulátoru. V reálné síti každá událost zabere nějaký čas než se vykoná. Například samotné zapnutí rozhraní zabere několik sekund. Z časového hlediska jsou tedy pro porovnání důležitější rozdíly časových značek mezi událostmi, než absolutní hodnoty.

7.1 Test ustavení sousedství

První test ukazuje proces vytvoření sousedství při zapnutí zařízení. Zařízení byla zapnuta ve stejnou chvíli, a byly pozorovány pakety, které prošly na lince R1-R2 během 80 sekund. Po této době byly zobrazeny tabulky sousedství.

7.1.1 CDP

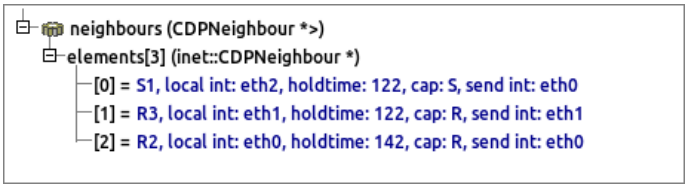
Z tabulky 7.1 lze vidět, že jakmile bylo rozhraní zapnuto, CDP přešlo do režimu rychlého startu. Poslalo tedy tři oznámení v řadě s rozestupem jedné sekundy. V reálné síti se na zařízení R2 projevil jeho pozdější start, a tak pakety začaly chodit o přibližně pět sekund později.

Simulovaná síť	
Směr	**Čas [s]**
R1 → R2	0
R2 → R1	0
R1 → R2	1
R2 → R1	1
R1 → R2	2
R2 → R1	2
R1 → R2	60
R2 → R1	60

Reálná síť	
Směr	**Čas [s]**
R1 → R2	0.30
R1 → R2	1.30
R1 → R2	2.31
R2 → R1	5.37
R2 → R1	6.37
R2 → R1	7.38
R1 → R2	57.55
R2 → R1	66.85

Tabulka 7.1: Výměna rámců na lince R1-R2 při ustavení sousedství.

Obrázky 7.2 a 7.3 zobrazují tabulky sousedství zařízení R1 v obou sítích.



Obrázek 7.2: CDP tabulka sousedství směrovače R1 ze simulace ustavení sousedství.

```
R1#show cdp neighbors
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
R2               Eth 0/0           121              R B   Linux Uni Eth 0/0
R3               Eth 0/1           149              R B   Linux Uni Eth 0/1
S1               Eth 0/2           139            R S I   Linux Uni Eth 0/0

Total cdp entries displayed : 3
```

Obrázek 7.3: CDP tabulka sousedství směrovače R1 z reálné sítě ustavení sousedství.

7.1.2 LLDP

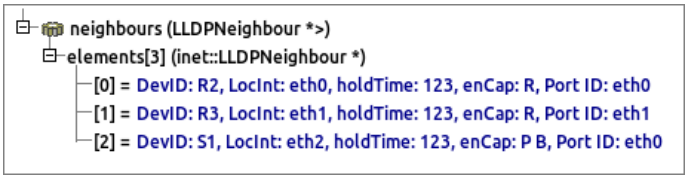
Výměnu paketů lze vidět v tabulce 7.2. Obě sítě se liší v počtů vyměněných paketů. Je to způsobené tím, že Cisco zařízení zapnou režim LLDP rychlého startu pouze ve dvou případech. V prvním případě když dojde k výměně LLDP-MED paketu a sousední zařízení není v LLDP tabulce sousedství. Druhý případ nastane, pokud se jedná o koncové zařízení, jako například telefon, a dojde k zapnutí linky. Ve standardu LLDP není takové chování popsáno, a proto není v této implementaci podporováno.

Simulovaná síť	
Směr	**Čas [s]**
R1 → R2	0
R2 → R1	0
R1 → R2	0
R2 → R1	0
R1 → R2	1
R2 → R1	1
R1 → R2	2
R2 → R1	2
R1 → R2	62
R2 → R1	62

Reálná síť	
Směr	**Čas [s]**
R1 → R2	1.6
R2 → R1	1.9
R1 → R2	61.3
R2 → R1	61.4

Tabulka 7.2: Výměna rámců na lince R1-R2 při ustavení sousedství.

Z obrázku 7.4 a 7.5 pak lze vidět, že naučení sousedi zařízením R1 si odpovídají v obou sítích.



Obrázek 7.4: LLDP tabulka sousedství směrovače R1 ze simulace ustavení sousedství.

```
R1#show lldp neighbors
Device ID           Local Intf     Hold-time  Capability      Port ID
R3                  Et0/1          180        R               Et0/1
S1                  Et0/2          180        R               Et0/0
R2                  Et0/0          180        R               Et0/0

Total entries displayed: 3
```

Obrázek 7.5: LLDP tabulka sousedství směrovače R1 z reálné sítě ustavení sousedství.

