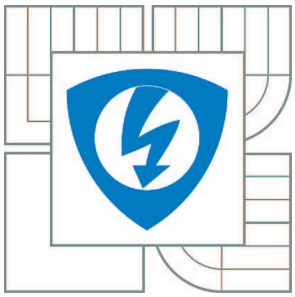
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

INSTALACE, KONFIGURACE A TESTOVÁNÍ SERVEROVÝCH SLUŽEB TYPU DHCP, FTP, VPN, NAT A SNMP V PROSTŘEDÍ IPV4 A IPV6

INSTALLATION, CONFIGURATION AND TESTING OF SERVER SERVICES DHCP, FTP, VPN, NAT AND SNMP IN IPV4 AND IPV6 ENVIRONMENTS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

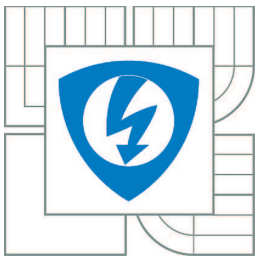
AUTOR PRÁCE — LIBOR BRÁZDA

AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. VÍT NOVOTNÝ, Ph.D.

SUPERVISOR

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Libor Brázda	***ID:***	125376
Ročník:	3	***Akademický rok:***	2011/2012

NÁZEV TÉMATU:

Instalace, konfigurace a testování serverových služeb typu DHCP, FTP, VPN, NAT a SNMP v prostředí IPv4 a IPv6

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s typickými serverovými službami v datových sítích IP a jejich protokolovým vybavením. Zaměřte se na služby typu DHCP, FTP a VPN, NAT a SNMP, posuďte odlišnosti implementace v prostředí IPv4 a IPv6 a nainstalujte a zprovozněte tyto v prostředích Windows Server a vybrané linuxové distribuci. Na základě zkušeností navrhněte laboratorní úlohy ve virtuálních strojích obou operačních systémů.

DOPORUČENÁ LITERATURA:

[1] C. SCHRODER: Linux - Kuchařka administrátora sítě. Computer press, ISBN 9788025124079, ČR, 2009

[2] S. CRAWFORD, CH. RUSSEL: Windows Server 2008 – Velký průvodce administrátora. Computer press, ISBN 9788025121153, 2009

Termín zadání:	6.2.2012	***Termín odevzdání:***	31.5.2012

Vedoucí práce: doc. Ing. Vít Novotný, Ph.D.
Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Tato bakalářská práce je zaměřena na vybrané serverové služby typu FTP, DHCP, VPN, NAT a SNMP a jejich realizaci v operačním systému Windows Server 2008 R2 a linuxové distribuci Debian. V teoretické části práce je rozebrána problematika protokolu TCP/IP včetně adresování v prostředí IPv4 a IPv6. V dalších kapitolách jsou podrobně popsány jednotlivé služby, jejich princip, zabezpečení atd. Poslední část je zaměřena na popis laboratorních úloh, jejich náplň a přínos pro studenta a také software použitý při realizaci úloh.

Výsledkem práce jsou vytvořené laboratorní úlohy na konfiguraci jednotlivých služeb v obou operačních systémech. Úlohy jsou tvořené stylem, aby i student, méně znalý této problematiky byl schopen úspěšně úlohy nakonfigurovat a otestovat.

KLÍČOVÁ SLOVA

DHCP, FTP, VPN, NAT, SNMP, IPv4, IPv6, server, Linux, Windows Server, PPTP, IPSec, IP maškaráda.

ABSTRACT

This bachelor thesis focuses on selecting server services such as FTP, DHCP, VPN, NAT and SNMP and their implementation in operating system Windows Server 2008 R2 and linux distribution Debian. The theoretical part analyzes the problems of TCP/IP protocol including addressing in IPv4 and IPv6 environments. In next chapters, thesis contains detailed descriptions of individual services, their principles, security, etc. The last part is focused on the description of the laboratory task, their activities and contribution to student and also software used in the implementation tasks.

The result of laboratory work is created job on the configuration of individual services in both operating systems. Tasks are made for students without previous knowledge of the problem, so they can successfully configure and test these tasks.

KEYWORDS

DHCP, FTP, VPN, NAT, SNMP, IPv4, IPv6, server, Linux, Windows Server, PPTP, IPSec, IP masquerading.

BRÁZDA, Libor *Instalace, konfigurace a testování serverových služeb typu DHCP, FTP, VPN, NAT a SNMP v prostředí IPv4 a IPv6*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. 91 s. Vedoucí práce byl doc. Ing. Vít Novotný, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Instalace, konfigurace a testování serverových služeb typu DHCP, FTP, VPN, NAT a SNMP v prostředí IPv4 a IPv6“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Téměř každý, internetem znalý uživatel ví, že existují zařízení zvané servery. Někteří dokonce vědí, že na nich běží aplikace, které poskytují klientům různé funkce. Mezi nejznámější serverové služby patří např. webový server, který poskytuje WWW stránky, nebo souborový server, který slouží jako centrální úložiště např. ve firmě. Nicméně na serveru běží také spousta jiných služeb, které klienti využívají, aniž by o nich vůbec věděli. Mezi tyto „neviditelné“ služby můžeme zařadit např. službu DHCP, která zajišťuje automatickou konfiguraci síťových parametrů, nutných pro funkčnost zařízení v síti. Dříve byly servery specializované pro běh úzkého okruhu služeb, které nabízely. Hlavním důvodem byly technické parametry počítačů. Současný hardware je však dostatečně výkonný, aby umožňoval mnohem větší funkcionalitu v jediném serveru.

Cílem bakalářské práce je seznámení se s vybranými serverovými službami, jejich instalací a konfigurací v operačním systému Windows Server 2008 R2 a linuxové distribuci Debian. Na základě získaných zkušeností navrhnout laboratorní úlohy v obou operačních systémech pro předmět Architektura sítí (BARS).

Bakalářská práce je rozdělena na dvě hlavní části, první část je čistě teoretická a popisuje serverové operační systémy, protokol TCP/IP včetně adresování v prostředí IPv4 a IPv6 a jednotlivé testované služby. Popis služeb je poměrně rozsáhlý a detailní, jelikož poté tvoří základ pro teoretický úvod jednotlivých laboratorních úloh. Druhá část je zaměřena na popis a obsah laboratorních úloh, jejichž pracovní postup je do bakalářské práce přiložen jako příloha.

1 SERVEROVÉ OPERAČNÍ SYSTÉMY

V dnešní době už téměř každá síť obsahuje prvek, do kterého jsou soustředěny informace a síťové služby, takovému prvku se říká server. Dříve bylo použití serverů výsadou jen větších nebo speciálně zaměřených sítí, ale dnes se využívají servery i v menších sítích a dokonce v některých domácnostech. Pod pojmem server si představujeme počítač připojený k síti, který poskytuje nebo přijímá informace od jiných počítačů v síti. Obvykle na něm běží speciálně upravený operační systém, který je optimalizován právě pro sdílení prostředků v počítačové síti. Server může plnit různorodé funkce, např. může fungovat jako souborový server, webový server, databázový server, k autentizaci uživatelů atd.

Architektuře, ve které jeden prvek posílá žádosti druhému prvku a ten na žádosti odpovídá se říká klient/server. Většina operací se prování na straně serveru, klientu přijdou již zpracovaná data, která si převede do vhodné formy. Nejznámějším příkladem klienta v tomto modelu, může být např. webový prohlížeč, který odesílá dotazy na webový server.

V současnosti patří mezi nejznámější a nejpoužívanější serverové operační systémy především nejrůznější typy Unixových systémů, serverové produkty od softwarového giganta Microsoft, nebo síťový operační systém Novell Netware.

1.1 Operační systémy Windows Server

Jeden z nejpoužívanějších operačních systémů pro servery je vyvíjen americkou společností Microsoft. Aktuální verzí operačního systému je Windows Server 2008 R2, který běží na podobném jádru jako klientský operační systém Windows 7, sdílí spolu tedy své funkce a architekturu a jejich společné nasazení přináší některé výhody.

Stejně jako klientské systémy, i operační systémy jsou rozděleny na edice[1]:

- **Standard Edition** – je přímou náhradou systému Windows Server 2003, je určen ke sdílení služeb a zdrojů jiným systémům v síti. Má bohaté možnosti nastavení a funkcí.
- **Enterprise Edition** – rozšiřuje poskytované služby oproti Standard edici, podporuje služby jako Cluster Service[2], Active Directory FS. Podporuje vyměnitelnou RAM za běhu systému.
- **Datacenter Edition** – má vylepšené clusteringové funkce a podporuje konfigurace s velkým množstvím operační paměti, až 2 TB při použití 64bitové architektury. Pro svůj běh vyžaduje minimálně 8 CPU, a maximálně jich podporuje až 64.

[1] Bližší informace např. na [1] [2].

[2] Servery mají uložena všechna potřebná data na jednom datovém úložišti.

- **Web Server** – edice určená pro poskytování webových služeb, tato edice podporuje pouze související funkce jako např. IIS, ASP.NET atd. Neobsahuje například Active Directory.

Mezi novinky, které Windows Server 2008 R2 nabízí, patří například technologie Direct Access, která slouží k připojení koncové stanice do firemní sítě bez nutnosti konfigurovat VPN spojení a bez interakce s uživatelem. Podporuje pokročilejší serverovou virtualizaci Hyper-V, technologii RemoteFX, která dovoluje využívat 3D grafiku a video ve vysokém rozlišení v prostředí vzdálené plochy. IIS nabízí podporu SSL a IPv6 pro FTP server a mnoho dalšího [3].

Microsoft dokonce nabízí serverový operační systém do domácností a SOHO podnikatelský segment, který nese název Windows Home Server (aktuálně ve verzi 2011). Je založený na operačních systémech Windows Server. Je zaměřený na pokročilé domácí uživatele a nabízí např. centralizované úložiště dat, vzdálený přístup k datům přes internet (FTP, HTTP), automatické zálohování počítačů v síti. I jeho cena je relativně přijatelná, pohybuje se kolem 1500 Kč (k 14.11.2011).

1.2 Operační systémy na bázi UNIX

Unixové systémy jsou hojně využívány jako operační systémy pro server, pracovní stanice a dnes i pro osobní počítače. První Unixový operační systém byl vyvinut už v roce 1969. Pod pojmem Unix, je také chápáno označení velké skupiny příbuzných operačních systému, které vycházejí z přesně definovaných vlastností. Systémy založené na Unixu jsou charakteristické tím, že:

- je jednoduchý
- je víceúlohový (podporuje multitasking)
- je víceuživatelský (na počítači s Unixem může pracovat více lidí současně)
- má hierarchický souborový systém

a mnoho dalších charakteristických vlastností. Mezi nejznámější operační systémy Unixového typu se dnes řadí především systémy BSD a systémy založené na jádře Linux.

1.2.1 Linux

Linux je víceuživatelský operační systém vycházející ze standardu Unix. Autorem prvního linuxového jádra s názvem Minix z roku 1991 je Linus Torvald, který byl v té době studentem informatiky v Helsinkách. Systémy založené na jádře Linux, se souhrnně označují jako GNU/Linux (název označuje spojení jádra Linux se softwarem vyvíjeného v rámci projektu GNU z dílny nadace Free Software Foundation).

Dnes se používá jen zkrácený název Linux, který je chápán jako celý operační systém (jádro + základní software).

Linus Tovarlds se rozhodl podělit se s odbornou veřejností o své linuxové jádro. Projekty vytvářené na základe GNU/Linux se dnes nazývají jako distribuce. Distribuce je operační systém GNU/Linux obohacený o administrátorské programy, které umožňují snazší instalaci, správu software a nastavení hardware. V dnešní době je několik stovek nekomerčních i komerčních distribucí, které stále prochází vývojem [4]. Mezi nejznámější patří např.:

- Debian
- Fedora
- Ubuntu
- Gentoo
- SUSE
- RedHat

V této práci budeme pracovat s distribucí Debian, která je jednou z nejstarších distribucí. Na jejím vývoji pracuje velké množství dobrovolníků z celého světa. Používá vlastní balíčkovací systém nazvaný jako APT (*Advanced Packaging Tool*). Softwarové repozitáře jsou jedny z nejrozsáhlejších a jsou rozděleny do tří větví: stabilní, testovací a nestabilní software. Jde o velmi oblíbenou distribuci jak na servery tak i na osobní počítače. Dokonce z něj vychází asi nejznámější a nejpoužívanější distribuce mezi běžnými uživateli s názvem Ubuntu [5].

2 TCP/IP PROTOKOL

TCP/IP (*Transmission Control Protocol/Internet Protocol*) je rodina protokolů, navržena pro komunikaci v počítačových sítí. Je to konkurenční protokol k referenčnímu ISO/OSI[1] modelu. Jeho počátek je datován v 60. letech minulého století a je spojen s činností agentury ARPA ministerstva obrany Spojených států amerických, která si tento protokol nechala vyvinout pro svou počítačovou síť ARPANET[2]. Protokol se nejdříve používal pouze v akademickém prostředí, ale později se začal implementovat do BSD Unixu. Díky své oblíbenosti se začal dostávat i na ostatní platformy a je implementován snad ve všech zařízeních, které jsou určené pro komunikaci v síti.

2.1 Architektura TCP/IP

Jak již bylo zmíněno, TCP/IP je konkurent k ISO/OSI modelu, ale jeho filozofie je poněkud odlišná. Nesnaží se zajistit spolehlivý charakter služeb již na úrovni síťové vrstvy, ale až na úrovni transportní vrstvy. Tím nemusí komunikační podsíť ztrácet svojí přenosovou kapacitu pro zajištění spolehlivého přenosu (potvrzování, opětovné přeposílání paketu) a také není tak složitá. TCP/IP předpokládá nespojovaný charakter přenosu v síti. Protokol se dělí na čtyři vrstvy, vrstva síťového rozhraní, síťová vrstva, transportní a aplikační viz obr. 2.1 [6].

2.1.1 Vrstva síťového rozhraní

Je to nejnižší vrstva architektury TCP/IP, slučuje v sobě funkci fyzické a linkové vrstvy. Zabývá se elektrickými, elektromagnetickými či optickými signály používané při komunikaci mezi bezprostředními sousedy. Popisuje kabely a konektory, délku vlny a způsob kódování bitů. Také má na starosti vše, co je spojeno s ovládáním konkrétní přenosové cesty a s přímým vysíláním a příjmem datových rámců. Vzhledem k častému připojování uzlů na lokální sítě typu Ethernet, je vrstva síťového rozhraní označována také jako Ethernetová vrstva.

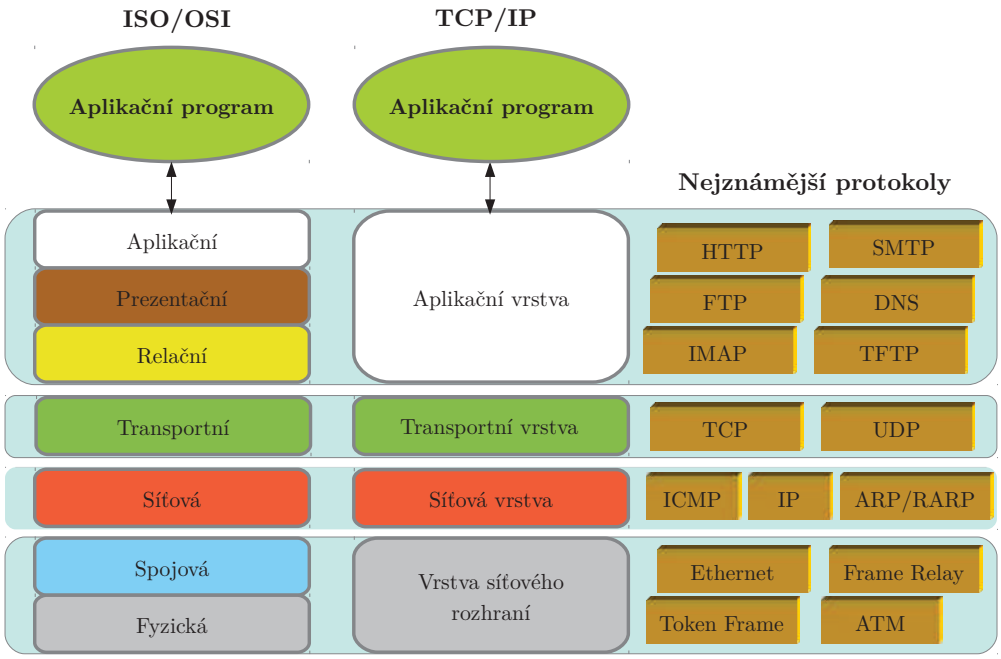
2.1.2 Síťová vrstva

Síťová vrstva zabezpečuje přenos dat mezi vzdálenými počítači v rozsáhlých sítí – WAN[3]. Základní jednotkou je IP datagram, který je zapouzdřen do linkového rámce.

