VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MONITOROVÁNÍ PROVOZU DHCP POMOCÍ IPFIX

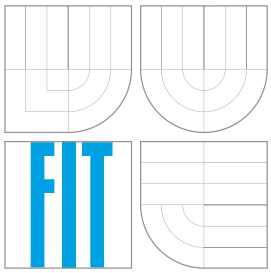
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MATĚJ VAŇÁTKO

BRNO 2014

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MONITOROVÁNÍ PROVOZU DHCP POMOCÍ IPFIX

DHCP MONITORING USING IPFIX

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MATĚJ VAŇÁTKO

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR MATOUŠEK, Ph.D.

BRNO 2014

Abstrakt

Tato práce popisuje postupy pro sledování provozu síťových protokolů BOOTP, DHCP pro IPv4 a DHCP pro IPv6 pomocí netflow sondy FlowMon od společnosti Invea-tech. Je zde nastíněna problematika těchto protokolů, funkčnost sondy FlowMon, obecný popis NetFlow a vlastní popis řešení pro sběr a vyhodnocení dat. Byla provedena důkladná analýza a poté byly sepsány moduly pro sondu FlowMon pro možnost monitoringu zmíněných protokolů. Jejich implementace, způsob testování a vyhodnocení získaných dat je v této práci popsán.

Abstract

This thesis describes procedures for traffic monitoring of network protocols BOOTP, DHCP for IPv4 and DHCP for IPv6 through netflow probes FlowMoon made by Invea-tech. There are outlined the issues of these protocols, the functionality of the FlowMoon probe, a general description of NetFlow and the description of the solution for collecting and evaluation of the data. A deep analysis was made, and later on the modules for FlowMoon probe was written giving the possibility to monitoring of these protocols. Their implementation, method of testing and evaluation of gathered data is described in this paper.

Klíčová slova

Monitorování provozu, DHCP, DHCPv6, BOOTP, IPv4, IPv6, NetFlow, IPFIX, sonda FlowMon

Keywords

Traffic monitoring, DHCP, DHCPv6, BOOTP, IPv4, IPv6, NetFlow, IPFIX, FlowMon probe

Citace

Matěj Vaňátko: Monitorování provozu DHCP pomocí IPFIX, bakalářská práce, Brno, FIT VUT v Brně, 2014

Monitorování provozu DHCP pomocí IPFIX

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Petra Matouška, Ph.D. a konzultantů Ing. Petra Špringla a Mgr. Martina Elicha

.....
Matěj Vaňátko
21. května 2014

Poděkování

Rád bych poděkoval mé rodině i mým zákazníkům za trpělivost, kterou se mnou měli při řešení této práce a také Ing. Petru Matouškovi, Ph.D. za odborné vedení práce, přátelské prostředí při psaní práce, za poskytnutí mnoha důležitých rad, za trpělivost, kterou se mnou doktor Matoušek při psaní práce měl a za tolerování mé souběžné práce při škole. Dále bych chtěl poděkovat lidem z firmy Invea-Tech, bez nichž by tato práce vznikala jen velmi obtížně.

Obsah

Kapitola 1

Úvod práce

V dnešní digitální době, kdy počítače ovlivňují náš život víc a víc se počítačové sítě a síť Internet dostaly v podstatě všude okolo nás. Rozšíření moderních technologií do mnoha oblastí našeho života způsobilo to, že je velmi nutné brát v potaz nejen jejich fyzické a logické zabezpečení, ale je zde i nutnost monitorování přenášených dat. Každého správce počítačové sítě jistě zajímají jak datové toky koncových uživatelů, tak provoz nutný pro správný chod sítě. Správce chce vždy vědět, jaká zařízení má do sítě připojena, co generují za síťový provoz, s kým (nebo s jakou službou) nejčastěji komunikují a jaké prostředky a protokoly k tomu používají. Jedním z možných sledovaných jevů jsou i konfigurační služby, které zajišťují nejen bezproblémové připojení koncových zařízení k síti, ale hlavně jejich správnou funkci – mít možnost komunikovat. Jedním z nástrojů pro správnou komunikaci klienta jsou i protokoly pro dynamickou a automatickou konfiguraci síťových rozhraní i síťových služeb klienta – protokol DHCP a jeho jednotlivé varianty. V práci jsou rozebírány v podstatě všechny hlavní typy protokolu DHCP – DHCP pro IPv4, DHCP pro IPv6[10] a starší protokol BOOTP. A právě sledováním síťového provozu generovaným těmito protokoly se tato práce zabývá.

Cílem této práce je tedy implementace modulů do sondy FlowMon firmy Invea–tech, umožňující sledování provozu těchto protokolů, analýzu jimi přenášených dat a jejich datových toků v počítačové síti, vyhodnocení těchto dat a jejich vizualizaci. V tomto dokumentu jsou rozebrány jak jednotlivé monitorované protokoly, tak i prostředky, které slouží pro sběr dat a správnou funkci FlowMon sondy. Dále je zde popsána architektura samotné FlowMon sondy, princip jejího programování, zpracování dat a také prezentace dat vůči uživateli, či možnosti nasazení v dnešních počítačových sítích. Velká část práce je take věnována testování funkčnosti obou modulů, které proběhlo nejen v laboratorních podmínkách, ale také nasazením do reálné sítě, konkrétně do počítačové sítě Obchodní akademie v Trutnově.

Kapitola 2

Rozbor použitých síťových protokolů

Pro správnou funkci koncových stanic lokální počítačové sítě pomocí protokolů z rodiny TCP/IP je nutné, aby měl počítač nastaveno několik základních parametrů. Těmi jsou zpravidla IP adresa, maska sítě, servery DNS či výchozí brána sítě. Všechny tyto údaje se dají nastavit dvěma způsoby. První možností je statické nastavení, kdy si nastavení těchto parametrů uchovává přímo operační systém klientského počítače na jeho pevném disku. Správce počítačové sítě tak musí všechny počítače osobně obejít a všechny potřebné údaje nastavit na každém z nich manuálně. To přináší nejen velkou administrativní zátěž, ale také mnoho chyb v nastavení a často též neaktuálnost údajů (např. vlivem stěhování počítače na jiné místo, pracoviště apod.).

- `BOOTP`
- `DHCP pro IPv4 -- dále jen DHCPv4`
- `DHCP pro IPv6 -- dále jen DHCPv6`

Proto se začaly využívat výše zmíněné centralizované protokoly pro dynamickou a automatickou konfiguraci všech potřebných údajů klienta pro komunikaci, které odstranily mnoho administrativní zátěže, veškerou konfiguraci velmi zjednodušily a také zpřehlednily. Správce počítačové sítě tak není nucen udržovat si dokumentaci síťových nastavení jednotlivých klientů ani je nijak fyzicky obcházet, potřebuje–li provést nějakou změnu v nastavení. Jestliže je nutná změna některého parametru (např. IP adresy serveru DNS), správce ho změní pouze v nastavení serveru pro dynamickou a automatickou konfiguraci klienta, díky němuž se automaticky dostane ke všem koncovým klientům počítačové sítě. Zavedení tohoto standardu odbouralo nejen chyby, které vznikaly při manuálním nastavování koncových zařízení počítačové sítě, ale přineslo také mnoho novinek. Jednou z největších je jistě možnost zavedení operačního systému přímo ze sítě, nikoliv z pevného disku počítače, jako tomu bylo doposud. Byl tedy umožněn vznik bezdiskových počítačových stanic a přístupových terminálů.

2.1 BOOTP

2.1.1 Princip a obecný popis protokolu

Prvním protokolem pro dynamickou a automatickou konfiguraci síťových rozhraní a služeb klienta byl právě `Bootstrap protocol (zkratka BOOTP)`. Protokol jako takový je již z roku 1985 a byl definován v `RFC 951[2], 1532[11], 1533[1] a 1542[12]`. Jeho hlavním úkolem nebylo až tak přidělování konfigurace síťových rozhraní, ale právě možnost startování a spouštění bezdiskových klientských počítačů ze spouštěcího serveru počítačové sítě. Princip protokolu je velmi jednoduchý. Na serveru musí být definován konfigurační soubor, ve kterém je jasně řečeno, jaké parametry klient dostane. Každý klient je rozlišen na základě své MAC adresy, která slouží jako jednoznačný identifikátor. Nevýhodou tohoto protokolu je, že MAC adresa a IP adresa klienta vždy tvoří pevnou dvojici. Pokud klient není zapnutý, jeho IP adresa nemůže být dynamicky přidělena někomu jinému.

2.1.2 Proces přidělení adresy

BOOTP protokol používá pro svoji komunikaci primárně transportní protokol UDP, port 67 pro naslouchání serveru a port 68, pro odpovědi na klientské požadavky. Samotný proces přidělení konfiguračních informací je velmi jednoduchý. Klient nejdříve pošle zprávu, kde sám sebe identifikuje (paket `bootrequest`) a kde žádá o přidělení konfigurace. Tuto zprávu pak pošle na IP adresu 255.255.255.255, čili na adresu všesměrového vysílání se zdrojovým portem 68 a cílovým 67. Jakmile BOOTP server zprávu zachytí, prohledá svůj konfigurační soubor a pokud nalezne klientovu MAC adresu, vygeneruje mu zprávu (paket `bootreply`), v níž mu přidělí všechny údaje, které má (např. IP adresu, masku podsítě, výchozí bránu, IP adresu serveru DNS aj.). Jestliže má klient zavést operační systém ze sítě, pošle mu k tomu ještě IP adresu spouštěcího serveru + jméno spouštěcího souboru. Formát paketu je stejný jak pro BOOTP dotaz, tak pro odpověď serveru. Mění se pouze vyplněné položky. Platnost BOOTP údajů je po celou dobu zapnutí klientské stanice. K obnovování dochází pouze po zapnutí počítače.

Operation	H/W Type	H/W Length	Hops
Transaction Identifier			
Seconds elapsed		Unused	
Client IP Address			
Your IP Address			
Server IP Address			
Router IP Address			
Client H/W address			
Server Host Name			
Bootfile Name			
Vendor Specific Area			

Obrázek 2.1: Formát paketu BOOTP

2.2 DHCP

2.2.1 Princip a obecný popis

Druhým protokolem dynamické konfigurace klienta je protokol DHCP. Dnes se jedná o snad nejrozšířenější protokol pro dynamickou konfiguraci a funkčnost počítačové sítě vůbec. Samotné DHCP je definováno v `RFC 1531`[3] (nejprve jako rozšíření BOOTP) a poté v `RFC 2131`[4] samostatně. DHCP používá stejně jako protokol BOOTP adresu MAC jako jednoznačný identifikátor komunikující stanice. Hlavním důvodem vzniku DHCP byla možnost lépe spravovat přidělený rozsah IP adres, než tomu bylo doposud v protokolu BOOTP. To neumožňovalo „pronájem adresy“, ale adresa byla zapůjčena trvale už ze své podstaty BOOTP protokolu a jeho konfiguračního souboru.

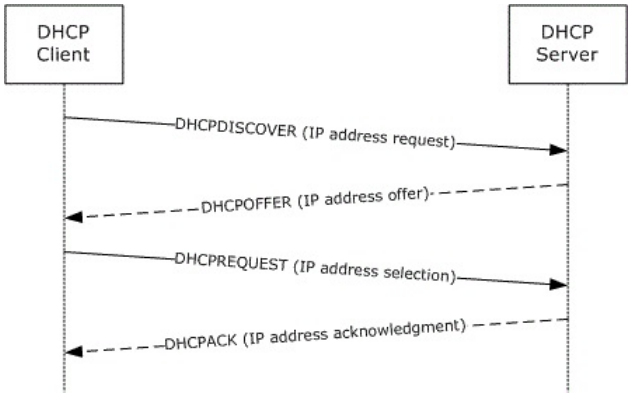
2.2.2 Změny oproti BOOTP

Nemožnost zapůjčení adresy jen na určitou dobu odstraňuje právě až protokol DHCP. Nově tedy klientovi nejsou poskytovány pouze konfigurační data, jako je např. IP adresa, maska sítě, IP adresa výchozí brány či IP adresa DNS serveru, ale je mu sdělena též doba, po kterou může tyto údaje klient používat. Jakmile si nepožádá o obnovení údajů, DHCP může poskytnutou IP adresu přidělit jinému klientovi a šetřit tak rozsah používaných adres. Pokud se podíváme na rozdíly mezi DHCP a BOOTP, je jich opravdu velmi málo. Pokud pomineme možnost přidělit konfigurační údaje na omezenou dobu, tak DHCP pouze změnilo nepoužívané 16 bitové pole hlavičky na položku `Flags -- příznaky` a pole `Oblasti definované dodavatelem` na pole `Možnosti` současně s jeho rozšířením ze 64B na max. 312B, což přineslo možnost konfigurovat pomocí DHCP daleko více věcí, než prostřednictvím BOOTP.