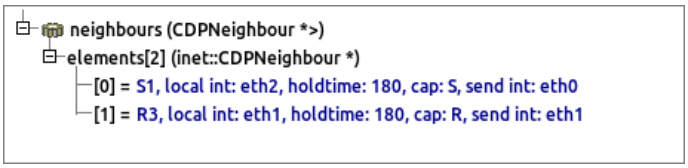
7.2 Test pádu rozhraní

Následující test zkoumá odstranění sousedství. Výchozí stav testu je, že je navázané sousedství mezi všemi sousedními zařízeními. V čase 50 sekund pak dojde k vypnutí rozhraní *e0/0* zařízení R1. Mělo by tedy po vypršení *holdtime* časovače (180 sekund) dojít k odstranění sousedství mezi zařízením R1 a R2.

K simulování pádu rozhraní se využívá modul `ScenarioManager`. Tento modul umí rozpojit spojení mezi dvěma rozhraními. K rozpojení kterého spojení a ve kterém čase je specifikováno v souboru `scenario.xml`.

7.2.1 CDP

Na obrázcích 7.6 a 7.7 lze vidět tabulky sousedství zařízení R1 v čase 200 sekund. V tomto čase již došlo k vypršení *holdtime* časovače souseda. Poslední aktualizace souseda byla ve druhé sekundě, a tak ve 182 sekundě došlo k odstranění souseda. Jde vidět, že simulační síť odpovídá reálné síti. *Holdtime* časy sousedů se v obou tabulkách liší, což je způsobeno tím, že OMNeT++ je diskrétní simulátor, a proto časy simulace nebyly aktualizované.



Obrázek 7.6: CDP tabulka sousedství směrovače R1 ze simulace pádu rozhraní.

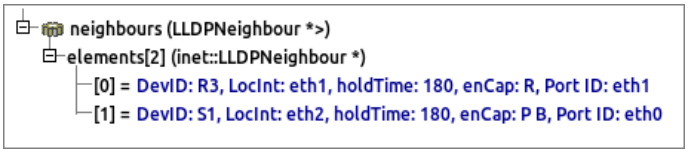
```
R1#show cdp neighbors
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
R3               Eth 0/1           162              R B   Linux Uni Eth 0/1
S1               Eth 0/2           162            R S I   Linux Uni Eth 0/0

Total cdp entries displayed : 2
```

Obrázek 7.7: CDP tabulka sousedství směrovače R1 z reálné sítě pádu rozhraní.

7.2.2 LLDP

V čase 200 sekund obě tabulky sousedů obsahují po dvou sousedech, což lze vidět na obrázcích 7.8 a 7.9.



Obrázek 7.8: LLDP tabulka sousedství směrovače R1 ze simulace.

```
R1#show lldp neighbors
Device ID           Local Intf     Hold-time  Capability      Port ID
R3                  Et0/1          180        R               Et0/1
S1                  Et0/2          180        R               Et0/0

Total entries displayed: 2
```

Obrázek 7.9: LLDP tabulka sousedství směrovače R1 z reálné sítě.

7.3 Test zapnutí rozhraní

Tento test navazuje na minulý test 7.2, kdy bylo rozhraní *e0/0* zařízení R1 vypnuto. V čase 200 sekund se toto rozhraní znovu zapne. Porovnávat se budou pakety, které touto linkou prošly během 30 sekund, a tabulky sousedství obou sítí v čase 230 sekund.

7.3.1 CDP

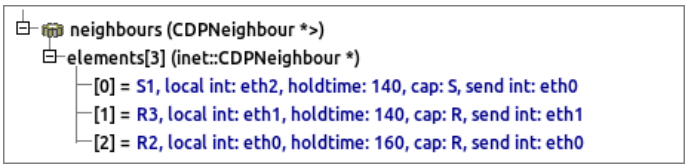
Z tabulky 7.3 lze vidět, že v obou sítích nejprve nastal mód rychlého startu a obě zařízení odeslala po třech paketech.

Simulovaná síť	
Směr	**Čas [s]**
R1 → R2	200
R2 → R1	200
R1 → R2	201
R2 → R1	201
R1 → R2	202
R2 → R1	202

Reálná síť	
Směr	**Čas [s]**
R1 → R2	199.48
R1 → R2	200.5
R2 → R1	201.5
R1 → R2	201.51
R2 → R1	202.51
R2 → R1	203.51

Tabulka 7.3: Výměna rámců na lince R1-R2 při zapnutí rozhraní.

V tabulce sousedství zařízení R1 znovu přibyl soused R2. Tabulky z obou sítí jsou na obrázcích 7.10 a 7.11.