[1] Sedmivrstvý referenční model pro komunikaci v počítačových sítí.

[2] Počítačová síť spuštěná v roce 1969, která byla zárodkem toho, čemu dnes říkáme Internet.

[3] *Wide Area Network* – počítačová síť, která pokrývá rozlehlé geografické území, nejznámějším příkladem je síť Internet.



Obr. 2.1: Vrstvový model ISO/OSI a TCP/IP s některými protokoly

Hlavním úkolem je, aby se jednotlivé pakety dostaly od odesílatele až ke svému skutečnému příjemci, přes případné směrovače (routery). Na této úrovni je přenos realizován jednoduchou nespolehlivou datagramovou službou. V Internetu obvykle leží mezi komunikujícími počítači jeden nebo více směrovačů. Směrovač vybalí paket z datagramového rámce a před odesláním do jiné linky ho opět zapouzdří do jiného linkového protokolu. Síťová vrstva nevidí zařízení pracující na nižších vrstvách (opakovače, přepínače, modemy atd.).

Síťová vrstva zajišťuje směrování mezi jednotlivými sítěmi a vyšším vrstvám tak vytváří iluzi jednotné homogenní sítě. Aby mohla provádět směrování (routování) mezi sítěmi, musí zavést jednotný způsob adresování. Těmto adresám se říká tzv. IP adresy, proto se někdy i síťové vrstvě říká IP nebo Internet vrstva. V dnešní době se používají dvě verze adresování na síťové vrstvě a to IPv4 a IPv6, které budou podrobněji popsány v kapitole 2.2 a 2.3. Dále na této vrstvě probíhá segmentace a skládání datagramů do/z rámců specifikovaných vrstvou síťového rozhraní. Další protokoly, které mohou běžet na této vrstvě jsou např. ARP, RARP, ICMP, IPSEC [6] [7].

2.1.3 Transportní vrstva

Třetí vrstva TCP/IP modelu je vrstva transportní. Hlavním úkolem je zajistit spojení dvou koncových bodů. Vrstva se spoléhá na funkce nižších vrstev, nemá žádné informace o topologii sítě, takže při přenosu jí nezajímají síťové prvky jako směrovače nebo přepínače, celá síť se jí jeví jako jedna homogenní síť. Je jí tedy jedno, jestli jsou koncové body v lokální síti, nebo v internetu. Transportní vrstva nabízí dvě varianty spojení:

- **TCP** (*Transmission Control Protocol*) – protokol poskytuje transportní službu se spojením – vytvoří se virtuální okruhy. Jednotlivé bajty jsou číslovány a poškozená nebo ztracená data jsou znovu vyžádána. Integrita dat při přenosu je zajištěna kontrolními součty. Poskytuje tedy spolehlivý charakter přenosu dat. Protokol je vhodné použít tam, kde vyžadujeme aby odesílaná zpráva měla stejnou informační hodnotu u odesílatele i příjemce. Používá se např. u FTP, SMTP, HTTP.
- **UDP** (*User Datagram Protocol*) – poskytuje jednoduchou transportní službu bez spojení. Protokol nezaručuje, že se přenášený datagram neztratí nebo nezmění pořadí doručených datagramů. Zodpovědnost nad spolehlivostí přenosu dat mají aplikační protokoly. Má nižší režii sítě než TCP, protože zde není žádné řazení nebo sledování spojení. Je vhodný pro videokonference nebo přenos hlasu po síti [6].

2.1.4 Aplikační vrstva

Účelem čtvrté vrstvy je poskytnout aplikačním procesům přístup ke komunikačnímu systému a tím umožnit jejich vzájemnou spolupráci. Aplikační vrstva požádá nižší vrstvu o vytvoření spojení, do kterého pak vkládá aplikační data. Předepisuje dialog a formát dat předávaných mezi aplikacemi. V této vrstvě mohou funkce v aplikační vrstvě provádět nejen programy a technické prostředky, ale i lidé.

Pro rozlišení aplikačních protokolů se používají tzv. porty, což jsou domluvená číselná označení aplikací. Každé síťové spojení aplikace je jednoznačně určeno adresou, číslem portu (rozsah 0–65535) a transportním protokolem. Porty jsou rozčleněny do třech skupin: dobře známé porty, registrované a dynamické/privátní. Dobře známé porty (*well known ports*), jsou porty v rozsahu 0 až 1023 a jsou vyhrazené pro nejběžnější služby např. FTP (port 21 a 20), SMTP (port 25), DNS (port 53) a HTTP (port 80). Za přidělování portů jednotlivým službám se v dnešní době stará nezisková organizace ICANN[4].

[4] *Internet Corporation for Assigned Names and Numbers* – organizace dohlížející na přidělování IP adres, správu kořenových zón DNS, přidělování portů a další náležitosti internetových protokolů.

2.2 Protokol IPv4

IPv4 je čtvrtá revize IP protokolu a zároveň první, která se začala hojně využívat. Společně se zatím méně používaným IPv6 vytvářejí základ pro komunikaci v rámci sítě Internet. IPv4 je specifikován v RFC 791. IP protokol dokáže dopravit data mezi dvěma libovolnými počítači v Internetu. Při cestě prochází přes mnohé LAN sítě, aby byly data od odesílatele doručeny k příjemci, musí protokol vybrat nejvhodnější cestu ke koncovému zařízení. Tomuto procesu se říká směrování a zařízení, která toto vykonávají jsou směrovače (routery). Směrovače vzájemně propojují jednotlivé sítě do rozsáhlé sítě, ke svojí činnosti potřebují jednotný styl adresování, bez ohledu na skutečné MAC[5] adresy. Protokol IP poskytuje službu bez spojení, tedy nenavazuje ani neudržuje spojení a ani neudržuje informace o poslaných datagramech. Spolehlivost doručení musí zajistit protokoly vyšších vrstev (např. již zmíněný TCP).

Každé zařízení, které chce odeslat nějakou informaci jinému zařízení, musí znát adresu svého příjemce. Adresy odesílatele (zdrojová) i příjemce (cílová) jsou uloženy v hlavičce IP datagramu. Každý směrovač, ke kterému přijde datagram si přečte informace o cílové adrese a podle použitého typu směrování vybere nejvhodnější cestu pro doručení datagramu. IP datagram nese i další důležité informace k doručení dat viz obr. 2.2.

2.2.1 Adresace v IPv4

Protokol IP verze 4 používá k adresování IP adresu o délce 32 bitů. IP adresa jednoznačně určuje síťové rozhraní systému. Takovéto jednoznačné adrese se říká *unicast*. V IPv4 máme ještě následující adresy:

- **broadcast** (všeobecný oběžník) – je určený všem stanicím v lokální síti. Broadcast obecně nedokáží směrovače předávat do dalších, muže se tedy šířit jen v rámci LAN. Používá se například při identifikaci zařízení v síti, také může způsobit zahlcení sítě, pokud není vhodně rozdělena nebo chráněna.
- **multicast** (skupinový oběžník) – je to metoda přeposílání dat z jednoho zdroje vybrané skupině koncových stanic. Od zdroje jdou datagramy jen jedním datovým tokem, až síťový prvek (směrovač, přepínač) replikuje data pro zájemce.
- **loopback** (virtuální programová smyčka) – obvykle běží na adrese 127.0.0.1 (ale může používat libovolnou adresu z rozsahu 127.0.0.0–127.255.255.255), slouží jako virtuální síťové rozhraní. Nejčastěji slouží k testování funkčnosti vlastního síťového rozhraní.

[5]Každé zařízení navržené pro komunikaci v síti, má svojí „neměnnou“ 48bitovou adresu přidělenou již z výroby.

Adresa IPv4 je tvořena čtyřmi bajty. Zapisuje se notací, kde se jednotlivé bajty od sebe oddělují tečkou. Zápis může být v binární, dekadické nebo hexadecimální soustavě. Nejčastější a nejjasnější zápis je v dekadickém tvaru, binární je naopak vhodný při počítání adres. IP adresa se skládá ze dvou částí:

1. **Adresa sítě** (*Network ID*) – první část adresy, která slouží k identifikaci podsítě a směrování. Adresa se dá vypočítat z masky sítě, v binárním tvaru obsahuje jedničky tam, kde se v adrese nachází síť a podsíť, a nuly tam, kde je uzel.
2. **Adresa uzlu v síti** (*Host ID*) – adresa zařízení v rámci konkrétní sítě.

Vzhledem k různým požadavkům na počet podsítí či hostů, se adresný rozsah rozdělil do pěti tříd viz Tab. 2.1.

Tab. 2.1: Třídy IP adres

Třída	**První oktet**	**První oktet (bin)**	**Síťová (N) a host (H) část**	**Maska sítě**
A	1–127	**0**0000000–**0**1111111	N.H.H.H	255.0.0.0
B	128–191	**10**000000–**10**111111	N.N.H.H	255.255.0.0
C	192–223	**110**00000–**110**11111	N.N.N.H	255.255.255.0
D	224–239	**1110**0000–**1110**1111	multicast	—
E	240–255	**1111**0000–**1111**1111	experimenty	—

- **Třída A** – má pro adresu sítě vyhrazeno 7 bitů a pro host část 24 bitů. Může tedy adresovat 128 (2^7) sítí a zhruba 16 miliónů ($2^{24} - 2$) koncových zařízení. Je tedy určena pro rozlehlé sítě.
- **Třída B** – používá první dva oktety adresy pro identifikaci sítě, další dva pro identifikaci rozhraní. Je zde asi 16 tisíc (2^{14}) možných sítí a asi 65 tisíc koncových zařízení ($2^{14} - 2$).
- **Třída C** – tato třída počítá s velkým počtem sítí, a jen s 254 (2^8-2) koncovými zařízení v každé síti. Časté použití v malý sítích, domácnostech.
- **Třída D** – tento rozsah adres slouží pro skupinovou adresaci (*multicast*).
- **Třída E** – třída s nejmenším rozsahem adres, slouží jen pro experimentální účely.

Z těchto tříd, byly ještě z důvodu nedostatku veřejných adres vyčleněny adresy privátních sítí Tab. 2.2.

Tab. 2.2: IP adresy privátních sítí

Třída A	10.0.0.0/8	10.0.0.0 až 10.255.255.255
Třída B	172.16.0.0/12	172.16.0.0 až 172.31.255.255
Třída C	192.168.0.0/16	192.168.0.0 až 192.168.255.255

Tyto sítě se nemohou nikdy objevit v Internetu. Jsou to adresy určené pro intranet[6], tedy za použití technologie NAT[7], nám pro celou privátní síť stačí jediná veřejná IP adresa pro komunikaci v Internetu. Nejen že to zásadně oddálilo vyčerpání adres IPv4 („pouze“ 2^{32} všech možných adres), ale také to zvyšuje bezpečnost privátní sítě [8] [9].

I přes veškeré snahy ušetření veřejných adres, jsme se dočkali doby, kdy už je vyprázdněn globální registr adres IANA. Vše tedy míří k rychlejšímu zavádění protokolu IPv6, „konečnému“ řešení problému s nedostatkem veřejných IP adres.

2.3 Protokol IPv6

Už v 90-tých letech se začalo přemýšlet o problémech protokolu IPv4, který souvisel s nedostatkem adresného prostoru a jeho technickými nedostatky v průběhu vývoje IT světa. Úsilí směřovalo do vzniku nového protokolu, IP protokolu nové generace (*IP Next Generation*), který se začal označovat jako IPv6. IP protokol verze 6 byl poprvé specifikován už v roce 1995, a i když už je několik let implementován snad na všech běžných systémech, stále není příliš rozšířený.

Hlavním požadavkem na nový protokol, bylo zvětšení rozsahu adresného prostoru, ta je stanovena na 128 bitů, tedy čtyřnásobek délky používané v IPv4. Je tedy k dispozici 2^{128} adres, což je asi $3{,}4 \cdot 10^{38}$. Takto obrovské číslo nám asi moc neřekne, ale když si představíme, že počet adres v IPv4 je roven objemu látky, která se vejde na kávovou lžičku, tak adresný prostor v IPv6 představuje objem Měsíce. Nebo si můžeme představit, že na jeden milimetr čtvereční zemského povrchu připadá $667 \cdot 10^{15}$ adres.

Změnou prošel i formát datagramu, počet jeho položek byl minimalizován a rozložení upraveno pro konstantní délku hlavičky datagramu. Volitelné položky byly přesunuty do samostatných hlaviček, které se mohou připojovat k pevnému základu. Více v kapitole 2.3.1.

[6] Počítačová síť, která používá stejné technologie jako Internet, ale je privátní, je omezena na malou skupinu lidí, např. školy, firmy, domácnosti.

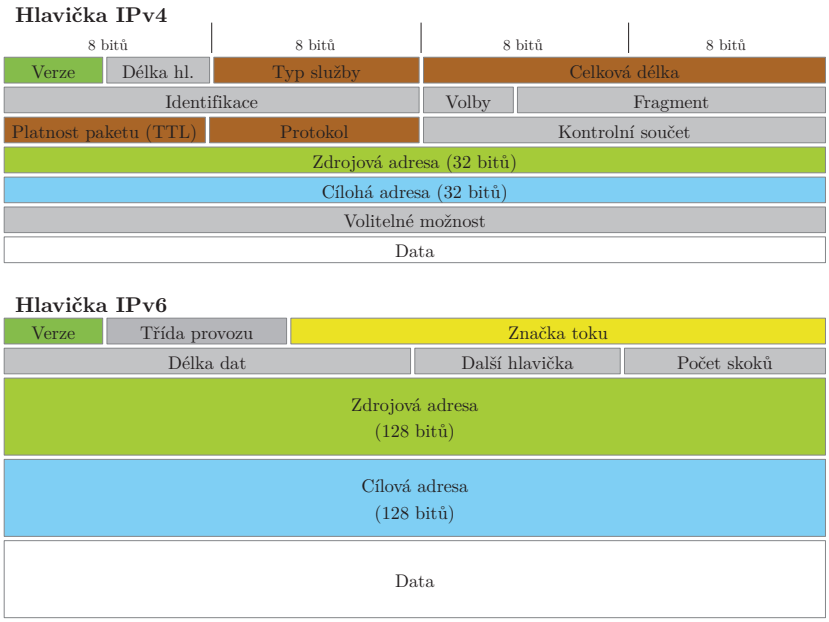
[7] *Network Address Translation* – je způsob úpravy síťového provozu skrz bránu přepisem výchozí nebo cílové IP adresy a portu.

Další změnu prodělala automatická konfigurace, aby byla pokud možno bezpracná. IPv6 zavádí dvě možnosti, první je stavová konfigurace, což je vlastně DHCP upravené pro potřeby nového protokolu. Druhou možností je bezstavová konfigurace, kdy si počítač dokáže sám stanovit adresu a naučí se směrovat bez jakéhokoli zásahu správce. Počítač na základě objevování směrovačů, parametrů sítě a testování jednoznačnosti adresy dokáže přidělit volnou adresu z konkrétní sítě.

Svým vývojem prošlo samozřejmě i zvýšení bezpečnosti. IP datagram obsahuje rozšiřující autentizační a šifrovací hlavičky. Autentizační ověřuje pravost odesílatele dat a zda nedošlo během přenosu k pozměnění dat. Šifrovací hlavička dokáže to samé, jen ještě přidává možnost zašifrovat celá přenášený obsah datagramu [10].

2.3.1 Datagram IP

Formát datagramu se stále skládá z hlavičky za kterou následují přenášená data. IP datagram verze 6 se skládá ze čtyřiceti bajtů základního záhlaví následovaného rozšířeními. Oproti IPv4 hlavičky má nová verze konstantní délku základní hlavičky, nemusí být tedy v každé hlavičce kontrolní součet, který každý směrovač musel přepočítat. Prvky které byly nepovinné jsou přesunuty do rozšiřujících hlaviček, které se přidávají podle potřeby. Díky tomu se hlavička u IPv6 zvětšila jen na dvojnásobek i při skutečnosti, že adresný prostor vzrostl čtyřikrát. Struktura záhlaví obou verzí protokolu je na obr. 2.2 [10].



Obr. 2.2: Porovnání datagramů IPv4 a IPv6

Verze – začátek datagramu, která nese číslo verze protokolu. Pro protokol IPv6 nese hodnotu 6.