2.2.3 Proces přidělení adresy

DHCP přináší i mírně modifikovaný způsob komunikace klienta se serverem. Pro svoji funkčnost používá čtyři zprávy, které jsou přenášeny podle typu buď jednosměrovým vysíláním nebo pomocí všesměrového vysílání, kdežto BOOTP používal dvě stavové zprávy přenášené vždy pomocí všesměrového vysílání. Díky zpětné kompatibilitě se ale nezměnil port serveru (číslo 67), ani port pro odpověď (číslo 68). Samotný proces přidělování začíná opět všesměrovým požadavkem klienta o konfigurační údaje zprávou `DHCP Discover`. Jakmile tento požadavek zaregistruje některý z DHCP serverů, odpoví klientovi pomocí jednosměrové zprávy `DHCP Offer` s nabídkou IP adresy a dalších konfiguračních údajů. Ten si z (teoreticky z několika) nabídek vybere jednu IP adresu, kterou chce používat a vygeneruje všesměrovou zpřavu `DHCP Request`, prostřednictvím níž požádá server o přidělení jeho vybrané adresy. Server mu toto potvrdí opět jednosměrovou zprávou `DHCP Ack`. Po obdržení této informace může klient IP adresu vč. dalších rozšiřujících nastavení začít používat, ale maximálně po dobu, kterou mu DHCP server stanovil.

Jestliže klient nepožádá o znovuobnovení adresy, DHCP server ji může dynamicky přidělit jinému klientovi. Pokud však klient chce používat adresu dál, musí požádat server, tentokrát již jednosměrovou zprávou `DHCP Request`, o obnovu. Když se server rozhodne adresu obnovit, zašle klientovi jednosměrově potvrzení `DHCP Ack` i s novou dobou, po kterou může klient adresu vč. dalších konfiguračních údajů použít. Jestliže však klient `DHCP Ack` neobdrží a jemu přidělená doba vypůjčení adresy vyprší, musí adresu okamžitě přestat používat a celý proces přidělení adresy opakovat tak jako na začátku.



Obrázek 2.2: Proces přidělení IPv4 adresy pomocí DHCPv4

Samotný DHCP protokol ale zachovává upravenou možnost protokolu BOOTP zapůjčení adresy trvale. Je možné vytvořit manuálně stejný seznam párů MAC adresy a IP adresy, jako v případě protokolu BOOTP a ná základě něj vždy klientovi přidělit odpovídající stejnou adresu. U DHCP se ale tento způsob liší tím, že klient musí po určité době vždy požádat o obnovu adresy, narozdíl od protokolu BOOTP, kde byla adresa přidělena trvale a k obnovování docházelo pouze při startu počítače. Tento mechanismus se pak nazývá rezervace IPv4 adresy.

Operation	H/W Type	H/W Length	Hops
Transaction Identifier			
Seconds elapsed		Flags	
Client IP Address			
Your IP Address			
Server IP Address			
Router IP Address			
Client H/W address			
Server Host Name			
Bootfile Name			
Options (Variable)			

Obrázek 2.3: Formát paketu DHCP

2.3 DHCPv6

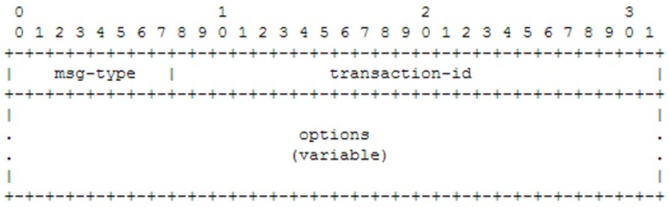
2.3.1 Obecný popis protokolu

Posledním protokolem, kterým se bude tato práce zabývat je protokol DHCPv6. Jak již název napovídá, je to úplně nový, stále se vyvíjející protokol, který funguje nebo může fungovat v sítích, kde je plošně nasazeno IPv6[10] a je v nich potřeba jistá část autokonfigurace. Samotné DHCPv6 definuje `RFC 3315`[6]. Další informace nutné k implementaci tohoto protokolu a k pochopení funkčnosti DHCPv6 je možné získat v `RFC 4861`[7], které popisuje princip objevování sousedů v IPv6 a `RFC 5175`[8], ve kterém jsou definovány jed-

notlivé příznaky, které poskytuje mechanismus oznamování směrovačů. Mě v této práci zajímal především příznak M a O, který rozhoduje o tom, jak se zachová klientský systém a zda je v síti použito stavové či bezstavové DHCPv6.

Samotné DHCPv6 je oproti DHCPv4 velmi elegantním a jednoduchým protokolem. Neobsahuje totiž de facto žádné povinné informace, jako tomu bylo u DHCPv4. DHCPv6 má povinné pouze dvě položky – `typ zprávy` a `ID transakce`. Vše ostatní už jsou volitelné položky, které z větší části definuje právě `RFC 3315`[6]. Další položky však definují mnohá další RFC, například `RFC 5908`[9], které do DHCPv6 implementuje podporu NTP serverů pro synchronizaci času. Takto pojatý protokol je velmi jednoduchý na pochopení, ale hlavně jednoduchý na zpracování. Každá volba ve svém těle obsahuje nejen položku, která udává číslo volby, ale i délku celé volby. Stejně jako u DHCPv4 lze tedy efektivně a jednoduše zpracovávat pouze ty volby, které nás zajímají. Ostatní volby se pouze přeskočí o jejich délku.

Velkou výhodou DHCPv6 je i to, že nepoužívá všesměrového vysílání. To je znemožněno již tím, že v IPv6 všesměrové vysílání neexistuje. DHCPv6 server komunikuje vždy pomocí vícesměrové adresy nebo pomocí své IPv6 adresy. Klient pro komunikaci použije vždy adresu z rosahu fe80::/10, čili link–local adresu. Není tedy nutné přidávat do identifikace IP toku žádné další údaje, protože se zde nemohou vyskytnout adresy typu 0.0.0.0 či 255.255.255.255, pomocí nichž komunikovalo v IPv4 více klientů a nebylo tak možné určit, který klient s DHCPv6 serverem právě komunikoval.



Obrázek 2.4: Formát paketu DHCPv6

2.3.2 Bezstavové DHCPv6

Jedná se o jednodušší verzi DHCPv6. Hlavní vlastností bezstavové DHCPv6 konfigurace je to, že DHCPv6 server klientovi **nepřiděluje** žádnou IPv6 adresu. Pro vygenerování adresy klient použije buď mechanismus `Privacy Extensions`, kterým klient nahodile vygeneruje nějakou IPv6 adresu, kterou samozřejmě v pravidelných intervalech obměňuje nebo použije algoritmus `EUI-64`, kdy si IPv6 adresu vygeneruje na základě své MAC adresy. Prefix sítě je mu vždy předán pomocí oznámení směrovače. DHCPv6 server pak slouží pouze pro přidělování dodatečných informací, které klient nezíská z oznámení směrovače. Tím jsou často IPv6 adresy DNS serverů, NTP serverů a podobně. Důležité je, že k bezstavové konfiguraci dochází vždy, pokud je do sítě zasíláno oznámení směrovače, které má nastaveno příznak O. Samotný DHCPv6 je v této konfiguraci velmi jednoduchý. Celá komunikace probíhá pouze pomocí dvou typů zpráv – žádost o informace a odpověď na ni. DHCPv6 server tedy nijak neřeší přidělení IPv6 adresy.

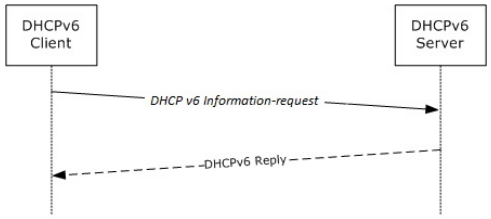
2.3.3 Stavové DHCPv6

Pravým opakem bezstavového DHCPv6 je takzvané stavové DHCPv6. Stavové DHCPv6 v síti neposkytuje pouze dodatečné konfigurační údaje, ale společně s nimi klientovi přiděluje i IPv6 adresu. Klient tak nemusí generovat adresu ani na základě `EUI-64`, ani pomocí `Privacy Extensions`. Správa IPv6 adres je tedy centralizovaná. Použití stavového DHCPv6 opět řídí mechanismus ohlášení směrovače. Klient z něj sice ignoruje oznámený prefix, ale reaguje na příznak M, který říká, že je v síti přítomen stavový DHCPv6 server a přikazuje klientovi ho kontaktovat.

Svoji funkcionalitou se tak velmi blíží DHCPv4 serveru, který zná každý správce sítě ve světě IPv4. Oproti němu se však v jedné věci podstatně liší. Tou věcí je identifikátor klienta. Zatímco server DHCPv4 jako identifikátor používá klientovu MAC adresu, která je mimochodem standardním způsobem neměnná, tak server DHCPv6 používá takzvaný DUID. Ten je unikátní v rámci operačního systému a je náhodně generovaný pomocí několika pravidel, která jsou popsána také v `RFC 3315`[6]. Toto řešení má však dvě nevýhody. Jelikož je DUID unikátní v rámci operačního systému, tak pokud je na počítači přítomno více operačních systémů současně, tak každý z nich bude mít svůj identifikátor. Jeden počítač se tedy bude jevit jako dva různé. Druhou nevýhodou je i to, že DUID není trvalý, ale je generován vždy při instalaci systému. Po přeinstalaci je tak opět vygenerován nový DUID. I na toto musí správce pamatovat a řešit to například centrální databází DUID a jeho změnou ihned po reinstalaci systému například na základě MAC adresy.

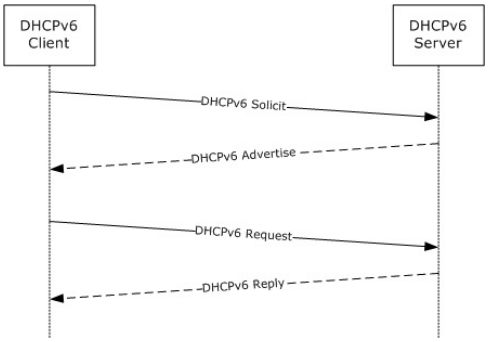
2.3.4 Proces přidělení adresy

Vlastní DHCPv6 server používá jako protokol transportní vrstvy UDP a naslouchá na portu 547. Klientovi pak odpovídá na cílový port 546. Díky nahrazení všesměrového vysílání za vícesměrové došlo také k definici adresy vícesměrového vysílání, na které DHCPv6 servery naslouchají. Jedná se buď o adresu `ff02::1:2` (všechny servery a agenti pro předávání v rámci síťového segmetu) nebo `ff05::1:3` (všechny servery v rámci organizace). Proces přidělení adresy se pak liší podle typu použitého DHCPv6 serveru. Jestliže je v síti přítomen bezstavový DHCPv6, IPv6 adresa z DHCPv6 serveru přidělována není a jsou pouze poskytnuty dodatečné konfigurační údaje jako například servery DNS. Klient pouze odešle zprávu `DHCPv6 Information-request` se seznamem požadovaných údajů na kterou DHCPv6 server odpoví zprávou `DHCPv6 Reply`, ve které konfiguraci poskytne.



Obrázek 2.5: Proces přidělení další konfigurace pomocí bezstavového DHCPv6

Naproti tomu stavový DHCPv6 již používá všechny typy DHCPv6 zpráv definovaných v `RFC 3315`[6], čili i proces přidělení adresy je složitější, avšak velmi podobný přidělování adresy pomocí DHCPv4. Klient žádající o adresu nejdříve odešle zprávu `DHCPv6 Solicit`, ve kterém se klient identifikuje svým DUID a žádá server o přidělení adresy. Server odpoví zprávou `DHCPv6 Advertise`, ve které nabídne nejen IPv6 adresu, ale i další konfigurační údaje. Klient pak serveru potvrdí, že údaje akceptuje a požádá server o potvrzení zapůjčení IPv6 adresy. Zašle tak serveru zprávu `DHCPv6 Request`. Pokud je vše v pořádku, server zapůjčení potvrdí pomocí zprávy `DHCPv6 Reply`.