Obrázek 7.10: CDP tabulka sousedství směrovače R1 ze simulace zapnutí rozhraní.

```
R1#show cdp neighbors
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
R2               Eth 0/0           152              R B   Linux Uni Eth 0/0
R3               Eth 0/1           137              R B   Linux Uni Eth 0/1
S1               Eth 0/2           137            R S I   Linux Uni Eth 0/0

Total cdp entries displayed : 3
```

Obrázek 7.11: CDP tabulka sousedství směrovače R1 z reálné sítě zapnutí rozhraní.

7.3.2 LLDP

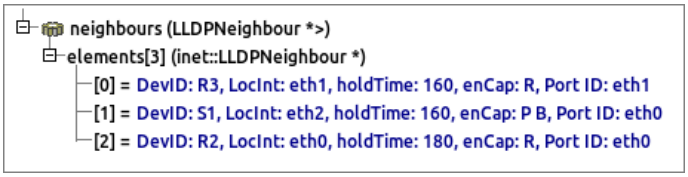
Z tabulky 7.4 lze vidět, že jakmile se rozhraní *e0/0* zařízení R1 zapnulo, okamžitě odeslalo LLDP rámec. R1 již nemělo R2 ve své tabulce sousedů, proto začalo režim rychlého startu. Jakmile R2 zpracoval první rámec, rovněž přešel do režimu rychlého startu. Důvod, proč se simulovaná síť liší od té reálné, je popsán v kapitole 7.1.2.

Simulovaná síť	
Směr	**Čas [s]**
R2 → R1	200.00
R1 → R2	200.01
R2 → R1	200.01
R1 → R2	201.01
R2 → R1	201.01
R1 → R2	202.01
R2 → R1	202.01

Reálná síť	
Směr	**Čas [s]**
R1 → R2	202
R2 → R1	205

Tabulka 7.4: Výměna rámců na lince R1-R2 při zapnutí rozhraní.

Obě tabulky sousedů znovu obsahují všechna sousední zařízení (viz obr. 7.12 a 7.13).



Obrázek 7.12: LLDP tabulka sousedství směrovače R1 ze simulace zapnutí rozhraní.

```
R1#show lldp neighbors
Device ID          Local Intf     Hold-time  Capability      Port ID
R3                 Et0/1          180        R               Et0/1
S1                 Et0/2          180        R               Et0/0
R2                 Et0/0          180        R               Et0/0

Total entries displayed: 3
```

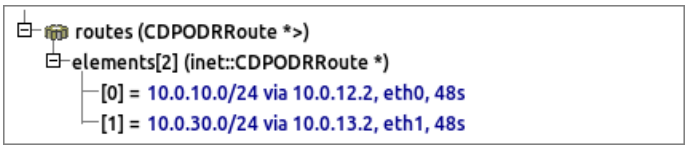
Obrázek 7.13: LLDP tabulka sousedství směrovače R1 z reálné sítě zapnutí rozhraní.

7.4 Test ODR

Tento test je rozdělen do dvou částí. V první části budou zobrazeny směrovací tabulky obou sítí v čase 30 sekund, kdy na R1 bude spuštěno ODR. Všechna rozhraní všech zařízení budou zapnutá. V druhé části se pak v čase 50 sekund vypne zařízení R2 a v čase 100 sekund zařízení R3.

7.4.1 Výměna směrovacích informací

Na obrázcích 7.14 a 7.15 lze vidět, že obě směrovací tabulky se rovnají, a tedy došlo ke správné výměně sítí.



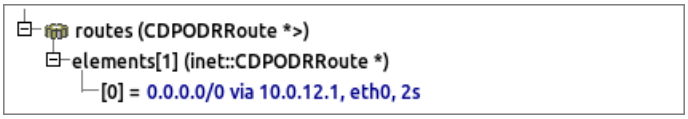
Obrázek 7.14: ODR směrovací tabulka směrovače R1 ze simulace.

```
R1#show ip route odr
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
o        10.0.20.0/24 [160/1] via 10.0.12.2, 00:00:30, Ethernet0/0
o        10.0.30.0/24 [160/1] via 10.0.13.2, 00:00:30
```

Obrázek 7.15: ODR směrovací tabulka směrovače R1 z reálné sítě.

Rovněž došlo k výměně výchozí cesty mezi R1 a R2. Tuto cestu lze vidět na obrázcích 7.16 a 7.17.



Obrázek 7.16: ODR směrovací tabulka směrovače R2 ze simulace.

```
R2#show ip route odr
o*    0.0.0.0/0 [160/1] via 10.0.12.1, 00:00:33, Ethernet0/0
```

Obrázek 7.17: ODR směrovací tabulka směrovače R2 z reálné sítě.

7.4.2 Vypnutí rozhraní

Směrovací tabulky z obrázků 7.18 a 7.19 byly vypsány v čase 250 sekund. Jelikož došlo k vypnutí R2 v čase 50 sekund a R3 v čase 100 sekund, došlo k odstranění sítě 10.0.20.0/24 a k označení sítě 10.0.30.0/24 jako *invalide*.