Třída dat – nabývá hodnost 0–15 a specifikuje přenášená data pro případ rozhodování při zahlcení sítě. V okamžiku zahlcení sítě zahazují směrovače datagramy, pole tedy specifikuje, které IP datagramy je možné zahodit dříve než jiné. Interval 0–7 je určený pro klasický provoz např. provoz na pozadí, uživatelem prováděné přenosy dat nebo řízení sítě. Rozsah 8–15 je určen pro přenosy v reálném čase. Datagramy s nižší hodnotou se zahazují dříve než datagramy s vyšší hodnotou.

Značka toku – umožňuje označit datový tok pro rychlejší zpracování směrovači. Funguje tak, že datagramy se stejnou značkou nemusí procházet každý zvlášť směrovací tabulkou, vždy jí projde jen první datagram, a ostatní ho „následují“ bez nutnosti procházet znovu tabulku. Tok je určen adresou odesílatele, adresou příjemce, polem identifikace toku dat a třídou dat.

Délka dat – specifikuje délku IP datagramu bez základního záhlaví. Pole je dlouhé 2 bajty, největší délka přenášeného datagramu je tedy 65 635 bajtů.

Další hlavička – nese informace o následující hlavičce, případně o protokolu vyšší vrstvy. Např. hodnota 6 specifikuje TCP protokol, 17 UDP protokol.

Počet skoků – odpovídá položce TTL[8] v IP verze 4. Určuje kolik skoků (přes kolik projde směrovačů) může datagram absolvovat, než bude zahozen. Při průchodu každým směrovačem se jeho hodnota sníží o 1 a při vypršení počtu skoků je datagram zahozen a odesílatel dostane ICMP zprávu o vypršení maximálního počtu přeskoků.

Zdrojová a cílová adresa – největším přínosem IPv6 je právě obrovský adresný prostor. Z celkové velikosti hlavičky tvoří adresy až 80 %. Podrobnější problematice adresace se bude věnovat kapitola 2.3.2.

2.3.2 Adresace v IPv6

Stejně jako u předchůdce jsou adresy přidělovány jednotlivým rozhraním zařízení. Jak již bylo zmíněno, IPv6 nabízí obrovský adresný prostor, ten byl navržený pro dělení do hierarchické směrovací domény, které odrážejí současný moderní Internet. Díky 128-bitové délce dovoluje navrhnout mnoho úrovní v hierarchii a flexibilitě při tvorbě topologie sítí, toto v dnešním Internete založeném na protokolu IPv4 chybí. Stejně jako v IPv4, i zde rozeznáváme několik druhů adres:

- **unicast** – individuální jednoznačná identifikace síťového rozhraní.
- **anycast** – adresa skupiny síťových rozhraní. Datagram adresovaný adresou typu *anycast* bude doručen jednomu z těchto rozhraní („nejbližšímu“ z hlediska

[8] *Time To Live* – omezuje maximální dobu existence IP datagramu.

topologie sítě).

- **multicast** – pakety adresované skupinovou adresou (*multicast*) jsou doručené všem rozhraním identifikovaných touto adresou.

V IPv6 protokolu neexistuje všeobecný oběžník (*broadcast*), ten je nahrazený skupinovými adresami.

Adresa u protokolu IPv6 je tvořena na podobném principu jako předchozí verze. Celý 128-bitový blok se rozděluje na menší části o délce 16 bitů, které jsou převedené na čtyřciferné čísla v hexadecimálním tvaru a jednotlivé části jsou odděleny dvojtečkou. Příklad IPv6 adresy je:

0123:0000:0000:0000:fedc:ba98:7654:3210

Již na první pohled se očekává, že uživatelé budou využívat automatických konfigurací a ručnímu psaní budou ušetřeni. Jelikož už tvůrcům se zdál takový zápas zdlouhavý a poměrně častá hodnota je nula, začaly se adresy psát ve zkráceném tvaru. Jsou dvě možnosti zkracování zápisu. U první se místo „0000“ píše jen „0“. Dokonce, když se vyskytuje několik nulových bloků za sebou, lze je nahradit zápisem „::“ (dvě dvojtečky). Například již zmíněnou adresu, můžeme zkrátit na

123:0:0:0:fedc:ba98:7654:3210

nebo dokonce v tomto případ, kdy jdou po sobě tři nulové skupiny, může psát

123::fedc:ba98:7654:3210

koncovou nulu však již vynechat nelze, jelikož by to změnilo adresu.

Adresy se zapisují ve tvaru prefixu (podobně jako u IPv4). Prefix je vždy následován lomítkem a počtem bitů, které tvoří adresu sítě. Např.

fa98::ff02/64

Adresný prostor byl rozdělen do několika skupin Tab. 2.3 Každá skupina sdružuje adresy se společnou charakteristikou. Příslušnost k jednotlivým skupinám určuje prefix adresy.

Největší část zabírají globální individuální adresy, jsou to adresy, které jednoznačně identifikují svého nositele v rámci celého Internetu, a musí být tedy celosvětově jednoznačné. Globální adresy jsou přidělovány hierarchicky, podobně jako veřejné adresy u protokolu IPv4, které jsou již dnes vyčerpané.

Jak již bylo zmíněno, IPv6 má také přepracovaný systém automatické konfigurace. Máme na výběr dokonce dva typy. První možností je stavová konfigurace, jejím základem je server spravující konfigurační parametry. Je to tedy obdoba dnes používaného DHCP. Pro stavovou konfiguraci byl navržen protokol DHCPv6. Pracuje na

Tab. 2.3: Části adresního prostoru v IPv6

prefix	význam
::/128	nedefinovaná adresa (nepoužívá se pro komunikaci)
::1/128	lokální smyčka (*lookback*)
2000::/3	jednoznačná globální adresa
fc00::/7	lokálně jednoznačné adresy
fe80::/10	individuální lokální linkové adresy
ff00::/8	skupinové adresy (*multicast*)

shodném principu jako předchozí verze, koncové zařízení rozešle na obecnou adresu dotaz ohledně svých konfiguračních parametrů a server mu je sdělí.

Druhou možností je bezstavová konfigurace, která je založena na tom, že jsou v síti směrovače, které vědí vše potřebné pro konfiguraci. Směrovače v pravidelných intervalech rozesílají ohlášení směrovače, ve kterých identifikují síť [10].

3 FTP

File Transfer Protocol je standardní síťový protokol určený pro přenos souborů mezi počítači přes počítačovou síť založenou na TCP spojení. Je postaven na modelu klient–server, může být provozován nezávisle na použitém operačním systému (je platformě nezávislý).

Protokol FTP vznikl v roce 1971 a byl definován v dokumentu RFC 959 v roce 1985. Je to protokol z aplikační vrstvy TCP/IP modelu, zapouzdřuje se do nižší, spolehlivé spojově orientované vrstvy TCP. Pro svůj běh využívá TCP porty 20 a 21. Port 21 slouží k navázání spojení a přenosu řídících příkazů (např. výpis adresáře, stažení souboru). Pro samotný přenos dat slouží port 20.

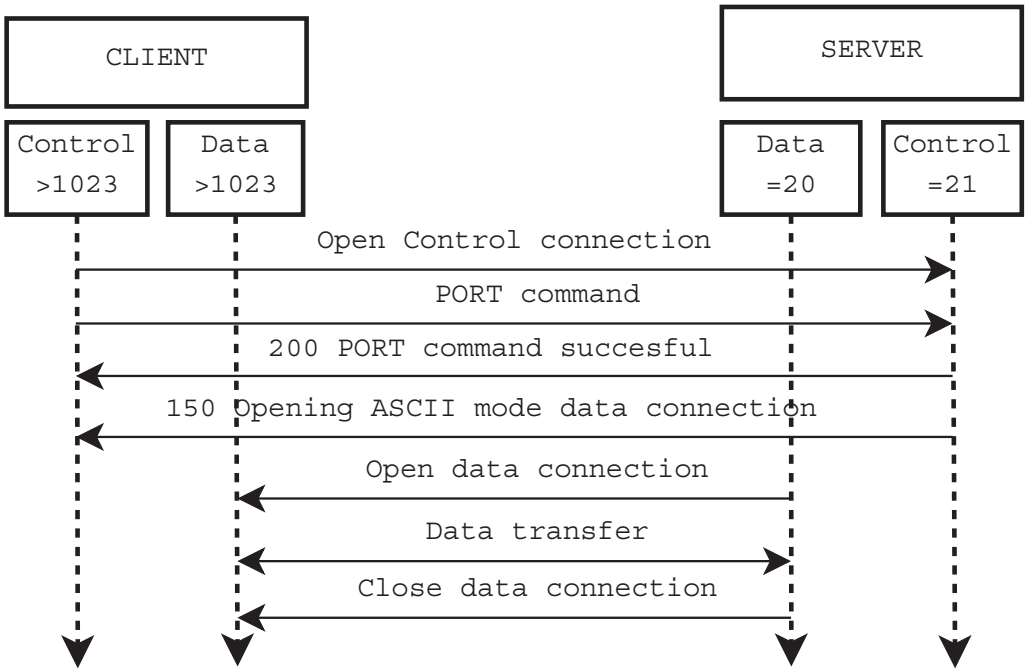
Přihlášení může být anonymní nebo autorizované pomocí uživatelského jména a hesla. Samotný přenos probíhá v binární nebo textové formě. Výhoda použití znakové formy spočívá v jednoduchém zajištění kompatibility různých operačních systémů s různým typem kódování. V dnešní době hraje protokol stále důležitou roli především v síti Internet. Velice často se setkáváme s jeho implementací u hostingových služeb, kdy odesíláme zdrojové soubory webových stránek na server.

Největší nevýhodou protokolu je jeho bezpečnost, hesla a soubory jsou ve standardním protokolu zasílána jako běžný text, je tedy poměrně jednoduché odposlechnout citlivá data. Z tohoto důvodu dnes existuje hned několik možností jak přenos zabezpečit.

3.1 Aktivní a pasivní režim

Jak již bylo zmíněno, FTP využívá pro svojí funkci příkazový a datový kanál. Příkazovým kanálem klient posílá na server požadavky. Datovým kanálem se přenáší požadovaná data, a může být vytvořen buď inicializací serverem nebo samotným klientem. Proto rozeznáváme dva režimy komunikace FTP protokolu: aktivní a pasivní.

- **Aktivní režim** – v této situaci navazuje připojení pro přenos dat server, klient naslouchá. Klient posílá příkaz `PORT`, který serveru říká na jakou IP adresu a na jaký port se server musí zpátky připojit ke klientu pro zahájení datového spojení. Po přijetí tohoto příkazu serverem se vytvoří datové spojení z lokálního portu 20 na klientskou IP adresu a port, který byl obsažen v příkazu `PORT`. Průběh spojení viz obr. 3.1 [12].
  1. Klient požádá správu volných portů v operačním systému o přidělení volného portu, ten musí být větší než 1023.
  2. Poté klient odesílá na server příkaz ve tvaru např. `PORT 109 238 148 132 4 12`, kde první čtyři skupiny čísel udávají IP adresu klienta a zbylá



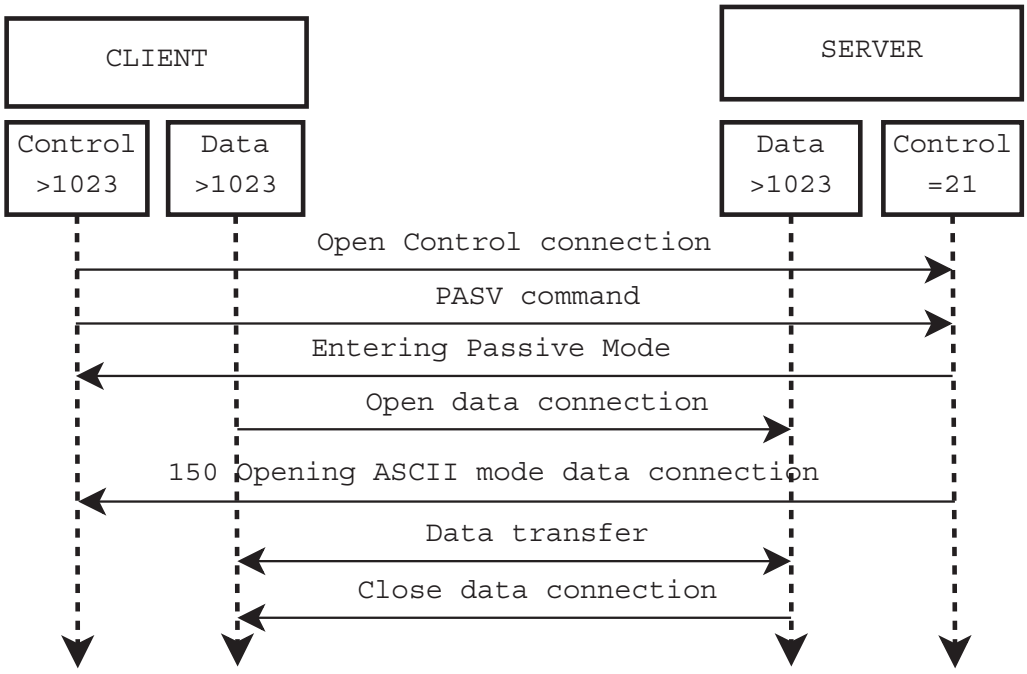
Obr. 3.1: Aktivní režim spojení s FTP serverem

dvojice čísel udává port, na kterém naslouchá. V tomto případě klient naslouchá na portu 1036 (Číslo portu je přenášeno ve dvou bajtech tj. 1036 je 04 0C šestnáctkově, ale FTP protokol uvádí každý bajt desítkově, tedy 4 12)

3. Server naváže protokolem TCP spojení s klientem na uvedené adrese a portu. Server tedy navazuje datové spojení s klientem!

Pro správnou funkci aktivního režimu je požadováno, aby měl klient veřejnou IP adresu. Při použití klienta z privátní sítě, kdy je jeho IP adresa překládána (NAT) může nastat problém. Taktéž se může problém objevit ve „špatně“ nakonfigurovaném firewallu pro příchozí spojení.

- **Pasivní režim** – v tomto režimu navazuje klient spojení jak pro příkazový kanál, tak i pro datový kanál. Tento režim se v Internetu vyskytuje častěji a je vhodný i pro klienty, kteří jsou schovaní za bránou NAT. Při zahájení spojení je odeslán příkaz `PASV`, který serveru říká, že komunikace má probíhat v pasivním režimu. Server si musí alokovat port pro datový kanál a klientu odesílá příkaz `PORT` se svojí IP adresou a číslem alokovaného portu, na který má klient navázat datové spojení. Průběh spojení na obr. 3.2 [12].



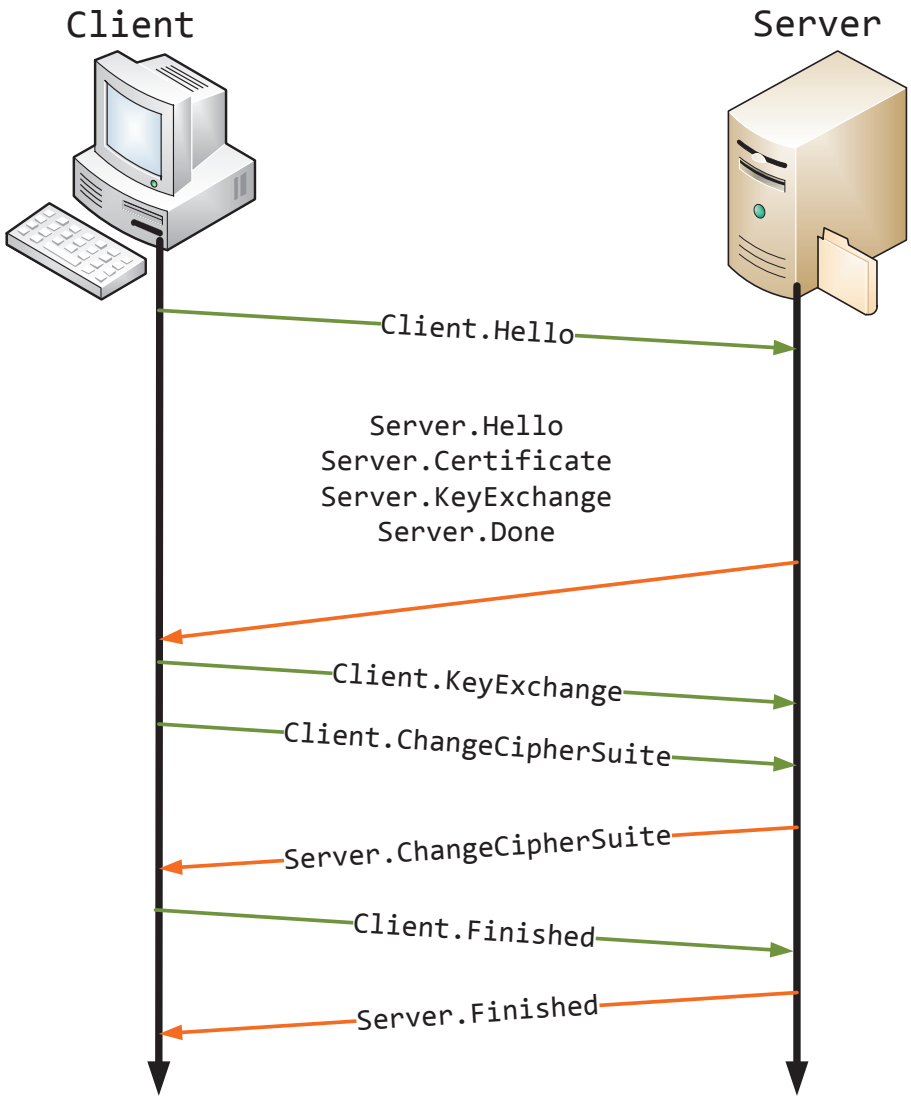
Obr. 3.2: Pasivní režim spojení s FTP serverem

3.2 Bezpečnost FTP

Samotný FTP protokol neobsahuje žádné mechanizmy pro zabezpečený přenos dat. Při připojování pomocí FTP protokolu jsou přihlašovací údaje (jméno a heslo) i data přenášena v textové podobě. Není tedy velký problém tyto informace odchytit a následně zneužít. Proto byly zavedeny různé rozšíření zajišťují zabezpečený přenos citlivých informací protokolem FTP.