Obrázek 2.6: Proces přidělení IPv6 adresy pomocí stavového DHCPv6

2.4 NetFlow

2.4.1 Obecný popis protokolu

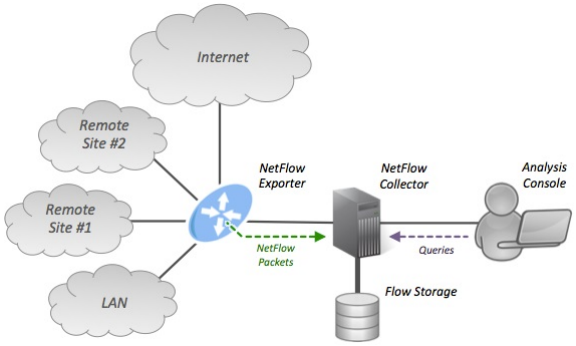
NetFlow je otevřený protokol primárně určený pro monitorování síťového provozu na základě jednotlivých IP toků a poskytuje tak správcům počítačové sítě podrobný přehled o provozu na počítačové síti v reálném čase. Původně byl NetFlow vyvinut americkou společností Cisco Systems jako doplňková služba k Cisco zařízením – převážně ke směrovačům. Dnes tvoří velmi důležitou a téměř nepostradatelnou součást zabezpečení každé IP sítě a je velmi důležitý nejen pro správce sítí, ale například i pro poskytovatele internetového připojení, kteří na základě analýzy NetFlow statistik mohou svým zákazníkům účtovat ceny na základě přenesených dat či podle typu přenesených dat. NetFlow protokol vznikl již v několika verzích. První masivně používanou verzí byla až verze NetFlow 5. Avšak v současnosti je již nahrazována verzí 9. Na základě NetFlow verze 9 však nedávno vznikl nový IETF standard `IPFIX`, který se také začíná hojně rozšiřovat. U tohoto protokolu je předpoklad, že se z něj brzy stane průmyslový základ a proto tvoří základ této práce.

Díky NetFlow je možné jednoduše plánovat efektivní rozvoj sítě, hledat úzká místa v síti, dominantní zdroje dat, sledovat, kdo s kým, jak dlouho a pomocí kterých protokolů komunikoval či detekovat vnitřní a vnější incidenty a hrozby. Toto všechno je možné právě díky rozlišení tzv. IP toků.

2.4.2 Obecná architektura NetFlow

Protokol NetFlow se zkládá ze dvou základních částí – `NetFlow kolektoru` a `NetFlow exportéru`. Zatímco exportéru je v počítačové síti většinou více, kolektor bývá v celé síti obvykle pouze jeden. NetFlow exportér je připojen přímo k monitorované lince a analyzuje procházející IP pakety a IP toky. Na základě zahycených dat jednotlivé IP toky analyzuje

a generuje NetFlow statistiky, které pak exportuje a odesílá do NetFlow kolektoru. To je síťové zařízení s dostatkem úložné kapacity, které sbírá data z jednotlivých exportérů a ukládá je do své databáze. Tyto data pak obvykle slouží pro běh aplikace, která je umí jednoduše prezentovat správci sítě pomocí soustavy grafů a tabulek. NetFlow data tvořená exportéry jsou přenášeny do kolektoru pomocí protokolu UDP nebo SCTP. Jakmile je záznam z exportéru poslán do kolektoru, je okamžitě z důvodu větší efektivity z exportéru smazán. Díky tomu může dojít ke ztrátě záznamu vlivem špatných vlastností linky, po které byl záznam do kolektoru odeslán. Pro export záznamů se často používají porty `2055, 3000--3010, 9555, 9995`.



Obrázek 2.7: Obecná architektura NetFlow

2.4.3 IP tok

IP tok je základem celého protokolu NetFlow. Právě na základě IP toku jsou generovány statistiky a NetFlow záznamy. V NetFlow terminologii je IP tok definován jako sekvence síťových paketů se stejnými údaji. Jsou jimi cílová a zdrojová IP adresa, cílový a zdrojový port a číslo L4 protokolu. U každého toku je sledován čas jeho vniku, délka jeho trvání, počet přenesených dat a paketů a další údaje. Je důležité si uvědomit, že každé obousměrné spojení je vždy popsáno nejméně dvěma toky. Jedním tokem ve směru od zařízení A k zařízení B a druhým tokem od zařízení B k zařízení A.

Tok vzniká v tu chvíli, kdy je zachycen paket, který svými vlastnosti nepatří do žádného z existujících toků. Další pakety se stejnými vlastnostmi jsou pak součástí tohoto nového toku. IP tok je prohlášen za ukončený, pokud nastane takzvané pasivní vypršení času. To je přesně definovaný čas, který uplyne od přijetí posledního paketu v daném IP toku. NetFlow exportéry ale mohou být nakonfigurovány tak, aby využívaly i takzvané aktivní vypršení času. To je opět přesně definovaný čas, který běží od přijetí prvního paketu daného IP toku a po jeho uplynutí je IP tok prohlášen za ukončený. Další přijatý paket, který by byl jinak součástí již tohoto ukončeného toku je prohlášen za první paket nového IP toku. Vlivem této konfigurace může docházet k tomu, že se dlouhé toky mohou rozpadat na několik časově kratších.

2.4.4 NetFlow záznam

Jak se měnila verze NetFlow, měnil se i samotný NetFlow záznam. Původní záznam NetFlow verze 5 obsahoval následující položky: číslo verze, číslo sekvence, SNMP index vstupního a výstupního rozhraní, čas začátku a konce IP toku, počet bajtů a paketů v toku, zdrojovou a

cílovou IP adresu, zdrojový a cílový port, IP protokol, číslo typu služby, IP adresu příštího hopu, masku cílové a zdrojové IP adresy a u TCP toků také množinu všech TCP příznaků.

Nová verze NetFlow číslo 9 a IETF standard IPFIX jsou definovány již jako plně flexibilní formáty. Záznam pak může obsahovat všechny hodnoty z NetFlow verze 5, ale také další volitelné položky, jako např. IPv6 adresy a porty, čísla VLAN apod.

2.4.5 Tradiční architektura Cisco

Architektura společnosti Cisco Systems předpokládá, že NetFlow exportér je vždy součástí síťového směrovače, který je tím pádem zodpovědný nejen za směrování v počítačové síti, ale také za sběr NetFlow záznamů a jejich export do NetFlow kolektoru. Toto řešení má ale několik poměrně vážných nevýhod. Tou hlavní je bezesporu vysoká pořizovací cena Cisco směrovačů, která brání jeho masivnímu nasazení v malých a středně velkých počítačových sítích. Samotný sběr dat, jejich analýza a generování NetFlow záznamů také negativně ovlivňuje výpočetní, a tím pádem také směrovací výkon celého zařízení. Proto také většina těchto zařízení využívá na vstupních rozhraních tzv. vzorkování paketů, tzn. že se pro tvorbu NetFlow záznamů využije každý x–tý paket. Toto opatření samozřejmě snižuje přesnost celého měření a dramaticky snižuje pravděpodobnost odhalení bezpečnostních problémů a incidentů.

2.4.6 Moderní architektura – FlowMon

Pro vyřešení výše uvedených problémů bylo nutné zavést úplně novou koncepci sběru NetFlow záznamů. V poslední době se stávají velmi oblíbenými řešení, které využívají tzv. pasivní NetFlow sondy. Takovou sondou je i ta od firmy Invea–tech, pro níž vzniká tato bakalářská práce. Pasivní NetFlow sondy jsou zařízení přímo specializovaná na monitorování přenášených dat a tvorbu a export NetFlow záznamů. Tyto sondy jsou také konstrukčně velmi jednoduché, což je činí velmi levnými. Odstraňují tak všechny výše uvedené nevýhody tradiční architektury společnosti Cisco a na rozdíl od aktivních směrovačů je lze umístit prakticky do libovolného místa počítačové sítě a to transparentně.

Sondy přenášená data pouze sbírají a monitorují, ale nijak jinak do nich nezasahují. Proto se těmto sondám říká pasivní. Exportované statistiky jsou do kolektoru odesílány prostřednictvím dedikované síťové linky. Komunikující zařízení tedy nemá žádné prostředky k tomu, aby zjistilo, že jeho komunikace přes tyto sondy proudí a jsou pro něj tedy transparentní (na L2 vrstvě a vyšších).

Kapitola 3

Vlastní programová implementace

3.1 Obecný popis problému

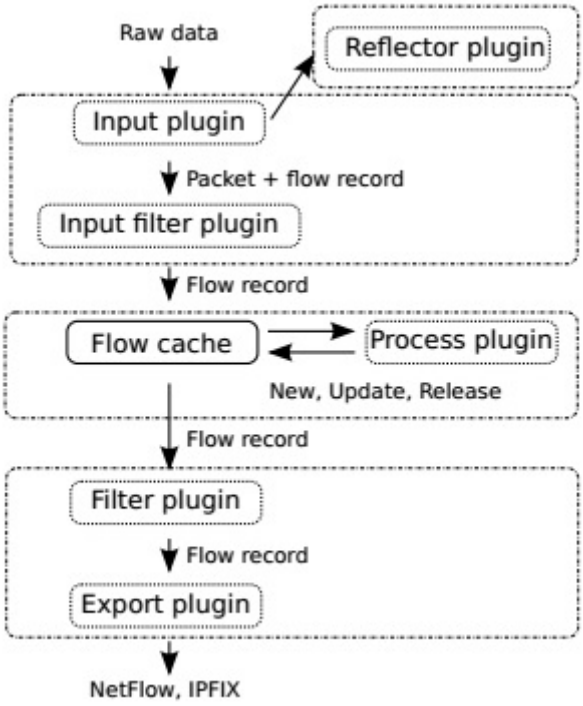
Jádrem celé této práce je zachytávání a analýza provozu protokolů BOOTP, DHCP a DHCPv6 v počítačových sítích. Abych byl úplně přesný, tak na síťové lince, kde je NetFlow sonda instalována. Základním problém bylo porozumění celému systému aplikace FlowMon. Sonda systému FlowMon je koncipována jako plně modulární zařízení, které se skládá ze svého funkčního jádra a mnoha rozšiřujících zásuvných modulů (tzv. pluginů). Programování těchto modulů probíhá plně v čistém jazyce C. Cílem této práce bylo tedy naprogramování modulů pro sledování a analýzu provozu výše zmíněných protokolů v počítačových sítích.

Každý modul se skládá z několika dalších menších částí. Jsou jimi vstupní modul, vstupní filtrační modul, procesní modul, filtrační modul a exportní modul. Řešení pro sledování a analýzu toků používá pouze modul vstupní a procesní. Celé řešení bylo rozděleno na dva samostatné a samostatně použitelné moduly. Jeden pro protokol BOOTP a z něj vycházející DHCPv4 a druhý pro nový, úplně jiný protokol DHCPv6. Toto řešení má výhodu v lepším nasazování(není nutné kontrolovat paket na DHCPv6 příznaky v sítích, kde běží pouze IPv4 a podobně.).

Další nutnou podmínkou bylo porozumění všem protokolům, se kterými bylo nutné v této práci pracovat. Došlo tedy k důkladné analýze a pochopení principů funkčnosti protokolů BOOTP, DHCP a NetFlow. Velká část analýzy byla popsána v kapitole dvě. Další rozšiřující informace budou popsány v následujících odstavcích, které se budou věnovat již tvorbě samotných modulů pro analýzu a export toků sledovaných protokolů. Bylo nutné projít bezpočet nařízení RFC a pochopit principy funkčnosti všech výše uvedených protokolů.

3.2 Architektura sondy FlowMon

NetFlow sonda firmy Invea–tech je zařízení, které má za úkol sledovat síťový provoz a konvertovat sledované protokoly a události do NetFlow záznamů zmíněných v předchozí kapitole. Je to plně modulární systém skládající se z mnoha zásuvných modulů, který podporuje mnoho formátů dat a zároveň jejich zpracování.