Obrázek 7.18: ODR směrovací tabulka směrovače R1 ze simulace vypnutí rozhraní.

```
R1#show ip route odr
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
o        10.0.30.0/24 is possibly down,
          routing via 10.0.13.2
```

Obrázek 7.19: ODR směrovací tabulka směrovače R1 z reálné sítě vypnutí rozhraní.

7.5 Zhodnocení

Ze všech provedených testů lze vidět, že tato implementace CDP, LLDP a ODR odpovídá Cisco implementaci těchto protokolů. Lze tedy předpokládat, že se jedná o správnou implementaci. Jediná odlišnost oproti Cisco implementaci nastala v LLDP protokolu. Cisco implementace má oproti standardu jinak implementovaný režim rychlého startu. Tato odlišnost je blíže popsána v podkapitole 7.1.2.

Kapitola 8

Závěr

Cílem práce bylo rozšíření simulačního nástroje OMNeT++ o protokoly pro správu sítě na vrstvě L2. Konkrétně se jedná o protokoly CDP, LLDP a ODR směrování. Výsledkem práce jsou funkční modely pro simulaci těchto protokolů, které využívají INET knihovnu sloužící pro simulaci počítačových sítí.

V první části práce jsou popsány principy a vlastnosti těchto protokolů. Ke každému protokolu jsou také vypsány jeho konfigurační a ověřovací příkazy na zařízeních firmy Cisco. Při následné implementaci bylo čerpáno právě z této části.

Návrhem a implementací se zabývá druhá část práce. V této části jsou popsány moduly, které byly v rámci této práce vytvořené. Je zde popsán i princip konfigurace jednotlivých protokolů.

Součástí práce je i popis odstranění závislostí na modulu `DeviceConfigurator` používaným v knihovně ANSAINET. Bylo rozhodnuto, že tento modul se již nebude vyskytovat v dalších verzích ANSAINETu, a proto došlo k jeho odstranění.

Poslední část práce se zabývá testováním implementace. Ověření korektnosti proběhlo simulováním různých situací, které mohou v síti nastat. Jednalo se například o zapnutí nebo vypnutí rozhraní. Výsledky byly porovnány se stejně zapojenou reálnou sítí. Výsledky z obou sítí se rovnají, což ukazuje na skutečnost, že implementace těchto protokolů byla provedena správně.

V blízké době bude snaha začlenit tyto protokoly do knihovny INET, aby mohly být používány širší veřejností. Jak bylo popsáno v kapitolách zabývajících se implementací, několik funkci těchto protokolů ještě nefunguje. Tyto funkce jsou již implementovány, ale bylo by potřeba poupravit INET knihovnu. Následný vývoj by mohl spočívat v úpravě knihovny INET, aby mohly fungovat všechny tyto funkce.

Literatura

[1] Configuring LLDP and LLDP-MED. Cisco Systems, 2014, [Online; navštíveno 11.12.2015].
URL http://www.cisco.com/c/en/us/td/docs/switches/connectedgrid/cg-switch-sw-master/software/configuration/guide/sysmgmt/CGS_1000_Sysmgmt/sm_lldp.html

[2] Frame Formats. Cisco Systems, 2015, [Online; navštíveno 11.12.2015].
URL
http://docstore.mik.ua/univercd/cc/td/doc/product/lan/trsrb2/frames.pdf

[3] IEEE Standard for Local and metropolitan area networks– Station and Media Access Control Connectivity Discovery Corrigendum 2: Technical and Editorial Corrections. *IEEE Std 802.1AB-2009/Cor 2-2015 (Corrigendum to IEEE Std 802.1AB-2009)*, March 2015, doi:10.1109/IEEESTD.2015.7056401.

[4] Alperovich, A.; Davidov, A.; Kostenko, B.: CDP Implemantation. 2003, [Online; navštíveno 11.12.2015].
URL http://www.cs.technion.ac.il/Courses/Computer-Networks-Lab/projects/spring2003/cdp2/web_cdp2/web_cdp2/cdp2_report.htm

[5] Automated Network Simulation and Analysis. ANSA, 2012, [Online; navštíveno 2.4.2016].
URL https://nes.fit.vutbr.cz/ansa/pmwiki.php

[6] Attar, V. Z.; chandwadkar, P.: Article: Network Discovery Protocol LLDP and LLDP-MED. *International Journal of Computer Applications*, roÄnÄk 1, Ä. 9, February 2010: s. 93–97, published By Foundation of Computer Science.