3.2.1 FTP s SSL/TLS

FTP s SSL/TLS nebo zkráceně FTPS obsahuje plnou podporu pro kryptografické protokoly SSL i TLS (průběh inicializace TLS spojení na obr. 3.3) včetně ověřování veřejného klíče certifikátu na straně serveru i klienta. SSL/TLS protokol běží nad transportní vrstvou, ale pod aplikační vrstvou. Proto lze relativně snadno implementovat pro všechny druhy aplikačních protokolů jako je HTTP, Telnet, POP3, IMAP4 a FTP. Certifikáty jsou použity nejen jako veřejné klíče pro autentizaci asymetrickou šifrovací metodou a vygenerování klíčů sezení, ale také jako podpis totožnosti. Certifikát může vydat ověřená CA (Certifikační autorita), nebo si můžeme vytvořit vlastní certifikát. FTPS rozlišuje dvě metody připojení: explicitní a implicitní.



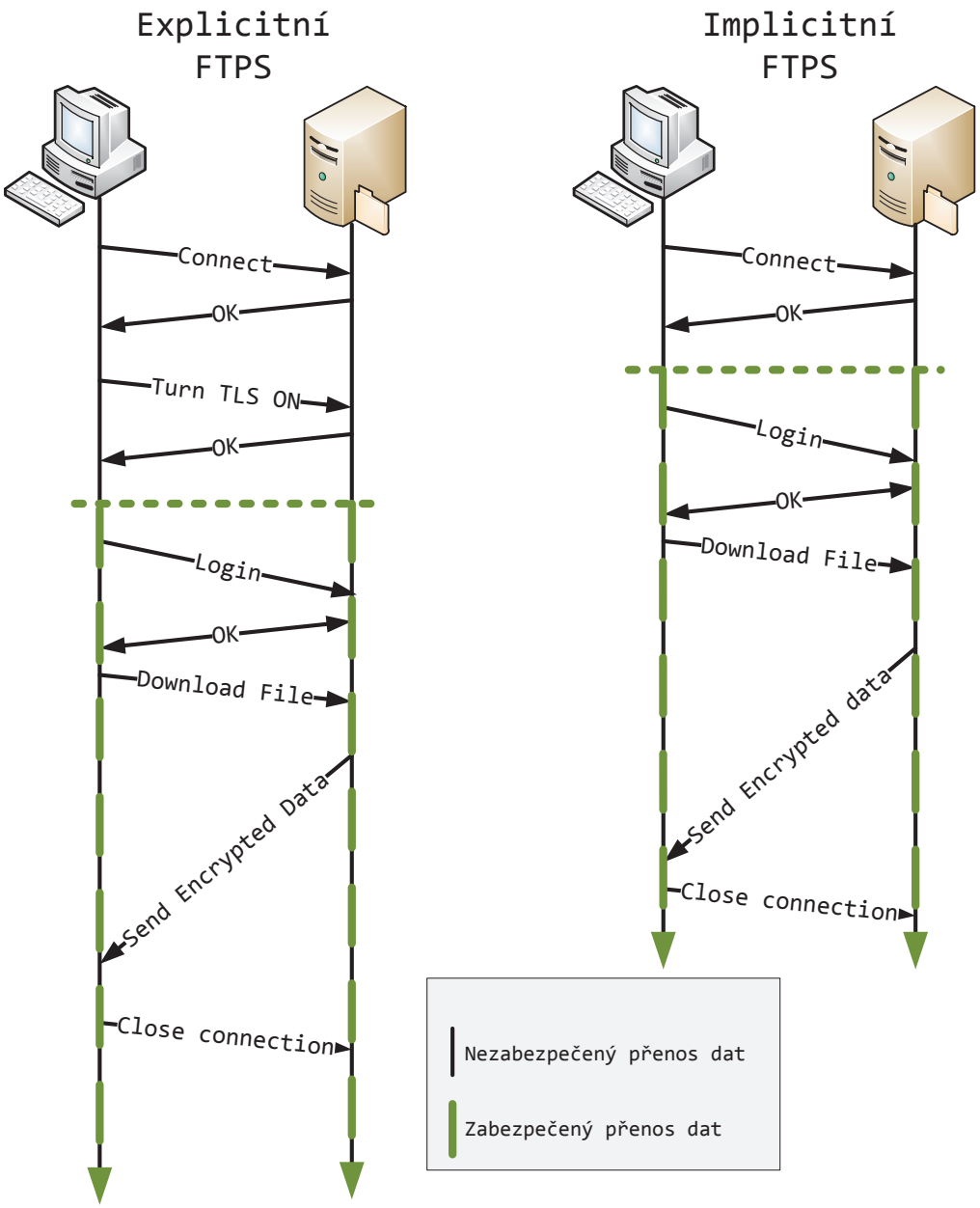
Obr. 3.3: Průběh TLS spojení

Explicitní FTPS

Je označováno také jako FTPES. V tomto režimu musí klient výslovně požádat o aktivaci TLS před tím, než budou posílána citlivá data, viz obr. 3.4. Jinak bude serverem odmítnut, nebo bude komunikace probíhat nezabezpečeně. Tento mechanizmus vyjednávání ověřování byl přidán v doporučení RFC 2228, který také zahrnuje nový příkaz `AUTH`. U explicitní metody se k připojení využívají výchozí porty pro FTP [13].

Implicitní FTPS

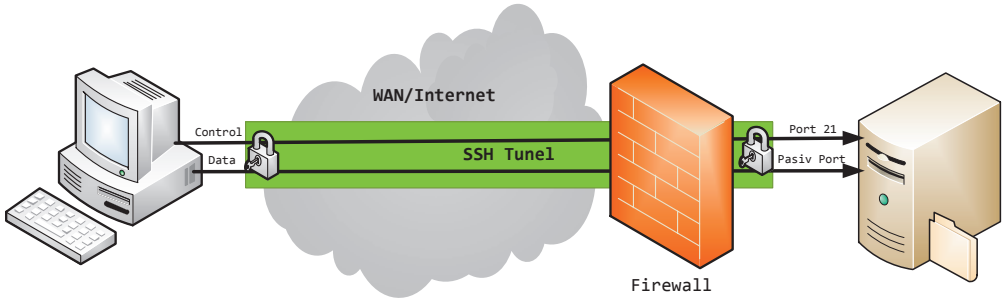
V implicitním režimu začne SSL připojení, jakmile se klient připojí k FTP serveru, viz obr. 3.4. Aby nedocházelo k nekompatibilitě s existujícími FTP klienty nepodporující TLS/SSL, využívá tento režim port 990 pro navázání řídícího kanálu a na portu 989 naslouchá pro datový kanál [13].



Obr. 3.4: Porovnání průběhu navázání spojení u explicitního a implicitního režimu

3.2.2 FTP přes SSH

Další možností zabezpečení FTP spojení je jeho tunelování skrze SSH[1] protokol. Je to v podstatě vytvoření zabezpečeného tunelu, do kterého jsou směrovány porty pro komunikaci. Pro vytvoření SSH spojení je obvykle nutné nainstalovat speciální software jak na straně serveru, tak i u klienta, např. OpenSSH. SSH tunelování se používá při pasivním režimu spojení FTP obr. 3.5 [14].



Obr. 3.5: Přenos FTP protokolu přes SSH kanál

Velice často se FTP přes SSH označuje zkratkou SFTP[2], což je špatně. Jelikož SFTP je náhradou za velmi jednoduchý protokol SCP[3] který rovněž pracuje s protokolem SSH ale s běžným FTP protokolem nemá nic společného (až na obdobné využití).

[1] *Secure Shell* – protokol, který slouží k zabezpečení dat při přenosu přes nedůvěryhodnou síť

[2] *SSH File Transfer Protocol* – novější verze protokolu SCP pro zabezpečený přenos dat

[3] *Secure Copy* – slouží k bezpečnému přenosu souborů mezi dvěma počítači pomocí protokolu SSH

4 DHCP

Dynamic Host Configuration Protocol je protokol aplikační vrstvy z rodiny TCP/IP a používá se pro automatickou konfiguraci síťových parametrů zařízení připojených do počítačové sítě. Výrazně tak ulehčuje práci síťovým správcům, jelikož připojené klienty informuje o sadě parametrů nutných pro komunikaci pomocí IP protokolu, správci odpadá ruční konfigurace každé stanice připojené do sítě. Klient tedy vůbec nemusí znát informace o síti, aby se k ní mohl bezproblémově připojit. Konfigurace se klientům přiděluje na určitou dobu, po vypršení časového intervalu si klient musí opět zažádat o obnovení konfiguračních parametrů (mohou se změnit) [15].

Významným způsobem tak zjednodušuje a centralizuje správu počítačové sítě, např. při přidávání nových stanic, hromadné změně parametrů atd. Pomocí DHCP se nejčastěji nastavují tyto parametry sítě:

- IP adresa
- maska sítě
- výchozí brána
- seznam dostupných DNS serverů
- a další údaje jako NTP, WINS, ...

DHCP server má celkem tři mechanismy pro přidělení správné IP adresy klientovi. Tyto mechanismy jsou:

- **Manuální alokace** – v tomto případě se nevyužívá DHCP serveru, ale konfigurace je ručně zapisována na jednotlivé stanice. Používá se tam, kde by změna adresy ohrozila chod sítě/aplikací např. servery, směrovače, tiskárny.
- **Statická alokace** – DHCP je nakonfigurováno tak, aby klientům přidělovalo stále stejnou IP adresu na základě jejich fyzické MAC adresy nebo podle DUID identifikátoru používaného v IPv6 sítí. Musí být tedy vytvořen seznam, který obsahuje záznamy, jaká IP adresa patří konkrétnímu identifikátoru. Tato metoda je vhodná všude tam, kde je klient stále připojen k síti, ale zároveň případná změna adresy neovlivní funkčnost aplikací/služeb.
- **Dynamická alokace** – DHCP server přiděluje volné IP adresy z vymezeného rozsahu. Adresy jsou přidělovány na omezenou dobu, dovoluje tedy právě nepoužité adresy přidělit jiným stanicím. Používá se například tam, kde je potřeba sdílet limitované množství adres mezi větší skupinu klientů, kteří ke své funkčnosti nepotřebují stále stejnou adresu [16].

V protokolu DHCP rozeznáváme 3 typy zařízení, které se mohou účastnit spojení:

- **Klient** – může být např. počítač, využívá DHCP pro získání konfiguračních parametrů.
- **Server** – odpovídá na žádosti DHCP klienta a poskytuje konfigurační parametry sítě.

- **Relay agent** – používá se v situaci, kdy existují dvě nebo více sítí oddělené směrovačem (viz obr. 4.1) a jen jedna síť obsahuje DHCP server. V takovém případě správce na směrovači zapne relay agenta a nastaví jej tak, aby všesměrové (broadcast) DHCP dotazy ze sítí bez DHCP serveru přeposílal DHCP serveru. Agent k přeposílanému dotazu přidá číslo sítě a masku sítě, na které klienta zaslechl, aby DHCP server poznal, ze kterého adresného rozsahu má klientovi adresu přiřadit.

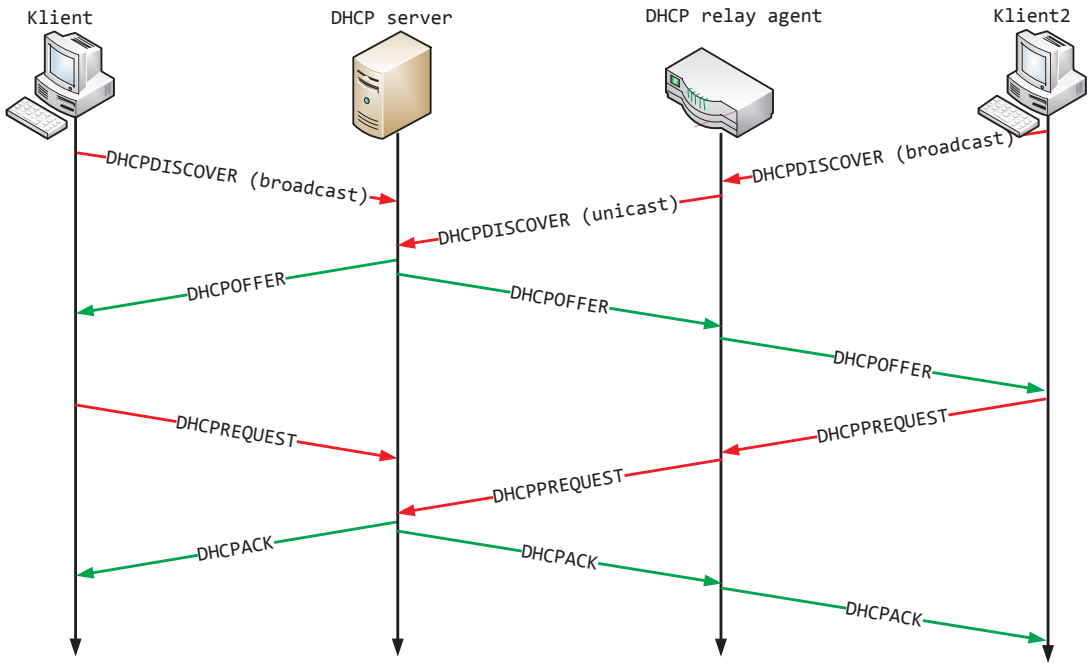
4.1 Princip výměny zpráv při konfiguraci klienta

Klienti žádají DHCP server o přidělení parametrů sítě, server u každého klienta udržuje záznam o zapůjčené IP adrese a čase, po který je přidělená adresa platná. Po uplynutí této doby zapůjčení (ang. lease time) musí klient zažádat o prodloužení platnosti, jinak může server tuto IP adresu přidělit dalšímu klientu. Při výměně zpráv klient komunikuje na UDP portu 68 a server naslouchá na UDP portu 67.

Prvotní proces přidělení IP adresy tvoří celkem čtyři kroky, viz obr. 4.1, při kterém probíhá výměna DHCP zpráv:

1. Klient vyšle broadcastem **DHCPDISCOVER** paket, kterým „osloví“ DHCP servery v síti.
2. Všechny DHCP servery, které obdrží DHCPDISCOVER zprávu, posílají klientu zprávu **DHCPOFFER** (nabídka). Zpráva obsahuje MAC adresu klienta, po které následuje IP adresa, kterou server nabízí, masku podsítě a časový limit po který bude adresa platná a IP adresu serveru, který nabídku učinil.
3. Po obdržení nabídek od serverů, klient musí oznámit pomocí zprávy **DHCPREQUEST** serveru, že nabídku přijal. Klient může dostat nabídku od spousty DHCP serverů, ale může přijmout pouze jednu, ostatní nabídky musí odmítnout. Zpráva se odesílá jako broadcast, ale musí obsahovat informaci o IP adrese serveru, ze kterého klient přijal nabídku na konfiguraci.
4. Po obdržení zprávy DHCPREQUEST serverem, se spouští finální proces konfigurace. Server potvrdí všechny parametry a následně je odešle klientovi ve zprávě **DHCPACK**. Tím je proces přidělení IP adresy dokončen.