Obrázek 3.1: Architektura sondy FlowMon

Sonda má tři hlavní části – vstupní část, část pro práci s vyrovnávací pamětí toků a exportér. Vstupní část, takzvané `input pluginy -- vstupní moduly`, zpracovává zdrojová data a pro každý přijatý paket vytvoří záznam o toku. Jakmile je zpracován paket, který už do nějakého toku patří, existující záznam o toku je aktualizován. Toto zajišťuje právě vyrovnávací paměť v čele s takzvanými `process pluginy - procesními moduly`. Ta pravidelně prohledává záznamy o tocích a kontroluje, zda je datový tok stále aktivní či nikoliv. Jestliže datovému toku vyprší timeout, je vyjmut z vyrovnávací paměti a odeslán k dalšímu zpracování do exportní části sondy, volitelně též do modulu pro filtraci. Ten může rozhodnout, jaké toky budou exportovány. Je možné exportovat pouze toky, které vyhovují požadavkům – například, že komunikace přichází ze specifické IP adresy či probíhá na daném rozsahu portů. Exportní část sondy slouží k samotnému vytvoření NetFlow či IPFIX záznamů, které poté odešle na kolektor. Umožňuje také generování dalších typů výstupů, jako jsou např. CSV soubory.

3.3 Vstupní modul

Vstupní modul je prvním a, dovolím si říct, zároveň nejdůležitějším modulem. Princip jeho činnosti je velmi jednoduchý. Modul vezme příchozí paket a zjistí, zda–li patří jednomu ze sledovaných protokolů. Pokud ano, zpracuje ho, vyčte z něj údaje, ze kterých sestaví flow record a ten pak postoupí ke zpracování procesnímu modulu. Jakmile je vstupní modul načten a inicializován, začíná samotná analýza paketů. Ty mohou přicházet z několika zdrojů. Lze tedy využít i jiný vstup, než je síťová karta. Často je jako vstup využíván také PCAP soubor.

Samotný vstupní modul umožňuje několik metod pro vstup. Mezi ně patří například `plugin_input_get_packet, plugin_input_get_flow, plugin_input_get_final_flow`. Pro moji implementaci byla vybrána metoda `plugin_input_get_flow`, protože je plně dostačující a zároveň pro ni byl k dispozici i jednoduchý ukázkový modul. Každý vstupní modul typu má minimálně tyto 4 základní funkce.

- `PLUGIN_INPUT_DESC`
  - Popis vstupního modulu + funkce nápovědy
  - Definice velikosti rozšíření flow recordu
- `PLUGIN_INPUT_INIT`
  - Inicializace modulu
  - Alokace privátní struktury
- `PLUGIN_INPUT_GET_FLOW`
  - Hlavní funkce celého modulu
  - Vyčítání a zpracování dat do flow recordu
- `PLUGIN_INPUT_SHUTDOWN`

Jelikož pracujeme s NetFlow verze 10, čili IPFIX, je možné díky flexibilnímu formátu jakékoliv rozšíření samotného NetFlow záznamu o další uživatelské položky. Toto umožňuje právě vstupní zásuvný modul, který definuje rozšiřující strukturu. Dále definuje privátní strukturu modulu a také definuje takzvané `gettery`. Gettery slouží pro definici množiny položek, se kterými pak pracuje samotný export. Jsou to vlastně údaje, které nás zajímají a které jsou exportovány do kolektoru.

3.3.1 Vstupní modul pro BOOTP a DHCPv4

Jak již bylo řečeno v kapitole dvě, protokoly BOOTP a DHCPv4 jsou spolu kompatibilní, protože DHCP vychází primárně právě z BOOTP. Z toho důvodu bylo možné sjednotit analýzu obou protokolů do jednoho vstupního modulu. Z hlediska zpracování dat se protokoly liší pouze tím, jak dlouhý samotný paket je. Tím není myšlena ethernet hlavička ani IP hlavička, ale délka dat přímo v těle paketu.

Prvním krokem při analýze paketu je nutnost zjistit, zda je paket vůbec validní a má validní ethernetovou i IP hlavičku. Pokud ano, jsou obě tyto hlavičky přeskočeny a zahozeny. Celkem je tedy zahozeno 14 bytů ethernetové hlavičky a 19 bytů IPv4 hlavičky. Jestliže se na datové vrstvě používá technologie VLAN, je přeskočeno i VLAN rozšíření, čili ethernetová hlavička nebyla přeskočena o 14, ale o 18 bytů. Předposlední zpracovávanou hlavičkou je

hlavička IPv4. Její délka není vždy stejná a její skutečná délka je tak dynamicky zjištěna vyčtením hodnoty položky `IHL` přímo z IPv4 hlavičky a jejím vynásobením čtyřmi. Jelikož není nutné zpracovávat z IPv4 hlavičky žádné informace, je možné ji opět hned přeskočit o vypočtený počet bytů. Další zpracovávanou hlavičkou je hlavička čtvrté vrstvy. Jelikož oba aplikační protokoly používají jako transportní protokol UDP, jeho hlavička je také přeskočena přesně o její stanovenou délku 8 bytů. Jelikož ani jeden z protokolů nepoužívá na transportní vrstvě protokol TCP, není nutné jej vůbec řešit.

V tomto okamžiku je kompletně naplněn standardní FlowMon záznam, který byl popsán v kapitole 2 a program je tak připraven k syntaktické analýze samotných dat. Jelikož je základní stavba dat shodná pro oba protokoly, stačí zkopírovat data z aktuálního ukazatele do paměti do nově definovaného paměťového prostoru struktury `dhcp_v4_record_t`, která je předem známá a je definovaná v hlavičkovém souboru `input--dhcp--v4.h`. Kopíruje se přesně tolik bytů dat, jaká je délka této struktury. Tento přístup má samozřejmě nevýhodu v tom, že struktura `dhcp_v4_record_t` je rozšířena i o jednotlivé sledované DHCP volby, čili samotná struktura je delší než užitečná data v paketu. To má za následek incializaci všech dalších položek struktury nevalidními daty. Toto ovšem ničemu nevadí, jelikož DHCP volby budou zpracovávány hned v následujícím kroku a nevalidní data přepíšou data nová, validní.

V této struktuře jsou nyní uloženy všechny základní údáje o DHCP komunikaci, jako je operační kód, IP adresa klienta, IP adresa serveru, HW adresa klienta a další, které byly popsány v kapitole 2. Tato data ovšem nestačí k tomu, aby bylo vůbec jasné, jaký typ protokolu byl zpracován. Tento problém vyplývá už se samotné stavby a podobnosti obou protokolů. Jediným možným řešením detekce typu protokolu je zpracování rozšířených voleb, čili možností. Každý protokol má na rozšiřující volby vyhrazenou jinak velkou část paketu. Na základě toho však nelze detekovat, zda byl zachycen paket protokolu BOOTP nebo paket DHCP. Jediným možným řešením je hladní možnosti číslo 53, která jasně říká, že byl zachycen DHCPv4 paket a o jaký typ požadavku se jedná. Pokud taková možnost není v paketu nalezena, jedná se vždy o paket BOOTP. V dalších částech této zprávy bude rozebírán převážně protokol DHCP, o jehož sledování stojí firma Invea–tech nejvíce.

S konzultanty z firmy Invea–tech bylo domluveno, že modul bude podporovat minimálně tyto rozšiřující možnosti DHCP.

- `Možnost 1 -- přidělenou masku sítě`
- `Možnost 3 -- všechny přidělené IP adresy výchozích směrovačů`
- `Možnost 6 -- všechny přidělené IP adresy serverů DNS`
- `Možnost 4 a 42 -- všechny přidělené IP adresy serverů NTP`
- `Možnost 50 -- klientem poptávaná IP adresa`
- `Možnost 51 -- čas, po který může klient konfiguraci používat`
- `Možnost 53 -- typ DHCP požadavku`

Možnost 53 je také velmi důležitá ke stanovení kontextu celé komunikace. Pokud bychom neznali hodnotu tohoto pole, nelze odhalit, jaký typ DHCP komunikace klient provádí. Nebylo by možné rozhodnout, zda klient žádá o IP adresu, nebo informuje ostatní o používání získané konfigurace, zda se vzdává své konfigurace či zda žádá o znovuobnovení konfigurace

a prodloužení doby její platnosti. Jelikož je ale možnost 53 známá, je poměrně jednoduché rozhodnout, o jaký typ toku se jedná.

Ostatní možnosti jsou přeskočeny tím způsobem, že je zkontrolováno číslo možnosti a pokud neodpovídá žádnému hledanému, je možnost přeskočena přesně o číslo, které v sobě má další bajt v paměti a které říká, kolik bajtů má datová oblast možnosti. Samozřejmě je nutné přeskočit ještě o dva bajty více – bajt s číslem možnosti a bajt, jež udává samotnou délku datové oblasti možností. Možnosti jsou prohledávány a zpracovávány tak dlouho, dokud není zpracován celý paket nebo není zpracována možnost číslo 255, která označuje konec variabilní části s možnostmi. Toto napovídá, že není problém modul kdykoliv rozšířit o zpracovávání dalších potřebných údajů z DHCP. Stačí pouze přidat správnou položku do rozšiřující datové struktury a implementovat reakci na číslo DHCP možnosti, kterou budeme chtít nově zpracovávat. Modul je tedy velmi dobře a velmi snadno škálovatelný.

Jakmile je dokončeno parsování celého paketu, je ze získaných údajů vypočítán hash, který jednoznačně identifikuje IP tok, který byl právě zpracován. Při práci s DHCP však bylo nutné poskytnout hashovací funkci více informací, než ve standardním IP toku. Pro správný výpočet hash se tedy používá následujících údajů:

- `Zdrojová IPv4 adresa`
- `Cílová IPv4 adresa`
- `Zdrojový L4 port`
- `Cílový L4 port`
- `Číslo L4 portu`
- `MAC adresa klienta -- položka chaddr z DHCP`

Pro správnou funkci modulu bylo nutné přidat do identifikace toku i MAC adresu komunikujícího klienta. Toto opatření bylo nutné z toho důvodu, že významná část komunikace těchto protokolů probíhá pomocí všesměrového vysílání s cílovou IP adresou 255.255.255.255 a zdrojovou IP adresou 0.0.0.0, což znemožňovalo identifikovat, který počítač právě komunikuje. Tento problém se ale vyřešil právě pomocí přidání MAC adresy klienta přímo z komunikace DHCPv4/BOOTP a díky tomu je možné bez problémů identifikovat veškerou DHCPv4/BOOTP komunikaci a zpracovat ji. Tato položka je vždy vyplněna a vždy obsahuje právě MAC adresu klienta, který se serverem DHCPv4/BOOTP komunikuje.

3.3.2 Vstupní modul pro DHCPv6

Zpracování protokolu DHCPv6 začíná stejně jako v předchozím případě nejdříve pomocí vstupního zásuvného modulu. Prvním krokem při analýze IPv6 paketu je opět nutnost odstranění ethernetové hlavičky a pokud je použita technologie VLAN, je nutné odstranit i její hlavičku. Přeskočení těcho hlaviček je provedeno úplně stejně, jako v modulu pro DHCPv4, čili není nutné ho dále rozebírat, protože je dostatečně vysvětleno již v předcházející sekci. Čím se ale zásadně zpracování liší je zpracování IPv6 hlavičky. Ta má na rozdíl od hlavičky IPv4 konstantní velikost. Bohužel, vývojáři IPv6 implementovali do tohoto protokolu takzvané rozšířené hlavičky. To není nic jiného, než lineárně vázaný seznam položek za sebou. Nelze zpracovat jinak, než projitím každé rozšířené hlavičky, zjitěním její délky z položky, která to udává a rozšířenou hlavičku přeskočit. Ještě předtím je však načteno ID následující hlavičky, aby bylo jasné, která se bude zpracovávat. Přeskakování se provádí tak dlouho,

dokud se nedojde na rozšířenou hlavičku UDP, která je také přeskočena. Za ní již následují samotné DHCPv6 data.

Jak již bylo řečeno v teoretické části práce, DHCPv6 neobsahuje žádnou pevnou strukturu paketu. Místo toho může obsahovat množství voleb, které se musí zpracovávat. Samotné zpracování je opět velice jednoduché, díky absenci základní struktury paketu daleko jednodušší, než v DHCPv4. Jako v případě tohoto protokolu, tak i v DHCPv6 je sledováno ID zpracovávané možnosti a její délka. Podle čísla možnosti lze pak jednoduše určit, co za data se zpracovává. Ty jsou pak ukládány do struktury `dhcp_v6_record_t`.