[7] Behavior of Cisco Discovery Protocol between Routers and Switches. Cisco Systems, 2015, [Online; navštíveno 11.12.2015].
URL http://www.cisco.com/c/en/us/support/docs/network-management/discovery-protocol-cdp/118736-technote-cdp-00.html

[8] Using Cisco Discovery Protocol. Cisco Systems, 2014, [Online; navštíveno 11.12.2015].
URL http://www.cisco.com/c/en/us/td/docs/ios/netmgmt/configuration/guide/12_2sx/nm_12_2sx_book/nm_cdp_discover.html

[9] Cisco Feature Navigator. Cisco Systems, [Online; navštíveno 2.4.2016].
URL http://tools.cisco.com/ITDIT/CFN/jsp/by-feature-technology.jsp

[10] LLDP-MED and Cisco Discovery Protocol. Cisco Systems, [Online; navštíveno 2.4.2016].

URL http://www.cisco.com/en/US/technologies/tk652/tk701/technologies_white_paper0900aecd804cd46d.html

[11] INET Framework. INET FRAMEWORK, 2016, [Online; navštíveno 2.4.2016].
URL http://inet.omnetpp.org/

[12] Link Layer Discovery Protocol. Hewlett Packard, 2006, [Online; navštíveno 5.4.2016].
URL ftp://ftp.hp.com/pub/networking/software/A-C12-LLDP.pdf

[13] Mikšíček, P.: *Zjištění síťové architektury u poplachového přenosového systému LAN-RING*. Diplomová práce, Vysoké učení technické v Brné. Fakulta elektrotechniky a komunikačních technologií, 2012.

[14] On-Demand Routing Commands. Cisco Systems, 2013, [Online; navštíveno 11.12.2015].
URL http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_odr/command/ird-cr-book/ird-cr-odr.html

[15] Designing Large-Scale Stub Networks with ODR. Cisco Systems, 2015, [Online; navštíveno 11.12.2015].
URL http://www.cisco.com/c/en/us/support/docs/ip/on-demand-routing-odr/13710-39.html

[16] OMNeT++. OMNeT++, 2015, [Online; navštíveno 2.4.2016].
URL http://www.omnetpp.org/

[17] Simulation Manual. OMNeT++, 2015, [Online; navštíveno 2.4.2016].
URL https://omnetpp.org/doc/omnetpp/manual/

[18] Černý, M.: *Modelování IPv6 v prostředí OMNeT++*. Diplomová práce, Vysoké učení technické v Brné. Fakulta informačních technologií, 2011.

Přílohy

Příloha A

Seznam zkratek

ACL = Access Control List
ANSA = Automated Network Simulation and Analysis
ASCII = American Standard Code for Information Interchange
ATM = Asynchronous Transfer Mode
CDP = Cisco Discovery Protocol
CPU = Ccentral Processing Unit
DSCP = Differentiated Services Code Point
EIGRP = Enhanced Interior Gateway Routing Protocol
EDP = Extreme Discovery Protocol
IEEE = Institute of Electrical and Electronics Engineers
IETF = Internet Engineering Task Force
IGMP = Internet Group Message Protokol
IOS = Internetwork Operating System
IP = Internet Protocol
IPX = Internetwork Packet Exchange
ISDP = Industry Standard Discovery Protoco
ISL = Inter-Switch Link
LAN = Local area network
LLC = Logical Link Control
LLDP = Link Layer Discovery Protocol
LLDPDU = LLDP Data Unit
LLDP-MED = Link Layer Discovery Protocol-Media Endpoint Discovery
MAC = Media Access Control
MIB = Management Information Base
MSAP = MAC Service Access Point
NED = Network Description
NMS = Network Management System
ODR = On-Demand Routing
OID = Object Identifier
OSI = Open Systems Interconnection model
OSPF = Open Shortest Path First
PoE = Power over Ethernet
PPP = Point-to-Point Protocol
RIP = Routing Information Protocol
TIA = Telecommunications Industry Association

TLV = Type-Length-Value
TTL = Time To Live
SNAP = Subnetwork Access Protocol
SNMP = Simple Network Management Protocol
STP = Spanning Tree Protocol
VLAN = Virtual Local Area Network
VLSM = Variable-Length Subnet Mask
VoIP = Voice over Internet Protocol
VTP = VLAN Trunking Protocol
XML = Extensible Markup Language

Příloha B

Obsah CD

V následující tabulce B.1 je uveden obsah přiloženého CD.

dip-xrajca00.pdf	Elektronická verze práce ve formátu PDF.
readme.txt	Obsah CD.
examples/	Simulační příklady `mixedNetwork` použité v kapitole 7 a několik dalších.
install/	Soubory pro instalaci OMNeT++ a INET.
latex/	Zdrojový text technické zprávy.
src/	Zdrojové kódy implementující CDP, LLDP a ODR směrování.
src-ansainet/	Knihovna ANSAINET-3.2.1 spolu s vytvořenými moduly.
src-del_DevConf/	Knihovna ANSAINET-2.2 se smazaným modulem DeviceConfigurator.