Po vypršení platnosti IP adresy, může klient pomocí zprávy **DHCPREQUEST** zažádat o prodloužení doby zapůjčení. Tento dotaz server opět potvrzuje zprávou **DHCPACK**.



Obr. 4.1: Proces výměny zpráv při přidělení IP adresy

4.2 Automatická konfigurace v IPv6

V prostředí IPv4 existuje jen jediná možnost automatické konfigurace síťových parametrů klientů připojených do sítě, ta se nazývá zkráceně DHCP. Při použití nového protokolu IPv6 máme na výběr ze dvou různých metod pro automatickou konfiguraci, stavovou a bezstavovou metodu.

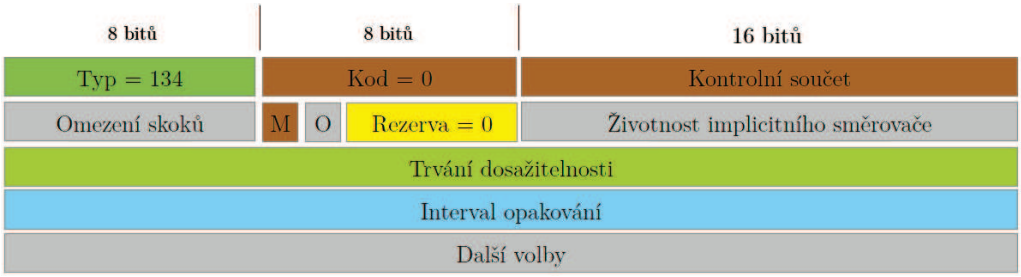
Stavová konfigurace není nic nového, její základ je server, který obstarává konfigurační parametry a ty poté klientům na požádání sděluje. Princip všech mechanizmů je podobný jako u DHCP v prostředí IPv4, který byl již popsán výše. Proto se stavová konfigurace IPv6 označuje jako DHCPv6.

Naopak bezstavová konfigurace je novinkou v tomto protokolu. Základem je zařízení (např. směrovač), které v pravidelných intervalech vysílá informace o síti (tzv. *Router advertisement*) jako prefix sítě nebo výchozí brána. Po obdržení informace, v jakém prefixu se klient nachází, si dokáže klient vygenerovat vlastní IPv6 adresu, bez použití DHCP serveru [10].

4.2.1 Bezstavová konfigurace

Jejím základním pilířem je objevování sousedů. Každý směrovač v určitých intervalech rozesílá do sítí, k nimž je připojen, takzvané *ohlášení směrovače* (RA - Router Advertisement). V něm jsou obsaženy základní informace, především prefixy adres dané sítě, a zda on sám může sloužit pro předávání paketů ven (*default gateway*).

Z ohlášení směrovačů, případně při startu může klient aktivně požádat pomocí *výzvy směrovači* (RS - Router Solicitation), se počítač dozví, jaké adresy používá zdejší síť. K nim si doplní identifikátor rozhraní (typicky 64 bitů), který si jednoznačně vygeneruje ze své MAC adresy. Tak získá platné IPv6 adresy pro své rozhraní. Veškerá komunikace při vyjednávání parametrů sítě probíhá s využitím protokolu ICMPv6. Jednoznačnost IPv6 adresy ověří pomocí detekce duplicit. Pomocí výzvy sousedovi se dotáže, zda vytvořenou adresu již někdo nepoužívá, pokud nepřijde ze sítě žádná odpověď, je adresa volná a může se v síti používat [10].



Obr. 4.2: Paket ohlášení směrovače

Na obr. 4.2, je zobrazena část paketu používaných jako ohlášení směrovače. Mezi jednu z nejdůležitějších položek paketu patří Životnost implicitního směrovače (*Router Lifetime*). Udává dobu, jak dlouho bude směrovač sloužit jako implicitní pro uzly z této sítě. Za nejdůležitější položky paketu RA se dá považovat dvojice příznaků *M* a *O*. Příznak *M* (Managed address configuration) oznamuje, že všechny konfigurační parametry (IPv6 adresu, masku atd.) přidělí DHCPv6. Příznak *O* (Other stateful configuration) udává, zda se budou některé dodatečné informace načítat z DHCPv6 serveru.

Možná si většina z vás teď řekne, k čemu to vlastně je, tak buď použiji stavovou nebo bezstavovou konfiguraci, tak proč používat kombinace těchto dvou metod. Odpověď je jednoduchá, jelikož ani jedno řešení není „dokonalé“. U bezstavové konfigurace nedokáže RA poskytovat informace o DNS serverech (nově je sice do RA implementován RDNSS podle doporučení RFC 6106, nicméně některé operační systémy nedokážou s těmito informacemi pracovat. Jedná se především o operační systémy od společnosti Microsoft). Při použití stavové konfigurace (DHCPv6) obdrží

Tab. 4.1: Význam možných kombinací příznaků M a O

M	O	Význam
1	–	Informace bude poskytovat pouze DHCPv6 server
0	**1**	Kombinace bezstavové konfigurace (adresa, prefix, výchozí brána) a DHCPv6 (ostatní parametry, např. DNS)
0	**0**	Použití pouze RA, DHCPv6 není k dispozici

klient téměř všechny potřebné parametry pro běh v síti, až na informace o výchozí bráně, což je také vážný nedostatek. Proto se nejčastěji používá kombinace obou metod, např. informace o prefixu a výchozí bráně nám sděluje RA parametry DNS serverů právě DHCPv6.

4.2.2 DHCPv6

DHCPv6 realizuje v prostředí IPv6 tak zvanou stavovou automatickou konfiguraci. Jak již bylo zmíněno, princip protokolu je velice podobný protokolu DHCP, který známe z prostředí IPv4. Pro svoji činnost vyžaduje specializovaný server, který přiděluje jednotlivým klientům jejich konfigurační parametry.

Podobně jako v IPv4 probíhá základní DHCPv6 konfigurace ve čtyřech krocích. Jen s tím rozdílem, že v prvním kroku klient neodesílá DISCOVER paket na broadcastovou adresu (v prostředí IPv6 se broadcastové adresování nepoužívá), ale na standardní skupinovou adresu všech DHCP agentů a serverů na lince, která je ff02::1:2. Zbytek procesu automatické konfigurace je stejný jako v IPv4.

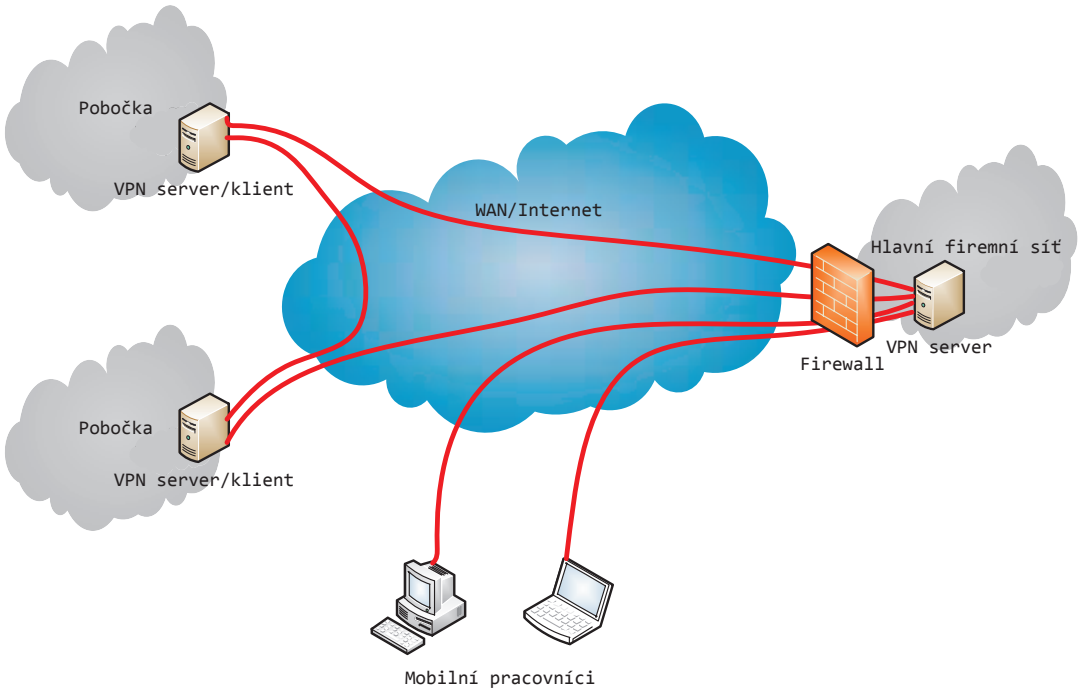
Významnou změnou v protokolu DHCPv6 je způsob identifikace klientů a serverů. Dříve se k tomuto účelu používala MAC adresa, na jejímž základě se např. mohla definovat statická adresa pro konkrétní ethernetovou adresu (zařízení v sítí obdrželo vždy stejnou IP adresu). Pro tento účel se v DHCPv6 zavádí pojem *Unique Identifier* (DUID), který by měl být stálý a neměl by se měnit ani při výměně síťové karty počítače. DUID je uloženo v daném operačním systému (u Windows je informace uložena v registru). Je definováno několik způsobů, jak DUID identifikátor vytvořit, např. výrobce udělí zařízení jedinečnou identifikační hodnotu (výrobní číslo), nebo se DUID vytvoří na základě linkové adresy, což nejvíce odpovídá praxi ze světa IPv4 [10].

5 VPN

Virtual Private Network(česky Virtuální privátní síť) je prostředek k propojení privátních sítí (LAN) napříč nedůvěryhodnou počítačovou sítí Internet. Je možné tedy dosáhnout stavu, kdy počítače v Internetu budou moci mezi sebou komunikovat, jako kdyby se nacházely v jedné uzavřené privátní síti viz obr. 5.1.

Veškerá komunikace je šifrovaná a ověřování probíhá pomocí digitálních certifikátů. VPN je síťové tunelování, kdy se prostřednictvím standardního síťového spojení vytvoří virtuální linka mezi dvěma zařízeními, v rámci které pak lze navázat další síťová spojení.

VPN může mít široké využití, používají ji např. firmy pro vzdálené připojení svých mobilních pracovníků do firemní sítě. Připojení může probíhat z jakéhokoliv místa na světě připojeného k Internetu. Data ve VPN tunelu jsou šifrovaná a nemohou být zneužita neoprávněnou osobou. VPN spojení může také mezi sebou propojovat jednotlivé pobočky firmy, tímto řešením může firma ušetřit náklady za pronajatou linku [17] [18].



Obr. 5.1: VPN spojení

Rozlišujeme dva typy sítě VPN:

- **Remote-Access** – sítě pro připojení vzdálených uživatelů k síti. Uživatelé se mohou např. z domu připojit se serverem v organizaci. Spojení je většinou

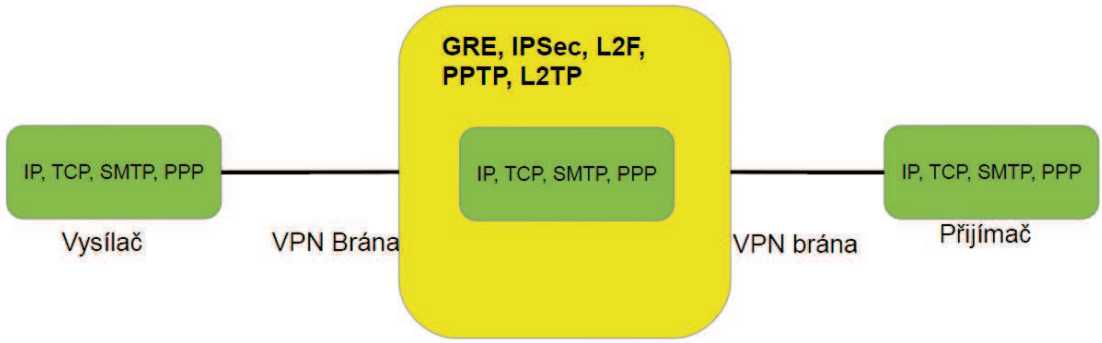
dvoubodové (point-to-point), používají se protokoly druhé vrstvy OSI modelu (L2TP, L2F, PPTP) nebo řešení založené na vyšších vrstvách (IPSec, SSL VPN).

- **Site-to-Site VPN** – v rámci architektury Site-to-Site VPN můžeme rozlišovat dva druhy použití sítě VPN:
  - **Intranet** – slouží ke spojení poboček v různých geografických lokalitách pomocí spojení VPN. Spojení vzniká vytvořením tunelu mezi hraničními uzly (směrovač, firewall) a používají se řešení založené na IPSec, MPLS.
  - **Extranet** – vytvoření sítě vně podnikového intranetu, přístupný pouze důvěryhodným partnerům [19].

5.1 Tunelování VPN

Tunelování je obecně technika používaná v počítačových sítí, která zapouzdřuje jedno nebo více síťových spojení do jiného síťového spojení. VPN využívá mechanizmu tunelování mezi koncovými klientskými sítěmi, kdy tunel představuje logický dvoubodový spoj, který ve skutečnosti může vést přes komplexní propojené sítě. Tunel definují dva koncové body: vstup a výstup z tunelu a mechanizmus přenosu paketu tunelem. Hraniční body tunelu zodpovídají za zapouzdření/rozbalení původních paketů přenášených přes transportní síť, autentizaci a řízení přístupu a dojednávání dalších bezpečnostních služeb.

Tunelování se používá jako transportní mechanizmus při budování VPN, ale lze jej využít i pro zajištění bezpečnosti, kdy se nezabezpečený paket vkládá do bezpečného paketu (nejčastěji zašifrovaného). K zabalení se používají protokoly GRE, IPSec, L2F, PPTP, L2TP.

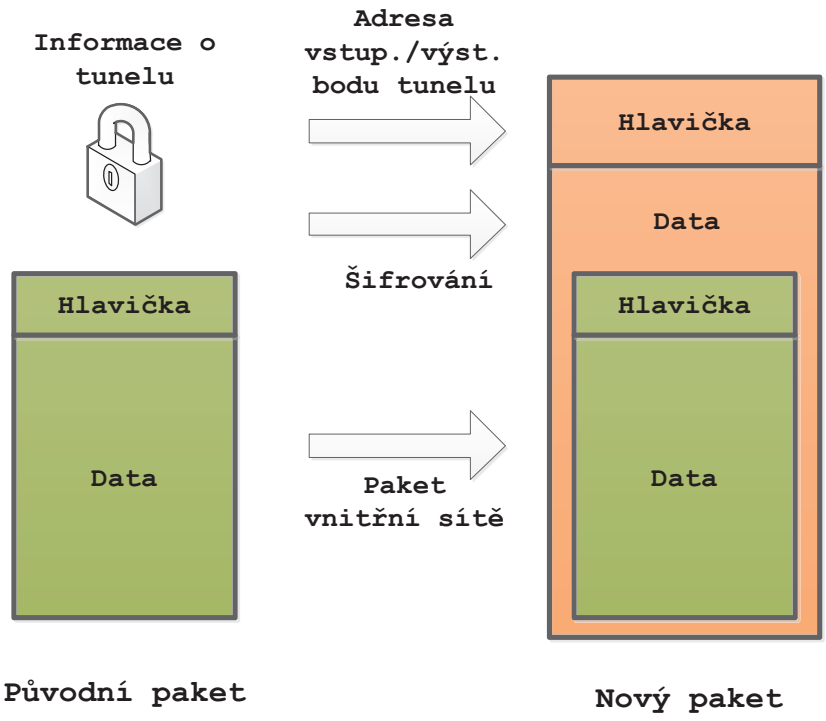


Obr. 5.2: Jednoduché schéma zapouzdření paketů

Nejběžněji používaným typem tunelování pro spojení mezi zdrojovým a cílovým směrovačem je GRE (Generic Routing Encapsulation) [21].