S konzultanty z firmy Invea–tech bylo domluveno, že modul bude podporovat minimálně tyto možnosti DHCPv6.

- `Možnost 1 -- Identifikátor klienta -- DUID`
- `Možnost 2 -- Identifikátor serveru -- DUID`
- `Možnosti 3 a 5 -- přidělená IPv6 adresa včetně času, po který je adresa platná`
- `Možnost 23 -- všechny přidělené IPv6 adresy serverů DNS`
- `Možnost 56 -- všechny přidělené IPv6 adresy serverů NTP`

Možnosti jsou opět prohledávány a zpracovávány tak dlouho, dokud není zpracován celý paket. Stejně jako v DHCPv4, tak i implementace DHCPv6 je napsána tak, že je modul možné kdykoliv rozšířit o zpracování dalších potřebných údajů a DHCPv6 možností. Stačí pouze opět přidat správnou položku do rozšiřující datové struktury a implementovat reakci na číslo DHCPv6 možnosti, kterou budeme chtít nově zpracovávat. Modul je tedy zase velmi dobře a velmi snadno škálovatelný.

Po ukončení parsování celého paketu je ze získaných údajů opět vypočítán hash, který jednoznačně identifikuje IP tok, do kterého zpracovávaný paket patří. Jelikož je však DHCPv6 jednodušší a elegantnější protokol, než starý DHCPv4, není nutné přidávat žádnou další položku do hashovací funkce. Je to dáno tím, že jak bylo psáno výše v teoretické části, DHCPv6 nepoužívá neurčité adresy typu 0.0.0.0 nebo 255.255.255.255, ale vždy je možné dle IPv6 adres určit, kdo s kým komunikuje. Hash je tedy počítán z těchto údajů:

- `Zdrojová IPv6 adresa`
- `Cílová IPv6 adresa`
- `Zdrojový L4 port`
- `Cílový L4 port`
- `Číslo L4 portu`

3.4 Procesní modul

Procesní modul představuje v podstatě rozšíření vstupního modulu o zpracovávání dalších paketů v již identifikovaném toku. Tato funkcionalita je možná díky tomu, že procesní modul je volán až z vlákna, které obstarává vyrovnávací paměť otevřených IP toků. Slouží tedy primárně pro práci s již vytvořeným záznamem, který vytvořil právě vstupní modul. Záznam je možné upravovat a vkládat do něj, či odebírat z něj, další získané informace. V mém případě je použito pouze vkládání dalších informací. Samotné ukládání je opět provedeno do patřičné privátní struktury. Uložení je provedeno ať se již jedná o nově získané informace nebo o aktualizaci předchozích informací. V rámci práce byl vytvořen modul zvlášť pro DHCPv4/BOOTP a zvlášť pro DHCPv6. Aby bylo možné programovat jednoduše a efektivně, je procesní modul vždy v rámci jedné sdílené knihovny pohromadě s daným vstupním modulemTato práce popisuje postupy pro sledování provozu síťových protokolů BOOTP, DHCP pro IPv4 a DHCP pro IPv6 pomocí netflow sondy FlowMon od společnosti Invea-tech. Je zde nastíněna problematika těchto protokolů, funkčnost sondy FlowMon, obecný popis NetFlow a vlastní popis řešení pro sběr a vyhodnocení dat. Byla provedena důkladná analýza a poté byly sepsány moduly pro sondu FlowMon pro možnost monitoringu zmíněných protokolů. Jejich implementace, způsob testování a vyhodnocení získaných dat je v této práci popsán. Tento přístup umožnil ukládat data přímo do privátní struktury vstupního modulu a nezatěžovat tak dále paměť. Jelikož principiálně je funkcionalita obou procesních modulů naprosto stejná, nebudu každý z nich popisovat zvlášť, ale popis bude proveden souhrně.

Funkce i stavba procesního modulu je však oproti vstupnímu velmi jednoduchá a nenáročná. Díky tomu nebylo nutné implementovat téměř všechny jeho funkce. Pro správnou funkci modulu bylo nutné implementovat pouze 2 funkce – `plugin_proc_desc` pro výpis nápovědy a infa o modulu a funkci `plugin_process_update`, která zařídí veškeré aktualizace privátní struktury modulu.

Samotná funkce `plugin_process_update` je velmi jednoduchá. Spouští se vždy, pokud příjde paket, který patří do daného IP toku. Jejím hlavním účelem je pouze spustit jednotlivé validační funkce, které se používají pro validaci dat již ve vstupním modulu, tentokrát však nad starými a novými daty. Pokud validační funkce vyhodnotí, že nová data mají nahradit stará, nahradí je. Pokud ne, nic se neděje a v privátní struktuře zůstanou data původní.

Validační funkce vždy vezmou stará data, podívají se na typ DHCP message a pokud se v daném DHCP message může daná položka měnit, změní ji. Jestliže ne, automaticky zůstávají stará data.

3.5 Finální zpracování dat a jejich export

Jakmile je spočítán hash IP toku, tak je předán dále k dalšímu zpracování procesnímu modulu, přesněji vyrovnávací paměti. V tu chvíli se dostává ke slovu patřičný procesní modul, jehož obecná funcionalita je popsána v předchozí sekci. Sběr a vyhodnocení dat je aktivní tak dlouho, dokud nevyprší čas, po který má být tok sledován nebo dokud není tok ukončen ze strany klienta. Princip aktivního a pasivního čekacího času je vysvětlen v kapitole 2.4.3, proto se mu zde již nebudu věnovat.

Po ukončení toku a analýze dat jsou získané údaje předávány dále do exportní části sondy pomocí externě definovaných položek, tzv. getterů. Ty říkají exportnímu modulu, které informace z privátní struktury si má či nemá vzít a poslat je na IPFIX kolektor.

Každá položka je před „vyzvednutím“ getterem validována a správně vyplněna. Na to slouží v každém modulu několik validačních funkcí – pro každou exportovanou položku jedna. Ty prověří pravidla, zda může být daná položka prohlášena za validní. V mé práci jsou tato pravidla sestavena na základě RFC, která definují jednotlivé zkoumané protokoly. Validační funkce pak zjišťují, zda se daná BOOTP/DHCPv4/DHCPv6 položka, kterou getter exportuje, může vyskytovat v typu BOOTP/DHCPv4/DHCPv6 zprávy, která je aktuálně zpracovávána. Například, zda může být vyplněna položka `Client IP address -- ciaddr` v DHCPv4 zprávě `Advertise`. Pokud položka validní není, neexportuje se. Jestliže však validní je, je sděleno getteru, jak je položka velká a proběhne předání dat z privátní struktury. Jelikož getter neumí zpracovávat datový typ pole, bylo nutné některé položky zredukovat a předávat je postupně pomocí několika getterů – typicky IP adresy DNS serverů nebo výchozích bran. Samotný modul tedy nepředává celou privátní strukturu, ale pouze jen některé její části.

Jak bylo již popsáno v předchozí sekci, validace exportovaných položek probíhá také v procesním modulu. Zde se neporovnává pouze kontext přijaté zprávy a zda může být daná položka v tělě zpracovávané BOOPT/DHCPv4/DHCPv6 zprávy, ale i to, zda může být stará hodnota dané položky přepsána hodnotou novou. V této části tedy nedochází k validaci při exportu, ale při aktualizaci dat v privátní struktuře. Pokud k přepsání dojde, jedná se vždy o přepsání nedefinované, tedy nevalidní hodnoy na hodnotu validní. Z principu všech zkoumaných protokolů nemůže dojít k přepsání validní položky nevalidní hodnotou.

3.5.1 Exportovatelné položky v rámci modulu pro BOOTP/DHCPv4

V současné době může modul exportovat tyto položky:

- `Adresa klienta -- položka ciaddr -- ip adresa komunikujícího klienta (např. obnovení adresy)`
- `Adresa serveru -- položka siaddr -- ip adresa komunikujícího serveru`
- `Vaše IP adresa -- položka yiaddr -- ip adresa, kterou server přiděluje klientovi (např. přidělení nové adresy)`
- `Adresa relay -- položka giaddr -- ip adresa agenta pro přeposílání DHCP zpráv`
- `MAC adresa klienta -- položka chaddr -- MAC adresa klienta komunikujícího se serverem DHCP`
- `Maska podsítě -- přídělená maska podsítě -- option 1`
- `Požadovaná IP adresa -- IP adresa, kterou požaduje klient přidělit (např. při žádosti o novou IP adresu -- option 50)`
- `Čas zapůjčení IP adresy -- čas, po který může klient IP adresu používat -- option 51`
- `IP adresy výchozích bran -- maximálně 2 IP adresy výchozí brány ze sítě -- option 3`
- `IP adresy serverů DNS -- maximálně 4 IP adresy DNS serverů v síti -- option 6`

- IP adresy serverů NTP -- maximálně 2 IP adresy serverů času -- option 4 a 42
- Typy DHCP zpráv v toku -- bitové vyjádření zpráv, které byly v rámci toku zaslány

Jak je vidět, modul umí ve svém základu zjistit a vyfiltrovat poměrně mnoho informací. Přidaným elementem je pak 8 bitový little endian vektor zjištěných DHPC zpráv v toku. Pokud je bit nastaven na 1, daný typ zprávy byl v toku zachycen. Naopak, pokud je bit roven 0, daná zpráva zachycena nebyla. Význam jednotlivých bitů je vysvětlen zde:

- Bit 0 -- DHCP Discover
- Bit 1 -- DHCP Offer
- Bit 2 -- DHCP Request
- Bit 3 -- DHCP Decline
- Bit 4 -- DHCP Ack
- Bit 5 -- DHCP Nak
- Bit 6 -- DHCP Release
- Bit 7 -- DHCP Inform

Toto pořadí se shoduje s tím z RFC 2132[5]. Vektor s informacemi například 00010111 tedy znamená, že v rámci toku byly zachyceny zprávy Discover, Offer, Request a Ack. Jedná se tedy s největší pravděpodobností o přidělení IP adresy úplně novému klientovi, který zatím žádnou IP adresu neměl.

3.5.2 Exportovatelné položky v rámci modulu pro DHCPv6

Jelikož DHCPv6 je daleko méně používaným protokolem, je i počet exportovatelných položek menší. V době psaní této práce umožňuje modul exportovat následující položky:

- Identifikátor klienta -- DUID -- option 1
- Identifikátor serveru -- DUID -- option 2
- Přidělená IPv6 adresa včetně času, po který je adresa platná -- option 5
- Všechny přidělené IPv6 adresy serverů DNS -- option 23
- Všechny přidělené IPv6 adresy serverů NTP -- option 31 a 56
- Typ DHCPv6 komunikace -- stavová nebo bezstavová
- Typy DHCPv6 zpráv v toku -- bitové vyjádření zpráv, které byly v rámci toku zaslány

První informací navíc je určení typu DHCPv6 komunikace. Zda se jedná o komunikaci stavovou nebo bezstavovou. Pokud je příznak nastaven na hodnotu 1, jedná se o bezstavovou komunikaci. Stavová je signalizována hodnotou 2. I v tomto případě byl přidán pomocný little endian bitový vektor, který udává, které DHCPv6 zprávy byly v rámci toku zachyceny. Není však 8 bitový, jak tomu bylo v předchozím případě, ale 16 bitový, jelikož modul reaguje celkem na 13 typů DHCPv6 zpráv. Význam jednotlivých bitů je popsán zde:

- `Bit 0 -- DHCPv6 Solicit`
- `Bit 1 -- DHCPv6 Advertise`
- `Bit 2 -- DHCPv6 Request`
- `Bit 3 -- DHCPv6 Confirm`
- `Bit 4 -- DHCPv6 Renew`
- `Bit 5 -- DHCPv6 Rebind`
- `Bit 6 -- DHCPv6 Reply`
- `Bit 7 -- DHCPv6 Release`
- `Bit 8 -- DHCPv6 Decline`
- `Bit 9 -- DHCPv6 Reconfigure`
- `Bit 10 -- DHCPv6 Information-request`
- `Bit 11 -- DHCPv6 Relay-forward`
- `Bit 12 -- DHCPv6 Relay-reply`

Toto pořadí se opět shoduje s RFC popisujícím DHCPv6, tedy s `RFC 3315`[6]. Vektor s informacemi například `0000000001000111` tedy znamená, že v rámci toku byly zachyceny zprávy Solicit, Advertise, Request a Reply. Jedná se opět s největší pravděpodobností o přidělení nové IPv6 adresy úplně novému klientovi, který opět žádnou IPv6 adresu neměl. Funkcionalita je tedy stejná jako v rámci BOOTP/DHCPv4.