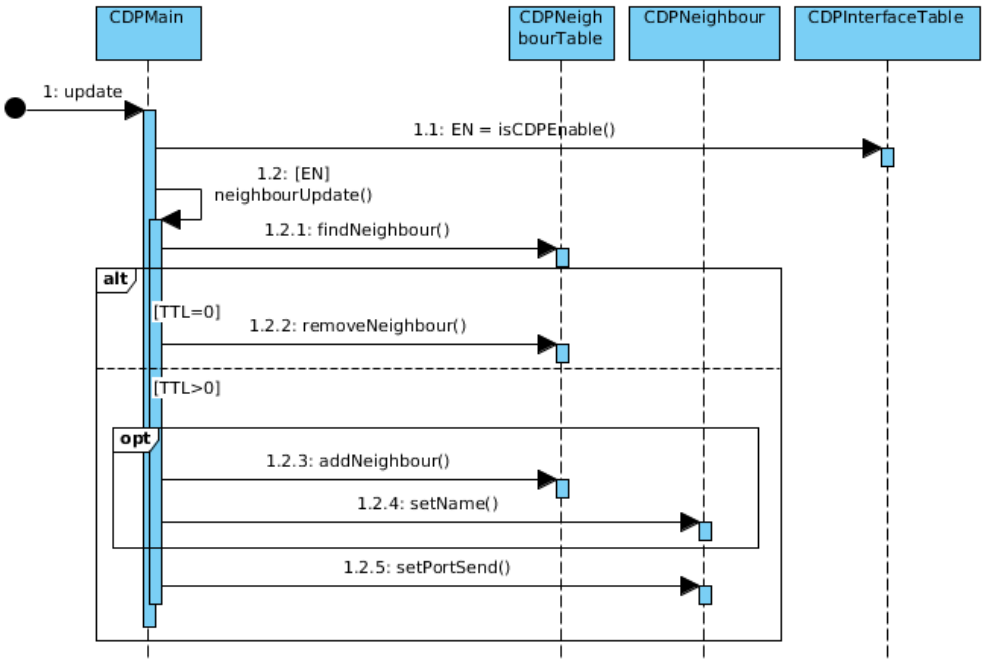
Tabulka B.1: Obsah přiloženého CD

Příloha C

CDP

C.1 Sekvenční diagram přijetí vnější zprávy

Obrázek C.1 zobrazuje příklad volání objektů při zpracování CDP zprávy.



Obrázek C.1: Sekvenční diagram přijetí CDP zprávy.