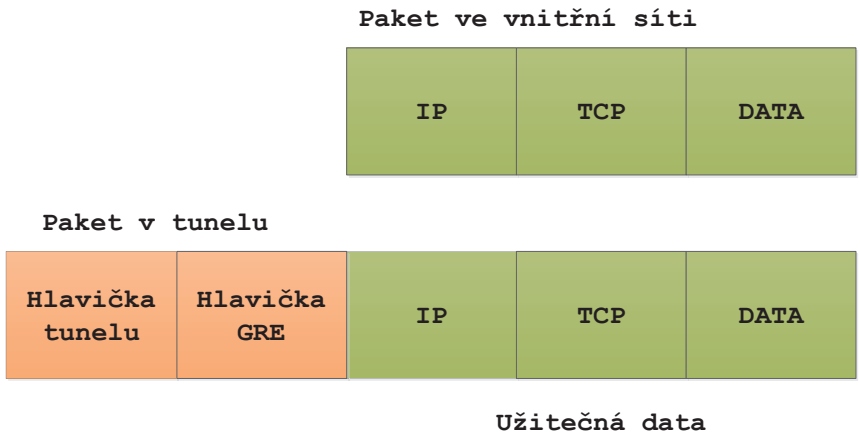
5.1.1 GRE

Generic Routing Encapsulation je protokol ze skupiny TCP/IP určený k zapouzdřování paketů jednoho protokolu do protokolu druhého. Nejčastěji se používá ve VPN, k přenosu IPv6 paketů v sítí IPv4 a k tunelování obecně. Originální paket je užitečným obsahem výsledného paketu viz obr. 5.3.



Obr. 5.3: Přenos dat tunelem

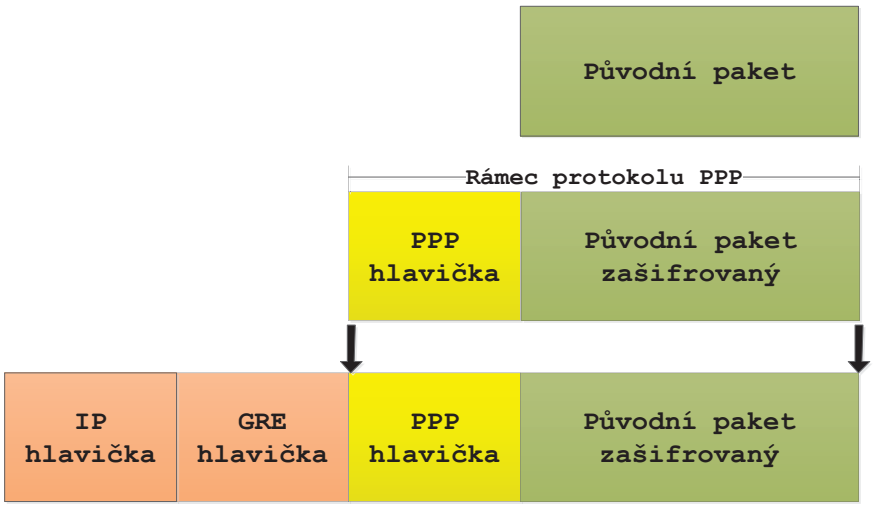
Tunely GRE jsou budovány směrovači, které slouží jako vstupní a výstupní body do páteřní sítě pro jednotlivé části VPN. Speciálně zabalené pakety přenášené tunelem obsahují přídavnou GRE hlavičku (GRE Header) a cílovou adresu, odpovídající směrovači na konci tunelu. V koncovém bodě tunelu dojde k rozbalení paketu a následné směrování paketu do cíle již pokračuje podle informací ve své původní IP hlavičce [18].



Obr. 5.4: Vytvoření GRE paketu

5.2 PPTP

Point to point tunneling protokol je jednou z nejrozšířenějších realizací virtuálních privátních sítí, využívá zapouzdření dat na druhé (linkové) vrstvě OSI modelu. Tunely operující na druhé vrstvě mohou teoreticky tunelovat libovolný druh paketu. Ke spojení s druhou stranou tunelu využívá PPTP virtuální point-to-point (PPP) zařízení. PPTP používá pro zapouzdření mechanizmus GRE.



Obr. 5.5: Zapouzdření protokolu PPTP

Původní paket je nejprve pomocí protokolu PPTP zapouzdřen do diagramů PPP.

Rámec PPP je vnořen do hlavičky protokolu GRE a hlavičky IP protokolu. V hlavičce protokolu IP je uvedena zdrojová a cílová adresa klienta a VPN serveru.

PPTP je realizováno dvěma síťovými relacemi, jedna pro data, druhá relace na TCP portu 1723 je používána pro zahájení a řízení GRE relace. Protože PPTP vyžaduje dvě síťové relace, je pro něj někdy nesnadné proniknout síťovým firewallem. PPTP umožňuje pouze jeden tunel současně mezi komunikujícími stranami [22].

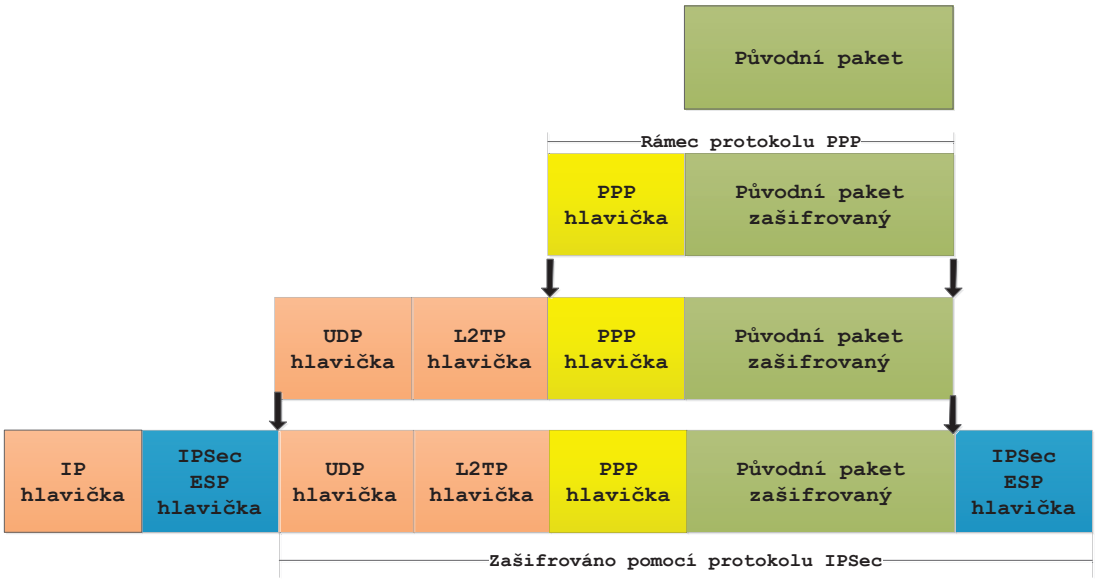
Spojení PPTP jsou ověřována pomocí autentizačních metod PAP, CHAP, Microsoft CHAP V1/V2 nebo EAP-TLS. Přenos je povinně chráněn šifrováním MPPE (Microsoft Point-to-Point Encryption). Nicméně už jsou známy zranitelnosti protokolu PPTP a některé autentizační metody (CHAP a MSCHAPv2) lze napadnout slovním útokem. Zatím bezpečnou volbou pro PPTP představuje certifikované EAP-TLS.

Protokol PPTP patří mezi nejrozšířenější realizace a to zejména díky společnosti Microsoft, která ho propaguje ve svých operačních systémech. Jeho velkou výhodu je poměrně snadná rozšiřitelnost a implementace na straně klienta, proto je protokol PPTP často k vidění u firem s požadavkem na snadný přístup svých zaměstnanců do firemní sítě.

5.3 L2TP

Layer 2 Tuneling Protocol kombinuje výhody protokolu PPTP a L2F (od firmy Cisco). Sám o sobě neposkytuje žádné bezpečnostní mechanismy, ale může být kombinován s jinými protokoly zajišťující bezpečnost, nejčastěji se k tomu využívá protokol IPSec. IPsec zajišťuje důvěryhodnost, integritu a autentizaci přenášených dat. Kombinace protokolu L2TP a zabezpečení IPSec se označuje jako připojení L2TP/IPSec. Zapouzdření paketů připojení L2TP/IPSec se skládá ze dvou vrstev viz obr. 5.6:

1. **Zapouzdření L2TP** – Rámec PPP je vnořen do hlavičky protokolu L2TP a do hlavičky protokolu UDP.
2. **Zapouzdření IPSec** – Zpráva protokolu L2TP je vnořena do hlavičky a do koncové části protokolu IPSec ESP (Encapsulating Security Payload) a opatřena koncovou částí zabezpečení IPSec, která zajišťuje integritu a ověření zprávy, a koncovou hlavičkou protokolu IP. V hlavičce protokolu IP je uvedena zdrojová a cílová adresa IP, která odpovídá klientovi a serveru VPN [22].



Obr. 5.6: L2TP/IPSec zapouzření paketu

5.4 IPSec

IP security je bezpečnostní rozšíření IP protokolu založené na autentizaci a šifrování každého IP datagramu. V architektuře OSI se jedná o zabezpečení na síťové vrstvě, poskytuje proto transparentně bezpečnost jakémukoliv typu přenosu. IPSec využívá různé kryptografické technologie a zajišťuje následující bezpečnostní služby:

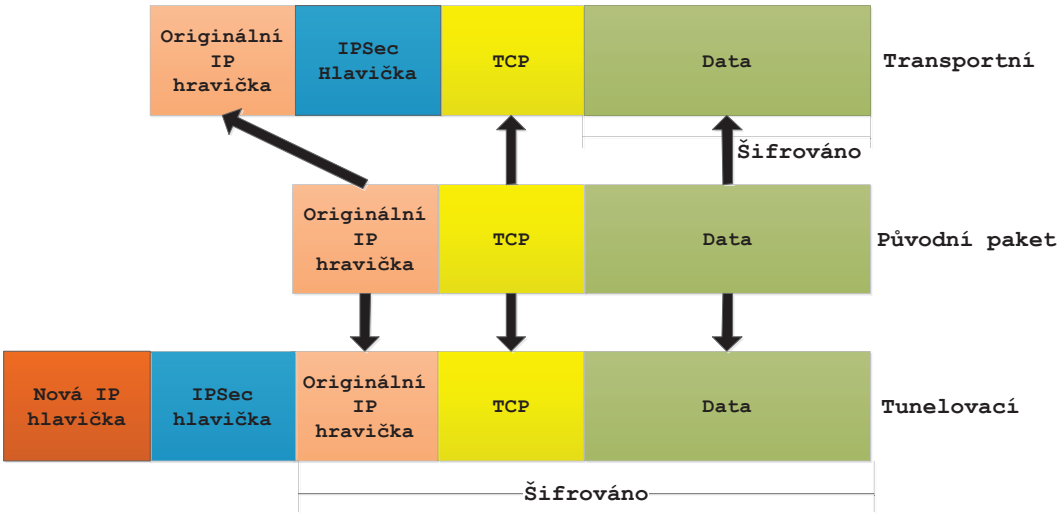
- Důvěryhodnost dat (šifrováním přenášených dat)
- Integritu dat (autentizační hlavičkou)
- Autentizaci zdroje dat
- Ochranu proti opakovanému posílání dat
- Kontrolu přístupu

Základem IPSec jsou protokoly *AH* (Autentification Header) a *ESP* (Encapsulating Security Payload), které pozměňují původní IP hlavičky paketů a celkově paket zabezpečují.

1. **AH** – zajišťuje autentizaci a integritu dat odesílatele, ochranu proti opětovnému poslání datagramu, ale vlastní data nejsou šifrována. Využívají se k tomu hashovací algoritmy HMAC-MD5 a HMAC-SHA1.
2. **ESP** – Poskytuje důvěryhodnost zapouzdřených dat IP datagramu, pro šifrování využívá symetrické šifrovací algoritmy DES, 3DES a asymetrickou RSA.

Pro IPsec tunel jsou definovány dva přenosové módy. Jde to transportní a tunelovací mód. Z obr. 5.7 je zřejmý rozdíl mezi oběma módy. Výhodou transportního módu jsou nižší nároky na přenosové pásmo. Tím, že jsou k dispozici informace

o cílovém zařízení, lze rovněž během přenosu paketu sítí aplikovat některé nadstandardní mechanismy (např. QoS). V tunelovacím módu je celý IP datagram šifrován a je vytvořena nová hlavička. Tento mód zase umožňuje, že některá zařízení mohou fungovat jako IPsec proxy. Tyto zařízení pak pošlou rozšifrovaný paket cílovému zařízení. Tímto způsobem lze používat IPsec bez nutnosti implementovat jej na všechna koncová zařízení. V současnosti je nejčastěji používán tunelovací mód [17].



Obr. 5.7: Tunelovací a transportní mód

5.5 SSL

Secure Socket Layer je bezpečnostní protokol, který definuje šifrování a autentizaci komunikujících stran. Následovníkem SSL je protokol TLS (Transport Layer Security), TLS 1.0 je až na malé rozdíly shodný se SSL ve verzi 3.0. Často se proto uvádí složený zápis SSL/TLS.

Z hlediska bezpečnosti nabízí SSL slabší ochranu než IPSec, ale je méně náročná na implementaci. Vhodné pro zabezpečení webové nebo emailové komunikace, případně sdílení souborů. Pro zajištění bezpečnosti využívá protokol SSL asymetrickou a symetrickou kryptografii. Před vlastním přenosem si obě strany (aplikace) ověří totožnost pomocí asymetrické kryptografie (za pomoci veřejných a soukromých klíčů). Po úspěšné autentizaci dále pokračují v komunikaci pomocí symetricky šifrovaných zpráv (např. s využitím algoritmů 3DES, RC4). Integrita přenášených dat pak bývá zajištěna pomocí hashovacích funkcí (např. SHA, MD5).

Nejznámější implementací virtuálních privátních sítí založených na protokolu SSL je OpenVPN. OpenVPN umožňuje vytvářet topologie sítí remote-access a Site-to-Site, pracuje na druhé nebo třetí vrstvě referenčního modelu OSI zajištěné protokoly SSL/TLS, podporuje autentizační metody založení na certifikátech, smart kartách a autentizaci jménem/heslem. Spojení lze tunelovat přes téměř každý firewall, pro příchozí spojení stačí mít otevřený jeden port. OpenVPN je multiplatformním nástrojem.

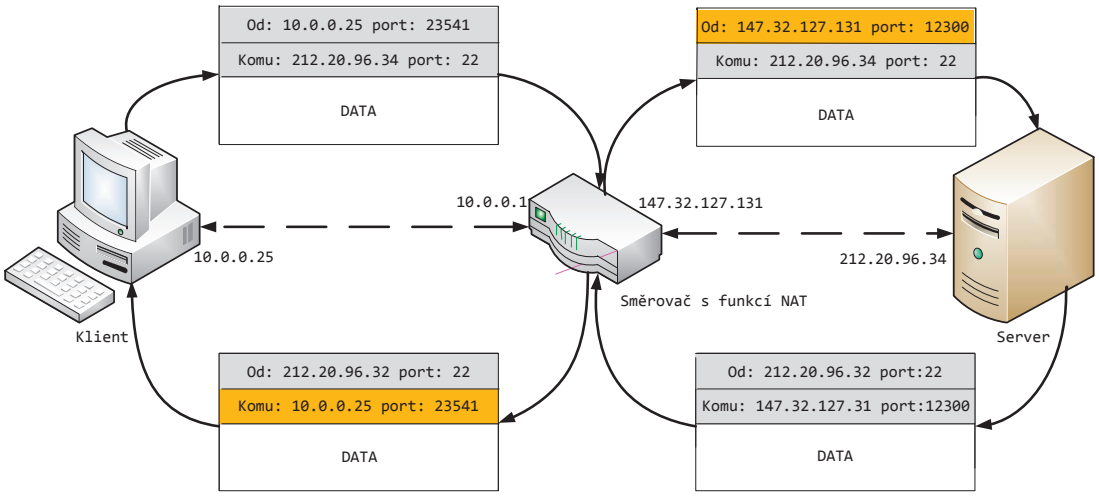
6 NAT

Network Address Translation (česky překlad síťových adres) je proces předkládání adresy z vnitřního adresného rozsahu do veřejného a naopak. Většinou se používá pro přístup více počítačů z lokální sítě do Internetu pod jedinou veřejnou IP adresou. NAT upravuje síťový provoz, přepisuje zdrojovou a cílovou IP adresu a často mění i čísla TCP/UDP portů příchozích paketů. Pracuje na 3. vrstvě referenčního modelu OSI.

Funkci překlad síťových adres plní běžně směrovače nebo různé systémy nazvané jako brány (gateway). Tento mechanizmus vznikl jako důsledek omezeného počtu veřejných IP adres. Dovoluje nám tedy „zamaskovat“ celou vnitřní síť (např. rozsah 192.168.1.0/24) za jedinou veřejnou IP adresu, pod kterou bude každé zařízení vystupovat v Internetu. Z toho vyplývá, že není možné se zvenčí přímo spojit s počítačem za NATem, bez dalších úprav.