3.6 Zapojení do systému

Zapojení do celkového systému je opět velmi jednoduché. Po překladu programu vzniknou dvě samostatné sdílené knihovny. Jedna modulu pro DHCPv4, druhá modulu pro DHCPv6. V každé z nich je jak příslušný vstupní modul, tak i procesní modul. Následující ukázky spouštění byly vytvořeny přímo na dodané testovací sondě stažené přímo ze serverů Invea–tech.

Pokud má být modul zapojen do systému sondy, je nutné ho před spuštěním nahrát pomocí přepínače –X, kterým se nahraje příslušná sdílená knihovna a –I, který určuje název vstupního modulu včetně zdroje dat. Celá syntaxe spouštění těchto daných modulů pak bude vypadat takto:

- DHCP v4 a vstup ze souboru pcap
```
sudo flowmonexp -X /home/flowmon/dhcp_v4/input-dhcp-v4.so -I \
input-dhcp-v4:pcap_file=/home/flowmon/dhcp_v4/test1.pcap \
-X /lib64/flowmonexp/flowmon-export-ipfix.so \
-E ipfixx:host=localhost
```

- DHCP v4 a vstup z rozhraní eth0
```
sudo flowmonexp -X /home/flowmon/dhcp_v4/input-dhcp-v4.so -I \
input-dhcp-v4:pcap_if=eth0 -X /lib64/flowmonexp/flowmon-export-ipfix.so
\
-E ipfixx:host=localhost
```

Jelikož spuštění modulu pro DHCPv6 je analogické, není třeba ho zde explicitně uvádět. Takto spuštěný modul se tedy nahraje do FlowMon sondy a zpracuje každý paket, který sonda zachytí. Samozřejmě, filtruje pouze informace, které ho zajímají. Pokud paket neodpovídá žádanému provozu, modul ho dále nijak nezpracovává. Toto spuštění sondy však nahraje pouze vstupní modul. Pro spuštění i procesního modulu je nutné přidat další přepínač –P, který sondě řekne, že má zapojit právě i procesní modul a jaký má tento modul název. Výsledný příkaz pro spuštění pak bude vypadat takto:

- DHCP v4 a vstup ze souboru pcap – včetně procesního modulu
```
sudo flowmonexp -X /home/flowmon/dhcp_v4/input-dhcp-v4.so \
-I input-dhcp-v4:pcap_file=/home/flowmon/dhcp_v4/test1.pcap \
-P process-dhcp-v4 -X /lib64/flowmonexp/flowmon-export-ipfix.so \
-E ipfixx:host=localhost
```

- DHCP v4 a vstup z rozhraní eth0 – včetně procesního modulu
```
sudo flowmonexp -X /home/flowmon/dhcp_v4/input-dhcp-v4.so \
-I input-dhcp-v4:pcap_if=eth0 -P process-dhcp-v4 \
-X /lib64/flowmonexp/flowmon-export-ipfix.so -E ipfixx:host=localhost
```

Spuštění modulu pro DHCPv6 je opět analogické, čili opět ho nebudu uvádět. Během své činnosti sonda sbírá data, které pak jednotlivé moduly zpracovávají a předávají exportéru, který je odesílá do IPFIX kolektoru. Oba dva moduly odesílají data vždy, jakmile sonda usoudí, že daný IP tok skončil. Jakmile sonda prohlásí IP tok za ukončený, exportér převezme všechna validní data, která modul vytvořil a odešle je do některého z IPFIX kolektoru či na více kolektorů.

Kapitola 4

Testování funkčnosti a sběr dat

Testování každého produkčního nástroje je vždy velmi důležitým prvkem ve vývoji produktu. Jelikož v této práci se jedná o produkt pro zařízení v počítačových sítích, bylo náročnější zřídit testovací prostředí pouze pro vývoj. Testovací počítače a nasazení FlowMon sondy nakonec proběhlo v prostorech dvou středních škol, ve kterých spravuji celou počítačovou síť. Vývoj probíhal v prostředí `Obchodní akademie v Trutnově`, kde jsou již použité VLAN sítě a bylo tedy jednoduché testovat moduly pouze na omezeném počtu počítačů. Samotné testovací nasazení potom probíhalo v síti bez používání VLAN v prostředí `Klasického a španělského gymnázia, Brno--Bystrc, Vejrostova 2`. Pro otestování této aplikace bylo tedy prostředí velké sítě přesně tím, co bylo potřeba.

Ukázalo se, že testování v průběhu vývoje bylo velmi důležitým aspektem pro celou práci a odhalilo mnoho nečekaných chyb – například chybu ukádání dat little endian vs. big endian při vyčítání některých DHCP možností. V následujících částech budou rozebrány detaily testování.

4.1 Laboratorní simulace

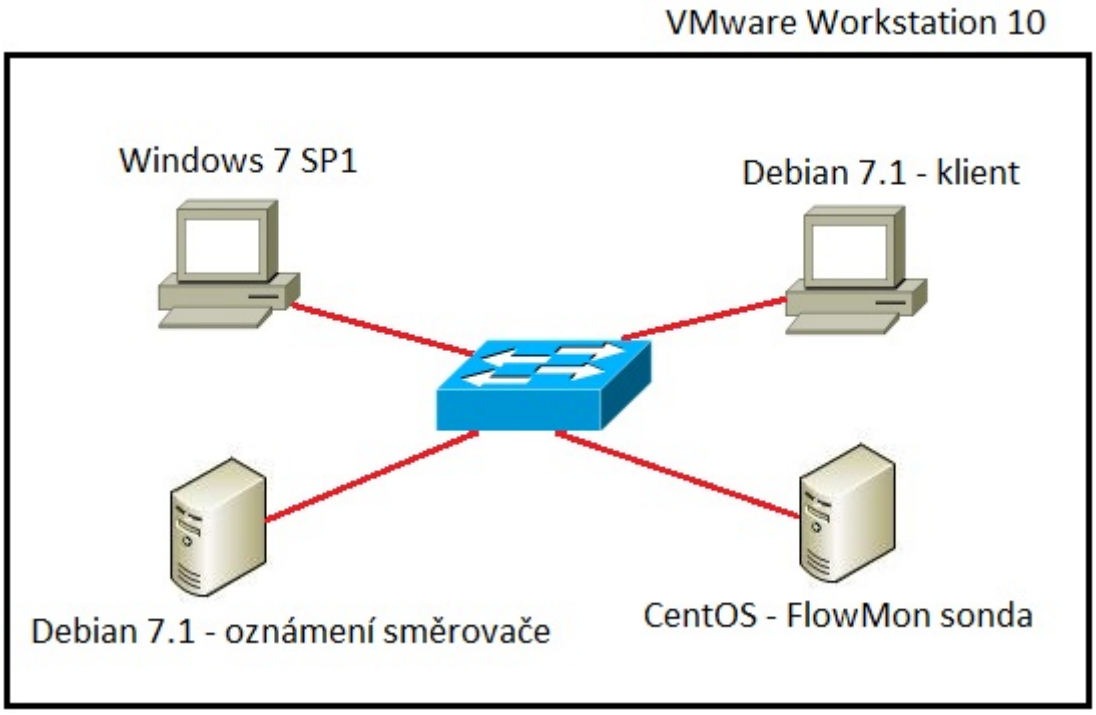
Laboratorní simulace probíhala přímo na mém domácím počítači ve virtualizačním prostředí `VMware Workstation 10`. Pro testování byl stažen obraz pevného disku počítače s operačním systémem CentOS s již nainstalovanou a plně funkční sondou FlowMon. Tento obraz byl dostupný přímo z webu firmy Invea–tech. Můj počítač byl v následující konfiguraci:

- CPU – Intel Core i5-4570
- RAM – 32 GB
- GPU – Gainward GeForce GTX 660 – pro význam testu nepodstatné
- Základní deska – Asus H87M–E
- HDD – SSD Crucial M500-480 GB

4.1.1 Popis a schéma testovací sítě

Následující obrázek ukazuje velmi jednoduché schéma testovací sítě. Obsahuje pouze server s instalovanou sondou FlowMon, testovací počítač s operačním systémem Debian 7.1, testovací počítač s operačním systémem Windows 7 SP1 a počítač pro vysílání oznámení

směrovače, opět se systémem Debian 7.1. Všechny počítače sdílely 1 virtuální přepínač, který byl nastaven tak, aby veškerý provoz přeposílal na port, na kterém byl připojen server s instalovanou sondou. Každý počítač měl k dispozici jedno virtuální jádro fyzického procesoru a dva GB operační paměti. Pro účely testování byl vybrán adresní rozsah `192.168.1.0/24` pro IPv4 a `2001:1111:2222:3333::/64` pro IPv6.



Obrázek 4.1: Testovací prostředí laboratorní simulace

4.1.2 Postup simulace a popis ladění programu

Jakmile bylo vytvořeno simulační prostředí na mém domácím počítači, začal jsem veškeré naprogramované kroky ihned testovat. To umožnilo celkově rychlejší vývoj, protože nebylo nutné případné chyby náročně hledat. Jejich objevení bylo jednoduché, stejně tak jejich oprava. Nasazení na domácím počítači umožnilo i práci bez připojení k síti Internet, což se několik dní vyplatilo, protože poskytovatel mého připojení měl se svojí sítí problémy a připojení tak bylo velmi nestabilní.

Po každé změně ve zdrojovém kódu jednotlivých modulů byl proveden první test právě v simulačním prostředí, kdy byla právě naprogramovaná funkcionalita ihned ověřena. Pro ověření funkčnosti vždy sloužily nejen ladící výpisy na standardní výstup, ale hlavně program `Wireshark`, který je mezi síťovými specialisty velmi oblíbený díky tomu, že dokáže dekódovat téměř veškerou síťovou komunikaci a zobrazit ji v uživatelsky přívětivé formě. Tento program mnou byl využíván velmi často, aby bylo možné zkontrolovat, zda nasbíraná data souhlasí s těmi, které se podařilo odchytit pomocí Wiresharku.

Nejdříve byl vytvořen kompletní vstupní modul pro BOOTP/DHCPv4. To mi umožnilo důkladně pochopit celou funkčnost sondy FlowMon i toho, jak se sonda programuje. Po odladění vstupního modulu byl naprogramován vstupní modul pro DHCPv6. Jeho programování bylo však o dost jednodušší a časově méně náročné, protože stačilo částečně přepsat

kód vstupního modulu BOOTP/DHCPv4.

Po důkladném otestování obou vstupních modulů byly doprogramovány pouze procesní moduly, jejichž funkčnost byla opět ověřena porovnáním zachycených dat sondou s těmi, které se podařilo odchytit ve Wiresharku. Na stejném prostředí byl i vyvíjen modul pro grafické zobrazení nasbíraných dat a jejich analýzu. Ten je více rozvinut v kapitole 5, proto ho zde zatím nebudu dále popisovat. Testovací prostředí se poté využívalo pouze v tu chvíli, kdy byla v kódu objevena nějaká chyba, na kterou jsem přišel při testování v reálném nasazení.

4.2 Nasazení v reálném provozu

Jak bylo řečeno v úvodu, testování v reálném prostředí bylo prováděno ve dvou reálných počítačových sítích. Ta na Obchodní akademii v Trutnově sloužila pouze pro prvotní otestování na malém vzorku dat a málo počítačích. Samotné reálné nasazení proběhlo až v prostředí počítačové sítě Klasického a španělského gymnázia, Brno–Bystrc, Vejrostova 2. Tato síť měla tu výhodu, že se v ní zatím nepoužívala technologie VLAN a bylo tak možné jednoduše sledovat veškerý provoz zkoumaných protokolů. Druhou výhodou této sítě byl i vysoký počet uživatelů a velký provoz všech sledovaných protokolů.