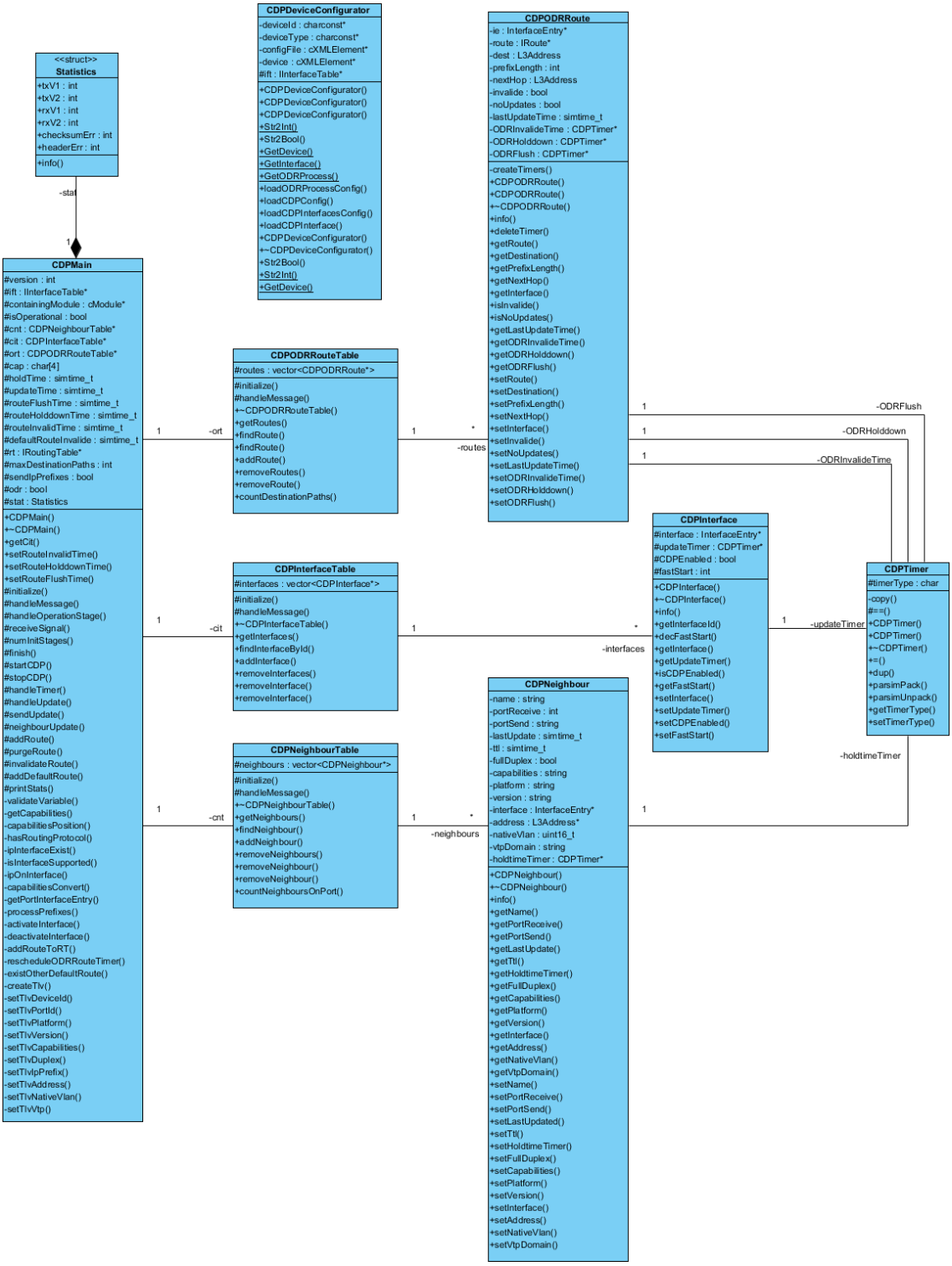
C.2 Pakety

Na obrázku C.2 je porovnání CDP paketů simulační sítě a reálné sítě.

Obrázek C.2: V levé části se nachází paket ze simulační sítě a v pravé z reálné sítě.

C.3 Diagram tříd

Na obrázku C.3 lze vidět diagram tříd CDP modulu.



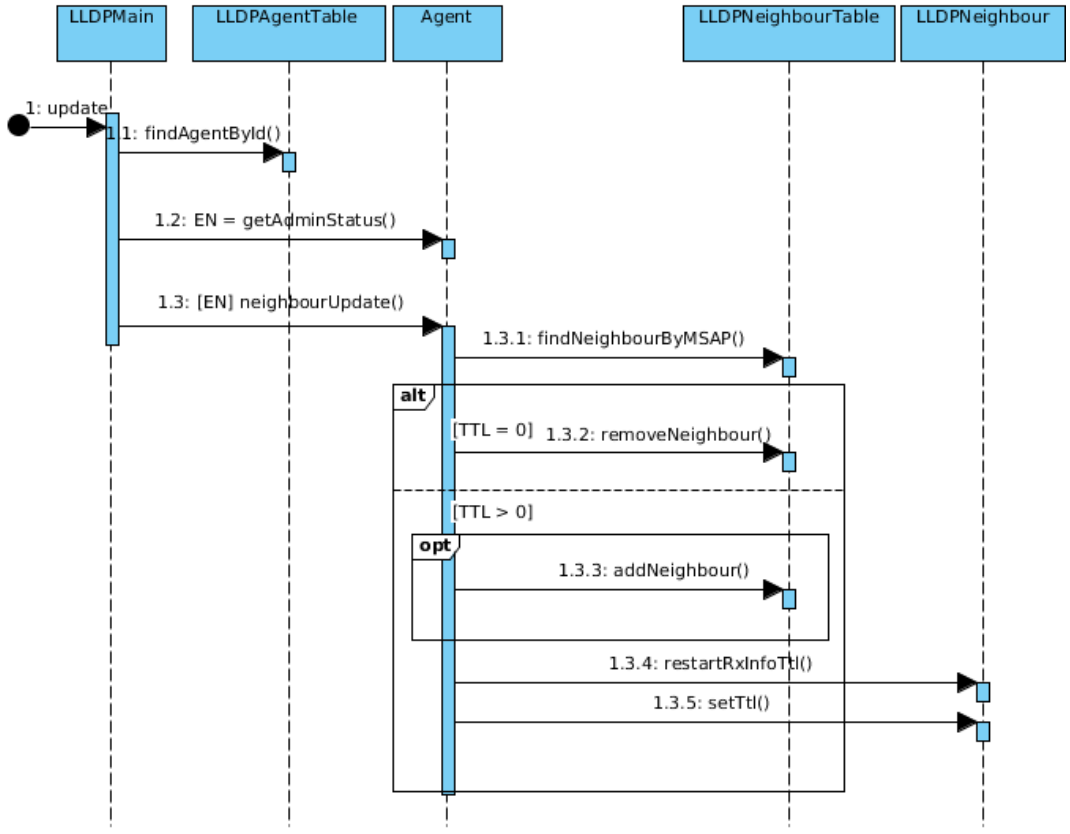
Obrázek C.3: Diagram tříd CDP modulu.