S NATem velice blízce souvisí i PAT (Port address translation). Díky překladu portů, můžeme právě „zamaskovat“ naši vnitřní síť za zařízení s veřejnou IP adresou. Pro jednotlivá spojení si vytváří páry veřejných a soukromých adres a jejich síťových portů viz obr. 6.4. PAT je podmnožinou NATu, někdy se jejich spolupráce označuje jako NAPT, případně souhrnně jen jako NAT [23] [24].

6.1 Princip činnosti

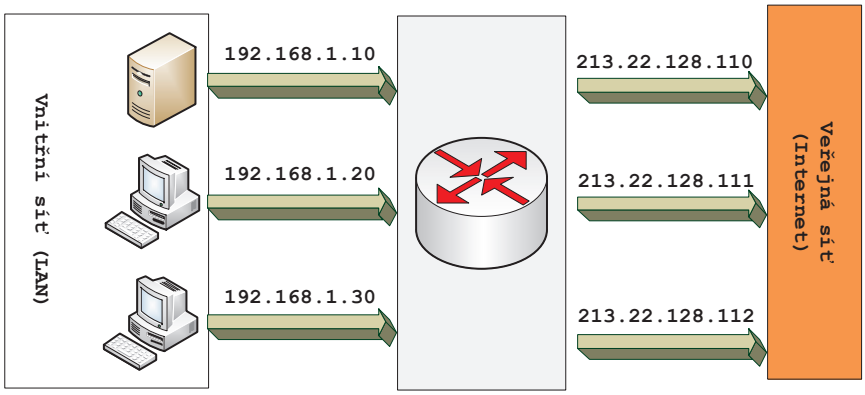


Obr. 6.1: Princip činnosti NAT

- Klient z vnitřní sítě odešle paket do Internetu (212.20.96.34) se svojí adresou (10.0.0.25) na místě odesílatele.
- Při průchodu NATem se přepíše adresa a port odesílatele (147.32.127.131) na vlastní veřejnou IP adresu a číslo portu, které přidělil odesílateli. Zároveň si do tabulky zapíše, jaká vnitřní adresa a port se právě překládají (10.0.0.25:23541<–>147.32.127.131:12300).
- Odpověď se poté směruje na ten samý port, NAT opět přepíše hlavičky, do cílové adresy a portu vloží informaci podle převodní tabulky a paket předá do lokální sítě.
- Klientovi tudíž dorazí odpověď s jeho adresou a nepozná tedy, že pakety byly během přenosu upravovány [23].

6.2 Statický a dynamický NAT

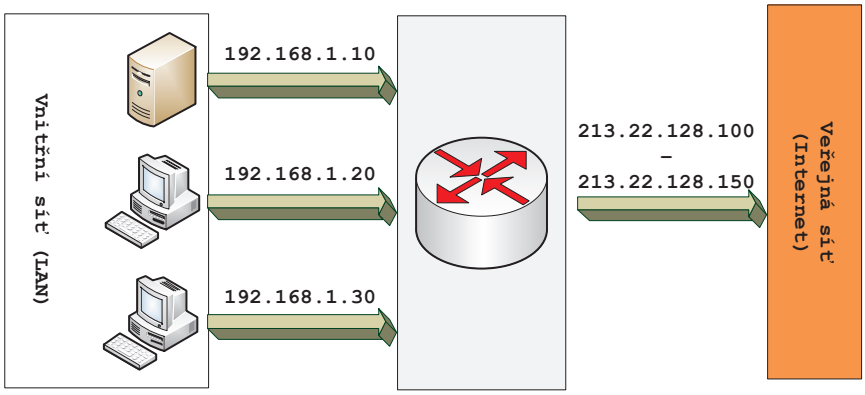
Při **statickém** NATu obr. 6.2 se mapuje vnitřní IP adresa na konkrétní vnější IP adresu, jde o tzv. mapování 1:1. Pří statickém mapování nedochází ke změně čísla portů. Pro každou vnitřní adresu musí být přidělena jedna vnější (veřejná) IP adresa. Je to užitečné, když požadujeme, aby zařízení ve vnitřní síti bylo přístupné z vnější sítě (Internetu).



Obr. 6.2: Statický překlad adres

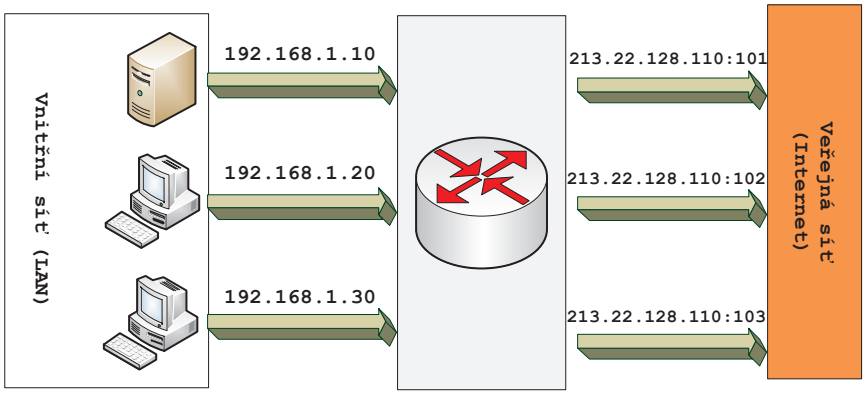
V statickém NATu, počítač s adresou 192.168.1.20 bude vždy přeložen na veřejnou IP adresu 213.22.128.111.

V **dynamickém** NATu obr. 6.3 se mapují vnitřní IP adresy na určitý přidělený rozsah veřejných adres. Počet současných spojení je omezen počtem veřejných IP adres. Počítač s adresou 192.168.1.20 může být přeložen na některou volnou adresu z veřejného rozsahu 213.22.128.100-213.22.128.150 [25].



Obr. 6.3: Dynamický překlad adres

Jednou z forem dynamického NATu je **PAT** (nebo také NAPT, Masquerade NAT nebo IP maškaráda). Princip je jednoduchý, více vnitřních adres se mapuje na jedinou vnější adresu, ale na různých portech. Komunikace musí být iniciována z vnitřní sítě, příchozí datagramy, bez záznamu v NAT tabulce jsou zahozeny.



Obr. 6.4: IP maškaráda

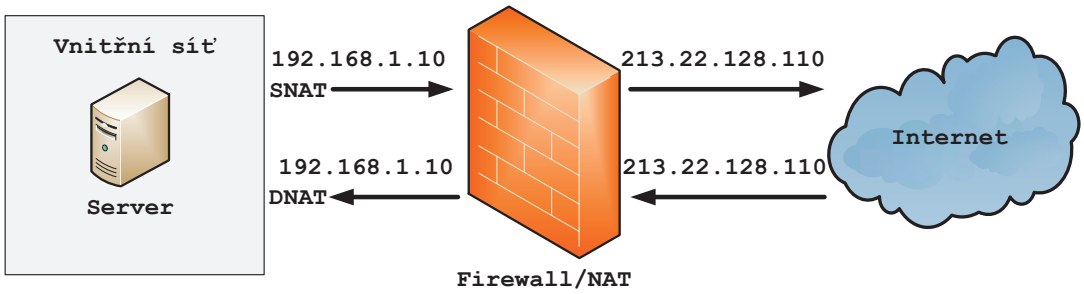
Každý počítač ve vnitřní síti se překládá na stejnou IP adresu 213.22.128.110, ale s jiným číslem portu [25].

6.3 Zdrojový a cílový NAT

Další možností, jak se může překlad adres dělit, je podle toho, jaká IP adresa se při průchodu NATem přepisuje. Zdrojový NAT (SNAT) přepisuje při průchodu zdro-

jovou IP adresu odesílatele. Např. IP adresa odesílatele z vnitřní sítě, se přepíše vnější adresou směrovače (s funkcí NAT), který je připojen do internetu. Všechny pakety z vnitřní sítě odesílané do internetu mají po průchodu směrovačem stejnou zdrojovou adresu. Nejznámější metodou SNATu je IP maškaráda.

Cílový NAT (DNAT) se používá, když potřebujeme měnit IP adresu adresáta datagramu. Používá se to při přesměrovávání datagramů na jiného adresáta. DNAT se obvykle používá při zveřejnění služby, umístěné v privátní síti, pro veřejně přístupnou IP adresu. Např. budeme mít v privátní síti WWW server s adresou 192.168.1.10, které je namapována veřejná IP 213.22.128.110. Všechny dotazy na WWW server z Internetu mají jako cílovou adresu 213.22.128.110, po průchodu směrovačem (s funkcí NAT) se cílová adresa změní na lokální adresu 192.168.1.10 a putuje vnitřní sítí až k serveru [25].

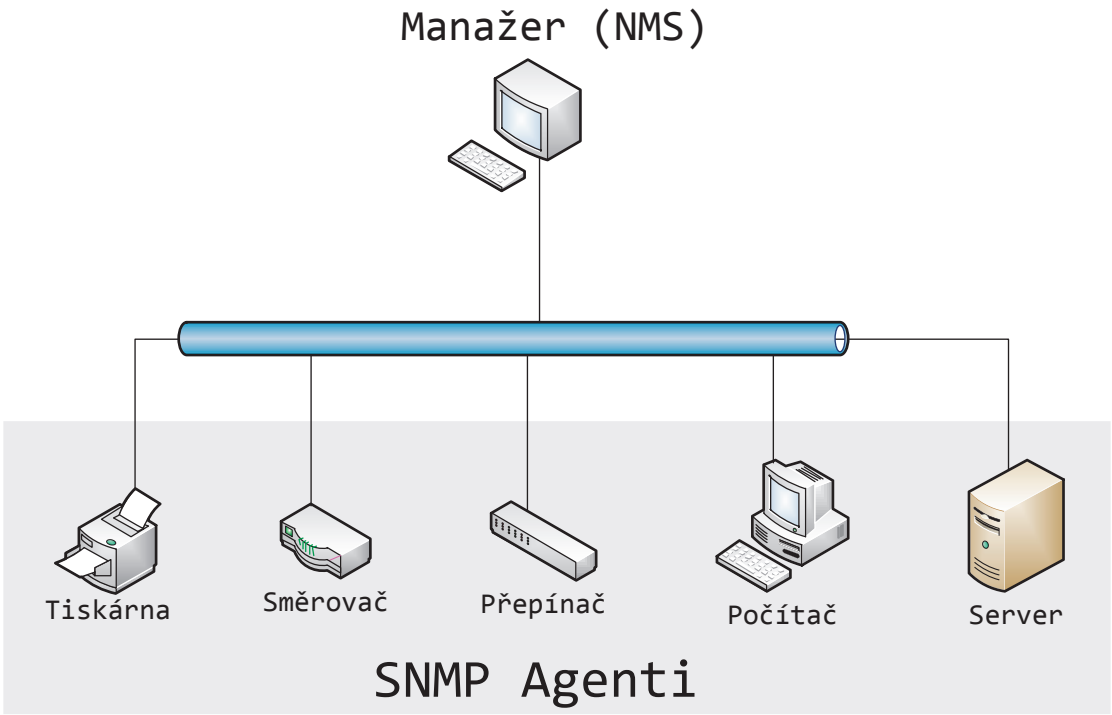


Obr. 6.5: Zdrojový a cílový NAT

7 SNMP

Simple Network Management Protocol je standardizovaný internetový protokol aplikační vrstvy, který slouží ke správě a získávání informací týkajících se stavu síťových prvků. Umožňuje průběžný sběr nejrůznějších informací pro potřeby správy sítě, a jejich následné vyhodnocování. Je implementován do většiny zařízení určených pro komunikaci v síti jako např. ve směrovačích, přepínačích, tiskárnách, operačních systémech, meteostanicích atd. Na tomto protokolu jsou dnes založeny téměř všechny nástroje pro správu sítě.

Protokol je založený na modelu klient–server, kdy serverem je zde SNMP manažer a klientem je SNMP agent viz obr. 7.1. Veškeré hodnoty poskytované agentem jsou jednoznačně definované pomocí unikátního číselného identifikátoru OID (*Object IDentifier*), které je tvořeno posloupností čísel oddělených tečkou. Celá stromová struktura OID identifikátorů je uložena v MIB databázi. MIB obsahuje jména a popisy jednotlivých identifikátorů [26] [27].



Obr. 7.1: Architektura SNMP

7.1 Princip činnosti

Na monitorované straně (agent) jsou shromažďovány informace o stavu zařízení, jako např. vytížení procesoru, datový provoz na lince, stav disků atd. Ty jsou poté v pravidelných intervalech odesílány jako odpovědi na požadavky od manažera. Získaný obsah zpráv se na straně monitorovací může dále zpracovávat do přehledných grafů a tabulek. Běžně je tedy komunikace zahajována dotazem ze strany manažera v časových intervalech, ale je také možné, aby agent na základě předdefinované události (výpadek některé linky, překročení mezních hodnot atd.) zasílal informace manažerovy. Takovéto konfiguraci agenta se říká SNMP TRAP.

Pro výměnu SNMP zpráv se využívá komunikace založená na UDP protokolu. Vzhledem k tomu, že UDP poskytuje nespojovou, nepotvrzovanou komunikaci, hodí se tedy do komunikace při velkém množství agentů a manažerů v síti, protože nezatěžuje síť tolik jako TCP.

SNMP agenti naslouchají na portu 161, manažeři využívají dynamického přiřazení portů, aby mohli současně komunikovat s více agenty. Zprávy TRAP se posílají na NMS, který tyto zprávy očekává na portu 162.

7.2 MIB databáze - Management Information Base

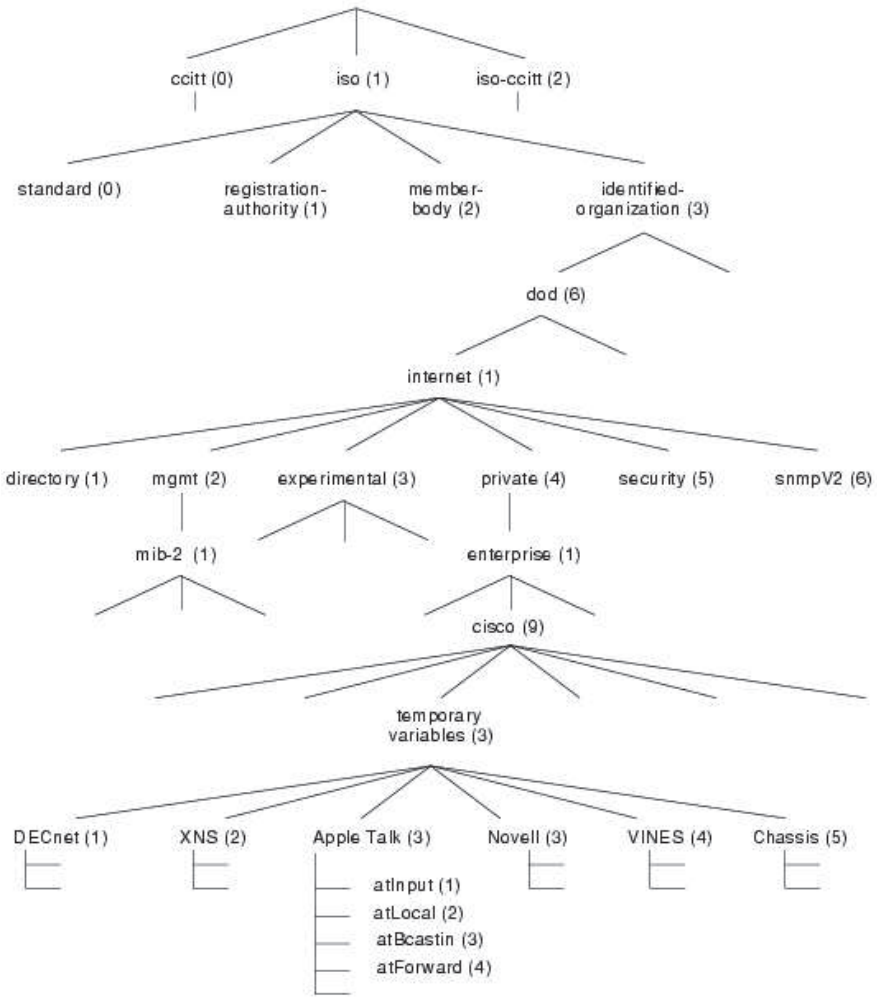
Každá hodnota v SNMP je jednoznačně identifikována pomocí číselného identifikátoru OID - Object Identifier. OID je tvořeno posloupností čísel oddělených tečkou, tato hodnota vznikne tak, že se vezme OID nadřazeného prvku a doplní se tečka a aktuální číslo. Celá tato stromová struktura je uložena v MIB databázi. Navíc MIB databáze obsahuje jména a popisy jednotlivých hodnot (OID) viz obr. 7.2 [27].