4.2.1 Popis reálné sítě pro nasazení

Pro reálné nasazení bylo opět nutné vybrat adresní rozsah pro testování. Obě reálné sítě používaly shodný adresní rozsah pro IPv4 – `192.168.1.0/24`. I proto byl tento rozsah vybrán pro laboratorní simulace.

Pro testování IPv6 byl opět zvolen rozsah `2001:1111:2222:3333::/64`, který byl společný v obou sítích.

Tu na Obchodní akademii v Trutnově sice tvoří 12 přepínačů firmy Cisco, cca 120 počítačů a mnoho dalších zařízení jako VoIP telefony či směrovače, ale pro účely testování byla použita pouze jedna VLAN síť s 24 počítači s operačním systémem `Windows 7 SP1` s různou hardwarovou konfigurací a různých výrobců a jedním přepínačem firmy `Cisco`, typ `C2960--24`. Dále byl pro potřeby vývojového testu vyhrazen jeden virtuální server, kde byla spuštěna FlowMon sonda, která monitorovala veškeré dění v dané VLAN.

Síť Klasického a španělského gymnázia byla tvořena 16 přepínači také značky Cisco, taktéž typu C2960–24, cca 140 počítači různé konfigurace a s různými operačními systémy, přes 70 IP telefonů Cisco 7912 a několik síťových tiskáren různých značek.

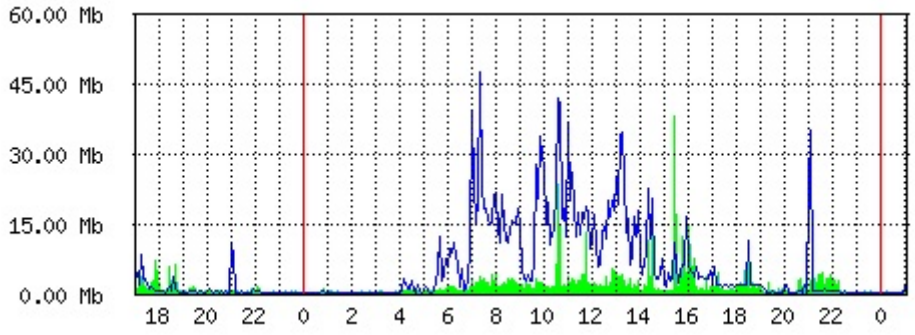
V obou sítích existovala pouze jedna výchozí brána, ale dva servery DNS a dva servery NTP. V každé síti byl dále přítomen jeden server DHCPv4 a jeden server DHCPv6. Ten půlku testovací doby fungoval v režimu bezstavové konfigurace, v němž si počítače generovaly IPv6 adresu samy a půlku doby v režimu stavové konfigurace, kdy IPv6 adresu přiděloval sám server.

Je jasně vidět, že oproti síti obchodní akademie je síť gymnázia o mnoho větší. Toho bylo využito pro závěrečné testování, které mělo ukázat, zda je modul možný opravdu reálně používat.

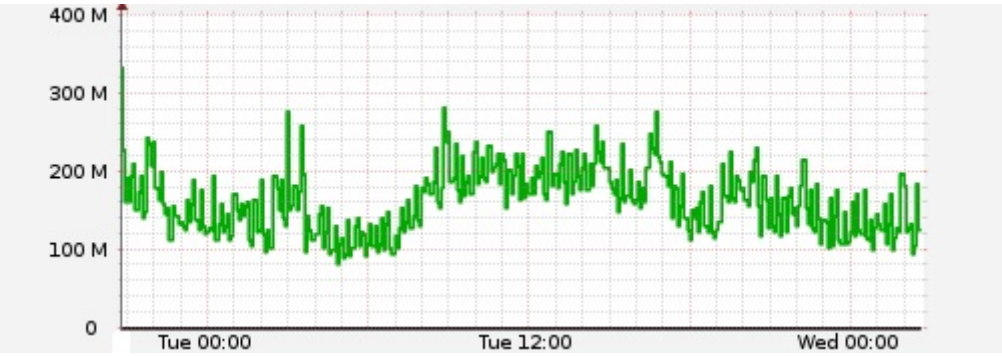
4.2.2 Průběh a problémy při nasazování a testování

Hlavním cílem testu v reálném prostředí bylo ověření funkčnosti vytvořených modulů pod větší, reálnou zátěží. Z toho důvodu byl na každé škole vyčleněn virtuální server, na němž byla opět zprovozněna FlowMon sonda a zároveň na něj byl přesměrován veškerý provoz z páteřního síťového přepínače. To umožnilo sbírat opravdu všechna data, která přes centrální přepínač procházela a důkladně tak ověřit správnost kódu a výkonnost implementace. Díky tomuto řešení jsem byl schopen zjistit, zda moduly dokáží zpracovat stovky megabitů provozu a přitom generovat správné a plně validní IPFIX záznamy nebo implementace neskočí předčasně například díky chybě `Segmentation fault`.

Z toho důvodu bylo testování v reálném provozu provedeno ve dvou odlišných sítích s různým počtem připojených klientů. Zatímco na Obchodní akademii v Trutnově dosahoval páteřní provoz ve špičce pouze desítek megabitů, v prostředí Klasického a španělského gymnázia byl ve špičce měřen provoz řádově ve stovkách megabitů. Ověřování probíhalo celkem 14 dní, během kterých byly samozřejmě postupně opravovány poslední chyby. Pro představu uvádím i grafy znázorňující datový tok na server v rámci jednoho dne.



Obrázek 4.2: Graf přenosů páteřní sítě OA Trutnov



Obrázek 4.3: Graf přenosů sítě Klasického gymnázia Bystrc

V rámci testování bylo každý den zpracováno až 80 tisíc DHCP a 10 tisíc DHCPv6 paketů, které sonda dokázala díky mé implementaci bez problémů zpracovat. Na bystrckém gymnáziu také v týdnu, z něhož pochází graf měření, probíhalo stěhování virtuálních serverů mezi fyzickými servery a byla prováděna též instalaci některých klientských počítačů přes počítačovou síť. Proto docházelo k velkému kopírování dat, která tekla přes centrální přepínač, čili i to ovlivnilo graf přenosů a zatížilo tak sondu včetně mnou vytvořených modulů.

Během testu nezhavaroval žádný z programovaných modulů a to mi umožnilo prohlásit programovou implementaci za dostatečně odladěnou. Jako možné rozšíření do budoucna se jistě nabízí možnost otestování stability a výkonosti mého řešení i v daleko větších sítích.

Kapitola 5

Analýza nasbíraných statistik

Součástí práce bylo také vytvoření jednoduchého webového rozhranní, které bude sloužit pro jednoduchý a rychlý přehled o statistikách a dění na síti, které se týká zkoumaných protokolů. Rozhranní bylo napsáno pomocí běžných prostředků pro tvorbu webových stránek – `HTML,CSS, PHP`.

Jelikož IPFIX kolektor je většinou dostupný pouze z počítače administrátora či správce sítě, nebylo nutné pro jednoduchost řešit žádné přihlašování do aplikace ani ověření uživatele. To je zajištěno právě firewallem kolektoru, který povolí HTTP provoz pouze ze správných počítačů. Bylo tedy možné se plně soustředit na implementaci co nejvíce funkčních věcí. Z toho důvodu byl vybrán i minimalistický, výpočetně a uživatelsky nenáročný vzhled, který dopomáhá administrátorovi k jednoduchému a přehlednému přístupu k získaným informacím. Webový modul jako takový je rozdělen do dvou částí. V první z nich lze nalézt získané informace a statistiky pro provoz BOOTP/DHCPv4, ve druhé pak statistiky pro DHCPv6.

Webový modul v současné době dokáže poskytnout tyto informace:

- BOOTP/DHCPv4
  - `IP adresy všech komunikujících klientů`
  - `IP adresy všech DNS serverů, které byly klientům přiděleny`
  - `IP adresy všech NTP serverů, které byly klientům přiděleny`
  - `Průměrný počet paketů v toku`
  - `Průměrnou délku toku v sekundách`
  - `Přehled všech komunikujících MAC adres`
- DHCPv6
  - `IP adresy všech komunikujících klientů`
  - `IP adresy všech serverů DHCPv6`
  - `IP adresy všech DNS serverů, které byly klientům přiděleny`
  - `IP adresy všech NTP serverů, které byly klientům přiděleny`
  - `Průměrný počet paketů v toku`
  - `Průměrnou délku toku v sekundách`
  - `Přehled všech komunikujících MAC adres`

– `Prehled všech klientských DUID`

V tomto webovém prostředí je potenciál pro budoucí rozšíření mé práce, pokud o něj firma Invea-tech projeví zájem. Implementaci prezentovanou v rámci této bakalářské práce lze považovat za první verzi, která bude jistě v průběhu času upravována. Ukázky webového prostředí jsou dostupné v příloze A.

Kapitola 6

Závěr

V rámci této mé práce byly úspěšně vytvořeny dva moduly do sondy FlowMon společnosti Invea-tech pro možnost monitoringu provozu protokolů BOOTP, DHCP a DHCPv6 v počítačových sítích. Z důvodu efektivity programování se vyplatilo spojit vždy vstupní modul a procesní modul dohromady do jedné sdílené knihovny, což umožnilo rychlejší vývoj díky jednoduššímu přístupu ke sdíleným datům. Naopak pro zvýšení praktického aspektu použití modulů byl vytvořen modul zvlášť pro protokoly BOOTP/DHCP a zvlášť pro DHCPv6.

Díky tomu je možné nabídnout zákazníkovi firmy pouze ten modul, který si objedná, což vytváří nejen lepší obchodní podmínky, ale nenutí to zákazníka používat to, co nepotřebuje nebo nechce. To má důsledek i na celkový výkon monitorovacího systému, protože sonda tak může zpracovávat jen ten protokol, jenž je v síti opravdu použit. Mnou vytvořený nástroj dává do rukou administrátora možnost sledovat kompletní dění na síti, které se týká výše zmíněných protokolů a umožňuje mu síť lépe analyzovat, případně odhalit bezpečnostní díry či neslušné uživatele. Toho je dosaženo nejen výpisem pomocí programu `fbitdump` v řádkové podobě, ale i jednoduchým a přehledným webovým rozhraním.

Celé řešení bylo testováno jak v laboratorních podmínkách v rámci virtualizační platformy VMware Workstation, tak i v reálných počítačových sítích. Tam moduly předvedly, že jsou plně funkční, stabilní a dokáží si poradit i s vysokými datovými toky.

Jako další rozšíření je možno implementovat sledování dalších DHCP a DHCPv6 možností, jejichž zpracovávání přinese administrátorům ještě detailnější informace o tomto typu provozu v rámci jejich počítačové sítě. Dalším možným budoucím zlepšením je i rozvinutí webového rozhraní, které by mohlo poskytovat daleko více statistik a přehledů, než je tomu nyní v čase psaní této práce.

Jelikož práce byla vyvíjena pod vedením lidí z firmy Invea-tech, je velice pravděpodobné, že mnou vytvořené moduly budou začleněny do obchodní nabídky společnosti a po dalším testování a případných úpravách budou nabízeny koncovým zákazníkům, kteří produkt FlowMon od Invea-tech kupují.

Literatura

[1] Alexander, S.; Droms, R.: RFC 1533 – DHCP Options and BOOTP Vendor Extensions [online]. `http://tools.ietf.org/html/rfc1533`, 1993-10 [cit. 2014-05-21].

[2] Croft, B.; Gilmore, J.: RFC 951 – BOOTSTRAP PROTOCOL (BOOTP) [online]. `http://tools.ietf.org/html/rfc951`, 1985-09 [cit. 2014-05-21].

[3] Droms, R.: RFC 1531 – Dynamic Host Configuration Protocol [online]. `http://tools.ietf.org/html/rfc1531`, 1993-10 [cit. 2014-05-21].

[4] Droms, R.: RFC 2131 – Dynamic Host Configuration Protocol [online]. `http://tools.ietf.org/html/rfc2131`, 1997-03 [cit. 2014-05-21].

[5] Droms, R.; Alexander, S.; Silicon Graphics, I.: RFC 2132 – DHCP Options and BOOTP Vendor Extensions [online]. `http://tools.ietf.org/html/rfc2132`, 1997-03 [cit. 2014-05-21].

[6] Kolektiv autorů: RFC 3315 – Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [online]. `http://tools.ietf.org/html/rfc3315`, 2003-07 [cit. 2014-05-21].