Příloha D

LLDP

Obrázek D.1 zobrazuje sekvenční diagram znázorňující přijetí LLDP zprávy, jež neobsahuje žádné volitelné TLV.

D.1 Sekvenční diagram přijetí vnější zprávy



Obrázek D.1: Sekvenční diagram přijetí LLDP zprávy neobsahující žádné volitelné TLV.

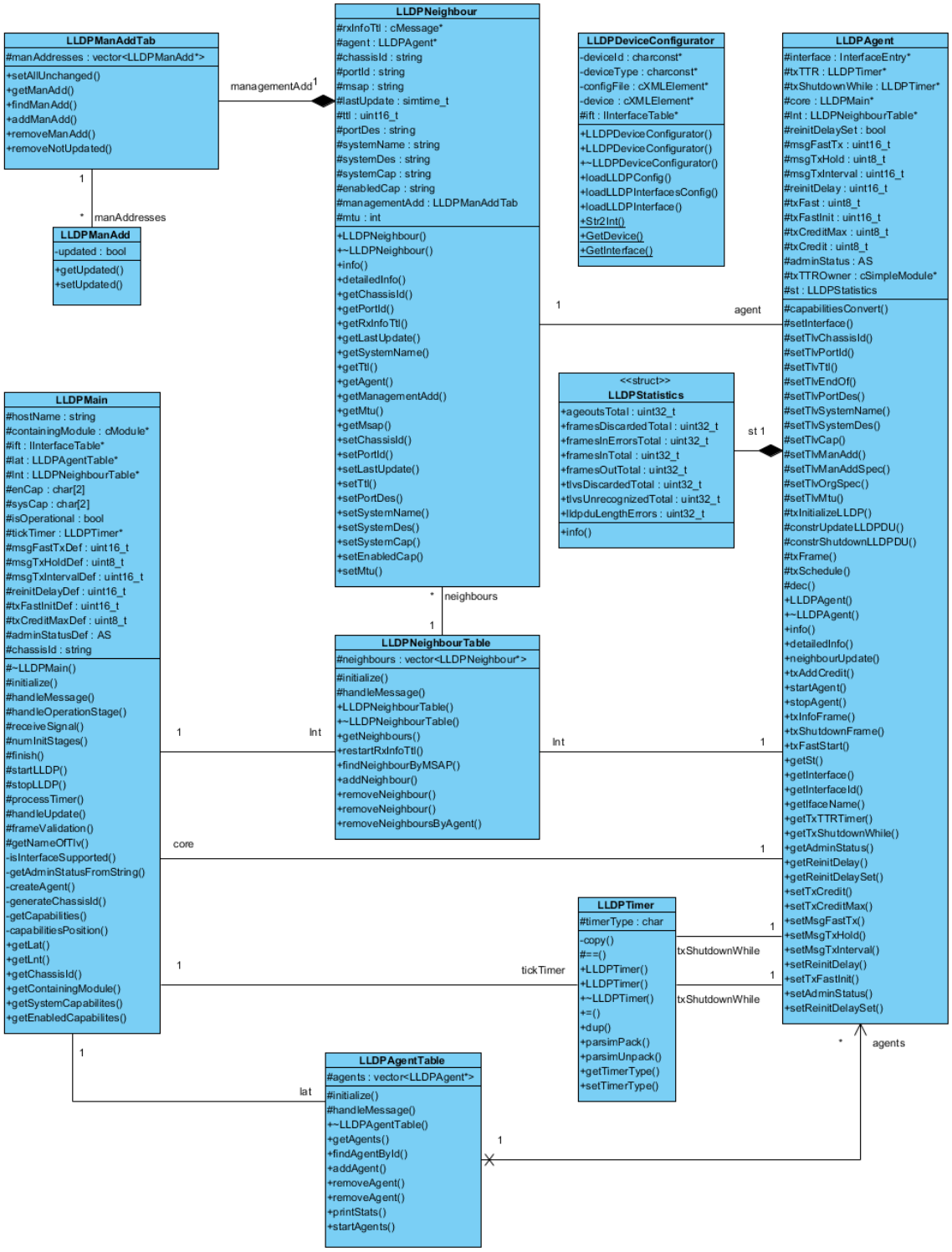
D.2 Pakety

Na obrázku D.2 je porovnání LLDP paketů simulační sítě a reálné sítě.

Obrázek D.2: V levé části se nachází paket ze simulační sítě a v pravé z reálné sítě.

D.3 Diagram tříd

Na obrázku C.3 lze vidět diagram tříd LLDP modulu.



Obrázek D.3: Diagram tříd LLDP modulu.