[7] Kolektiv autorů: RFC 4861 – Neighbor Discovery for IP version 6 (IPv6) [online]. `http://tools.ietf.org/html/rfc4861`, 2007-09 [cit. 2014-05-21].

[8] Kolektiv autorů: RFC 5175 – IPv6 Router Advertisement Flags Option [online]. `http://tools.ietf.org/html/rfc5175`, 2008-03 [cit. 2014-05-21].

[9] Kolektiv autorů: RFC 5908 – Network Time Protocol (NTP) Server Option for DHCPv6 [online]. `http://tools.ietf.org/html/rfc5908`, 2010-06 [cit. 2014-05-21].

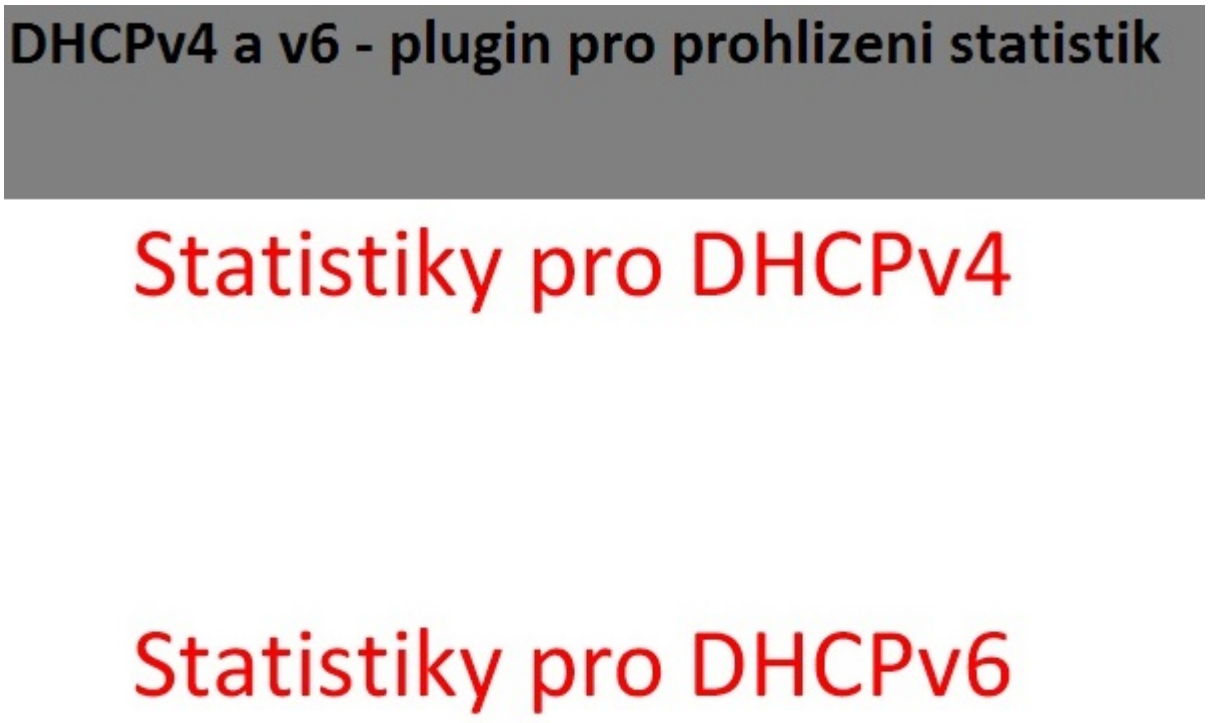
[10] Satrapa, P.: *IPv6 (3. vydání)*. CZ.NIC, 2011, iSBN 978-80-904248-4-5.

[11] Wimer, W.: RFC 1532 – Clarifications and Extensions for the Bootstrap Protocol [online]. `http://tools.ietf.org/html/rfc1532`, 1993-10 [cit. 2014-05-21].

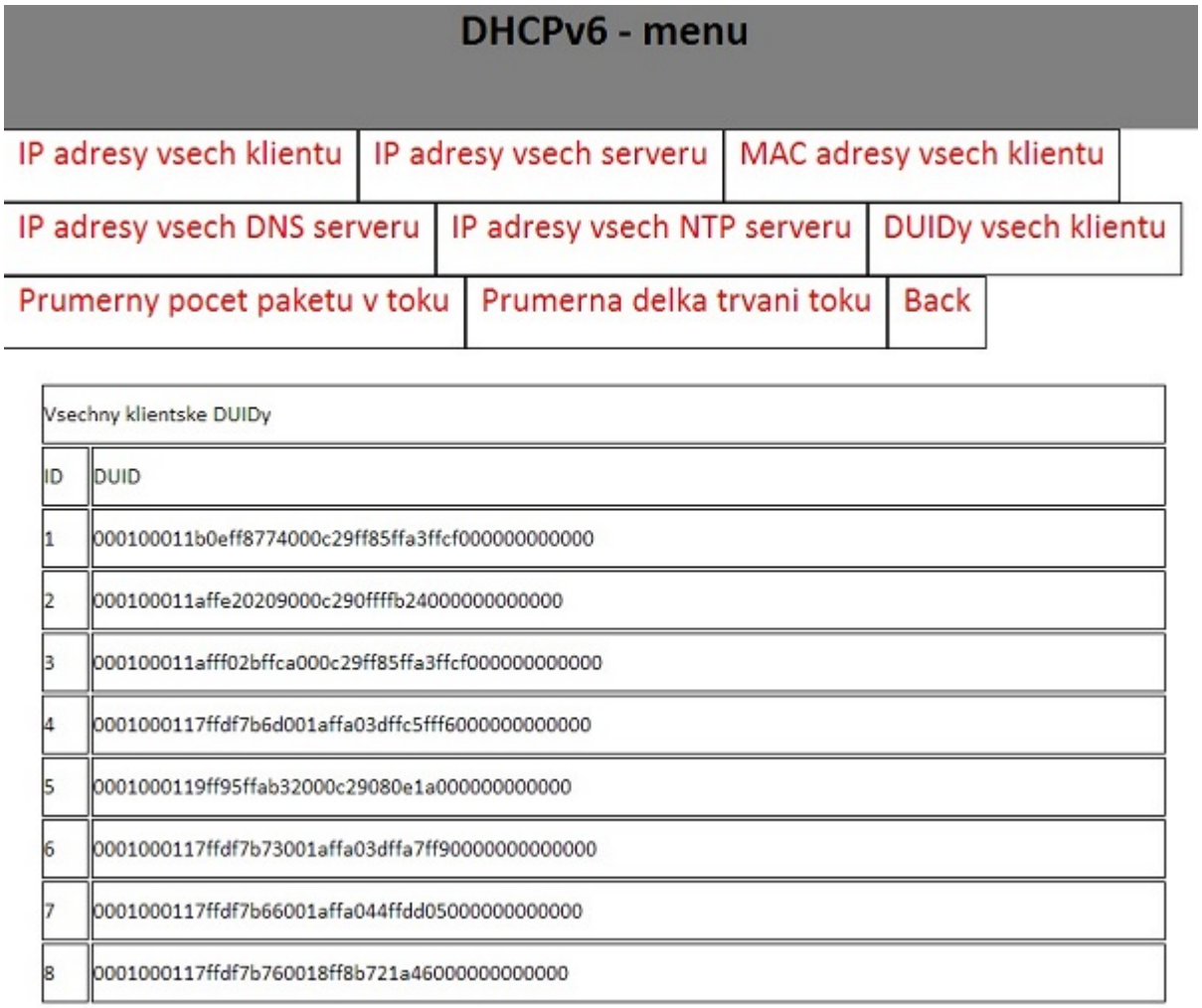
[12] Wimer, W.: RFC 1542 – Clarifications and Extensions for the Bootstrap Protocol [online]. `http://tools.ietf.org/html/rfc1542`, 1993-10 [cit. 2014-05-21].

Dodatek A

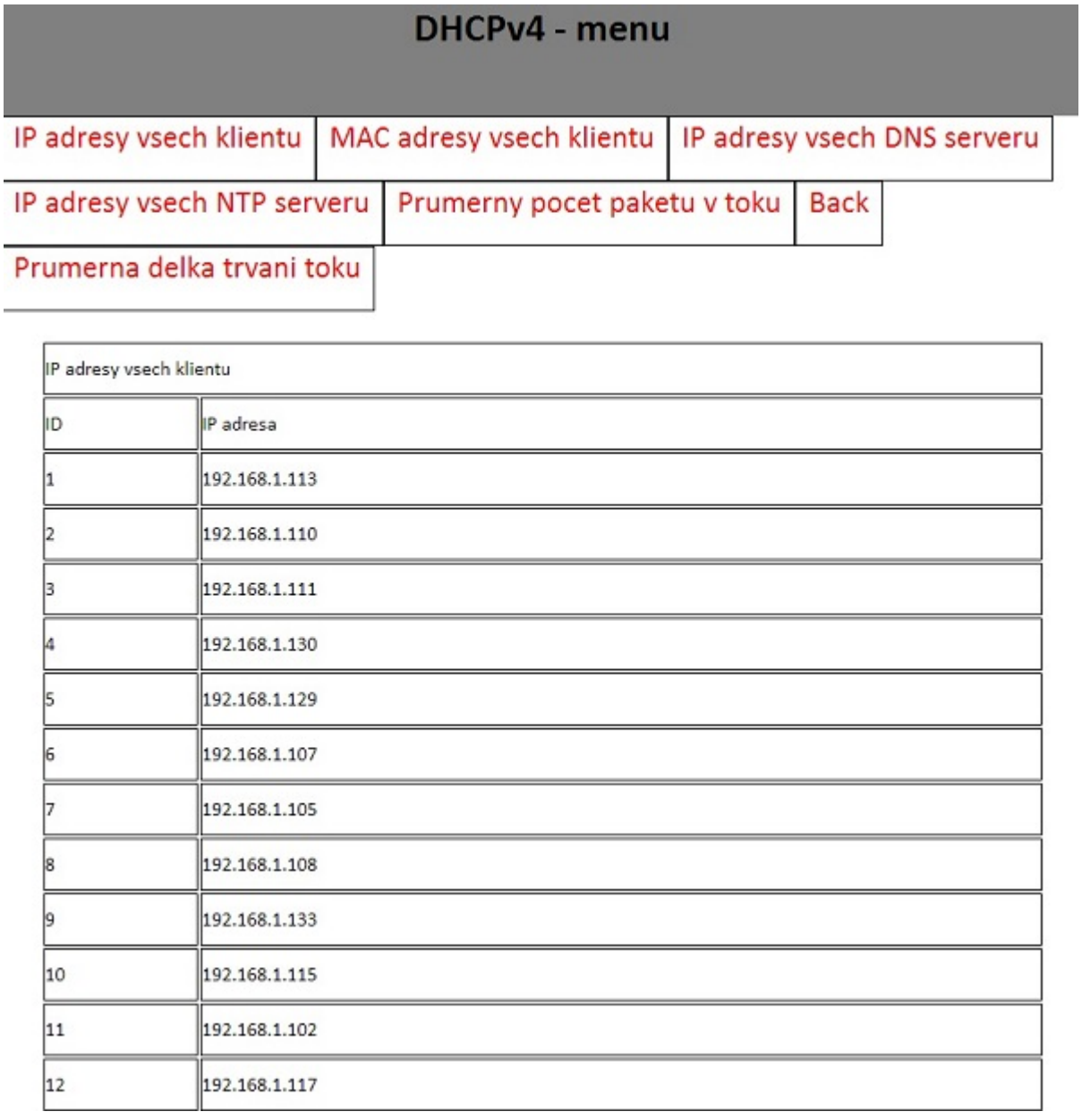
Ukázky webového prostředí



Obrázek A.1: Menu volby statistik



Obrázek A.2: Výpis všech zjištěných klientských DUID v DHCPv6



Obrázek A.3: Výpis všech zjištěných IPv4 adres všech komunikujících klientů

Dodatek B

Obsah CD

Elektronická verze této bakalářské práce se nachází přímo v kořenovém adresáři přiloženého CD. Na disku se nacházejí také tyto složky:

- Adresář `src`
  - Zdrojové kódy vyvinutých modulů včetně testovacích souborů a Makefile
- Adresář `tex`
  - Zdrojové kódy této bakalářské práce ve formátu systému LaTeX