Příkladem OID může být třeba hodnota 1.3.6.1.4.1.9.3.3.1, které odpovídá textová verze z MIB databáze *iso.identified-organization.dod.internet.private.enterprise.cisco...*

7.3 SNMP operace

V protokolu SNMP jsou definovány tyto operace:

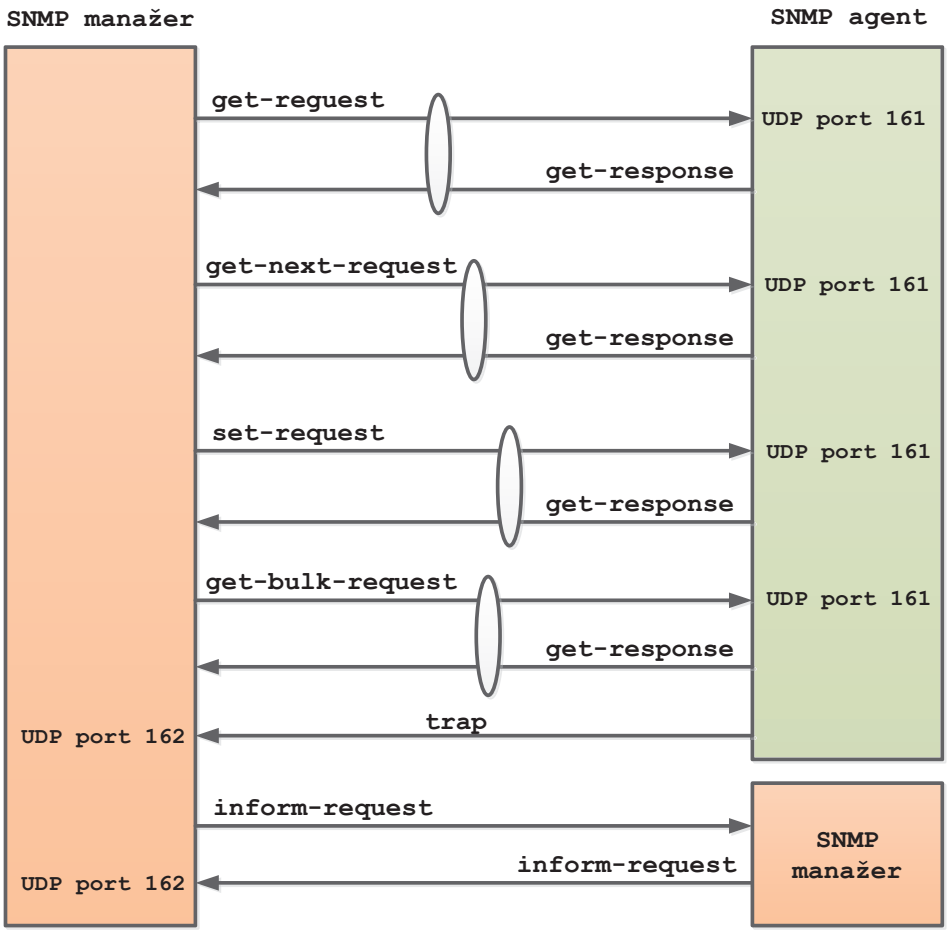
- **Get** – slouží k získání informace z MIB agenta.
- **GetNext** – umožňuje manažerovi získat informace o objektech v MIB bez znalosti jejich přesných jmen, umožňuje postupné procházení celým hierarchickým stromem.
- **GetResponse** – reakce agenta na zprávy typu Get, odpověď obsahuje i dotaz, protože protokol nezajišťuje souvislost mezi dotazem a odpovědí.



Obr. 7.2: Hierarchie MIB (Převzato z [30])

- **Set** – nabízí možnost manažerovi měnit nastavení jednotlivých zařízení a ne z nich jen číst data. Manažer je tak schopen pomocí zprávy SetRequest změnit některé hodnoty v agentově MIB tabulce.
- **Trap** – jediný typ příkazu vysílaný bez předchozího vyžádání, agent jej zasílá manažerovi jako reakci na specifikovanou událost (výpadek některé linky, přetížení atd.).
- **GetBulk** – Umožňuje vyžádat si k přečtení celou skupinu informací z MIB, čímž se urychluje komunikace.
- **Inform** – zpráva odeslána manažerem k jinému manažerovi za účelem výměny informací uložených v jejich MIB.
- **Report** – slouží opět ke komunikaci mezi manažery, nejčastěji je využívána k hlášení chyb při zpracování SNMP zpráv.

V první verzi SNMP protokolu, je definováno jen prvních pět operací, zbylé jsou podporovány až v novějších verzích SNMPv2 a SNMPv3 [28] [29].



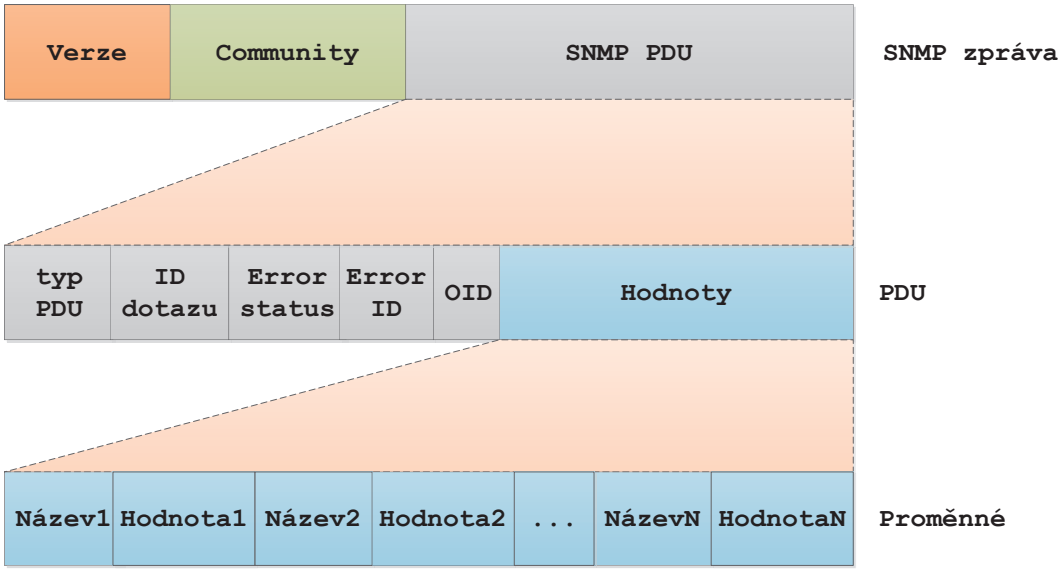
Obr. 7.3: SNMP operace

7.4 Formát zprávy SNMP

SNMP zpráva se skládá ze dvou částí - hlavička paketu a vlastní PDU (*Protocol Data Unit*) viz obr. 7.4. V hlavičce jsou uloženy informace o verzi SNMP a tzv. Community string, který slouží k zabezpečení. Vlastní datová část zprávy obsahuje jeden ze specifikovaných SNMP příkazů a příslušný operand (položku objektu, která je předmětem transakce). PDU obsahuje hodnoty [29]:

- Typ PDU – určuje, o jakou se jedná operaci (get, trap atd.)
- ID dotazu – označuje příslušné dvojice dotazu a odpovědi
- Error status – udává, zda byl požadavek úspěšný nebo skončil chybou

- Error ID - podrobnější informace o chybě a přiřazuje jí určitou hodnotu
- OID – identifikátor objektu
- Hodnota – konkrétní hodnota proměnné



Obr. 7.4: Formát SNMP zpráv

7.5 Verze SNMP

SNMPv1 – první verze protokolu specifikována v roce 1990. Při dotazování využívá pět základních typů operací (GetRequest, GetNextRequest, GetResponse, SetRequest a Trap), které jsou zapouzdřeny v datové jednotce PDU. Hlavní nevýhodou SNMPv1 bylo jeho nízké zabezpečení. Systém autorizace přístupu uživatelů byl na velmi nízké úrovni, hesla byla přenášena společně s daty nezabezpečenou cestou a bylo tak možné je zachytit.
SNMPv2 – Vychází z první verze protokolu, k základním operacím přidává další jako Get-Bulk, InformRequest, Report, Notification (blíže viz 2.3.4), které pomohly zvýšit efektivitu protokolu. Umožňuje v jednom paketu přenést více než jednu dotazovanou hodnotu. Do druhé verze protokolu byla zahrnuta funkce potvrzování na aplikační vrstvě, takže stanice již mohly rozpoznat, zda byla nebo nebyla zpráva doručena. Největší nevýhodou je nekompatibilita mezi SNMPv2 a SNMPv1.
SNMPv3 – Nejvýznamnější změnou oproti starším verzím je nový přístup k zajištění bezpečnosti. Zabezpečení v přenosu zprávy je aplikováno jako Access Control (ověření pomocí jména a hesla) a je začleněno do protokolových operací.

8 LABORATORNÍ ÚLOHY

Výstupem bakalářské práce, jsou laboratorní úlohy navržené pro jednotlivé serverové služby rozebírané v rámci této práce. Kompletní laboratorní úlohy včetně teoretického úvodu, postupem práce a doplňujících otázek se nachází jako příloha na CD dodaném s bakalářskou prací. V samotné práci se nachází v příloze A pouze „pracovní postup“ jednotlivých úloh. Byly navrženy následující laboratorní úlohy:

- Konfigurace FTP(S) serveru v systému Windows Server a Debian
- Konfigurace DHCP serveru v systému Windows Server a Debian
- Konfigurace VPN spojení v systému Windows Server a Debian
- Konfigurace NAT a firewallu v systému Windows Server a Debian
- Konfigurace SNMP v systému Windows Server a Debian

V této části bakalářské práce bude popsána náplň jednotlivých úloh, výběr použitého software a také přínos úloh při budoucím použití v laboratořích.

8.1 Výběr operačních systémů

V laboratorních úlohách jsou popsány realizace jednotlivých serverových služeb v operačním systému Windows Server 2008 R2 a linuxové distribuce Debian 6. Oba operační systému běží virtualizovaně pomocí multiplatformního nástroje *VirtualBox*. Změny provedené ve virtualizovaných systémech tedy neomezí běh „hostovaného“[1] operačního systému.

Ve virtualizačním nástroji byla nakonfigurována dvě síťová rozhraní, první slouží jako síťový most do školní sítě, druhé vytváří virtuální propojení virtalizovaných operačních systému. V operačním systému Windows Sever byly po instalaci provedeny následující změny, ve firewallu došlo k povolení odpovědí na příchozí *icmp* pakety, pro lepší přehlednost byla přejmenována síťová rozhraní na eth0 (síťový most) a eth1 (spojení mezi systémy) a bylo povoleno připojení ke vzdálené ploše, jinak nebyly oproti „čisté“ instalaci provedeny žádné dodatečné změny.

Použitý operační systém Debian 6.0.4 (kernel 2.6.32-5-686) byl instalovaný z tzv. businesscard[2] média. Při instalaci byly nainstalovány pouze základní součásti pro běh systému, vše ostatní bylo nutné doinstalovat ručně. Po instalaci byly doinstalovány součásti jako prostředí GNOME 2.30.2, gnome-terminál, podpora pro PPTP a OpenVPN spojení, internetový prohlížeč Chromium a další nástroje např. Putty, FileZilla, nmap, Wireshark atd. Stejně jako u Windows Serveru i zde byla přejmenována síťová rozhraní.

[1] Hlavní operační systém, nainstalovaný na fyzickém hardwaru.

[2] Médium obsahu pouze samotný instalátor, vše ostatní je nutné stáhnout z internetu, proto se velikost média pohybuje kolem 50 MB.

8.2 FTP

V laboratorní úloze se student seznámí s jedním z nejznámějších přenosových protokolů založený na architektuře klient–server, jde o protokol FTP. V teoretickém úvodu jsou uvedeny také možnosti zabezpečení tohoto protokolu.

V praktické části se nakonfiguruje nezabezpečený FTP server v operačním systému Windows Server 2008 R2 za pomocí vestavěných nástrojů v systému, konkrétně s použitím IIS7 (Internet Information Services 7). Paketovým analyzátorem se zachytí průběh navazování spojení a přenos souborů, tím se ověří, že přenášené informace skutečně putují v otevřené podobě a je tedy poměrně snadné tyto informace zneužít. Druhá část je zaměřena na konfiguraci FTP serveru s podporou šifrovaného přenosu dat v operačním systému Debian. K vytvoření FTP serveru byl použit zdarma dostupný software *proftpd* a jako podporu pro šifrované spojení byl použit nástroj *openssl*. Konfigurace probíhá formou úpravy konfiguračních souborů, nástrojem *openssl* se vygeneruje potřebný klíč a certifikát pro realizaci šifrovaného přenosu.

8.3 DHCP

Laboratorní úloha zaměřená na službu DHCP seznámí studenta s principem fungování DHCP protokolu a také jak funguje tento protokol v prostředí IPv6.

Dále s praktickou realizací DHCP serveru v operačním systému Windows Server za pomocí vestavěných služeb v operačním systému. Ověřování konfigurace bude probíhat z druhého operačního systému, kde se budou také zachytávat pakety při přidělování konfigurace z DHCP. V linuxové distribuci Debian se poté seznámí s automatickou konfigurací v protokolovém prostředí IPv6, za pomocí tzv. ohlášení směrovače (RA) a DHCPv6. Jako software zde byl použit nástroj *radvd* a *isc-dhcp*. RA démon bude nakonfigurovaný tak, aby poskytoval pouze informace o výchozí bráně a vše ostatní (jako IP, masku, DNS) bude nabízeno DHCPv6 serverem.

8.4 VPN

V úloze zaměřenou na VPN se student seznámí s obecnými vlastnostmi tunelování síťového provozu a nejpoužívanějších VPN řešení, včetně jejich výhod a nevýhod.

V praktické části poté nakonfiguruje v operačním systému Windows Server 2008 R2 PPTP server, jehož funkčnost ověří z druhého systému. PPTP byl zvolen kvůli jeho velkému rozšíření v tomto operačním systému. V systému Debian se bude realizovat VPN spojení založené na SSL/TLS šifrování, za pomocí nástroje *OpenVPN*.

Zde bude také nutné vytvořit infrastrukturu veřejného klíče pomocí skriptů obsažených v *OpenVPN*.

8.5 NAT

V úloze zaměřené na překlad síťových adres, se student dozví metody, jakým mohou být adresy překládány a také základní informace o firewallech a práci s nástrojem *iptables*.

V praktické části je nejprve nutné upravit adresování a síťová rozhraní, aby mohla být úloha na NAT vůbec realizovaná. Virtuální zapojení je nutné uspořádat tak, aby konfigurovaný operační systém sloužil jako brána do internetu. V operačním systému Windows server se poté nakonfiguruje překlad adres, včetně ukázky přesměrování konkrétního portu. Ověření bude probíhat ze dvou virtuálních operačních systému, jeden se bude nacházet před NATem (ve vnější síti) a druhý ve vnitřní. V systému Debian bude poté pomocí nástroje *iptables* nakonfigurována IP maškaráda (typ SNATu) a jednoduché nastavení firewallu.

8.6 SNMP

V této laboratorní úloze se student dozví, k čemu slouží protokol SNMP, jak pracuje, jaké typy operací podporuje a jak vypadá formát SNMP zpráv. V praktické části se poté seznámí, jak nakonfigurovat SNMP agenta v operačním systému Windows Server a jak nainstalovat a zprovoznit monitorovací nástroj *Cacti* v systému Debian. Úloha také obsahuje zachytávání paketů při vyměňovaní SNMP informací.

9 ZÁVĚR

V rámci této práce jsem se seznámil s principem činnosti serverových služeb typu DHCP, FTP, VPN, NAT a SNMP, jejich instalací a konfiguracích ve dvou různých operačních systémech. V práci jsem se také nově seznámil s protokolem IPv6 a jeho adresováním, novými vlastnostmi a výhodami oproti starší verzi protokolu.

Pro pochopení všech souvislostí nutných k porozumění a realizaci jednotlivých služeb jsem do první části zahrnul i teoretický popis TCP/IP protokolu, se zaměřením na dva nejpoužívanější protokoly síťové vrstvy IPv4 a IPv6.

V praktické části jsem všechny vybrané služby implementoval. Při implementaci služeb v operačním systému Windows Server byly použity vestavěné nástroje v systému, nebylo tedy nutné hledat programy třetích stran. V operačním systému Debian byla práce poněkud zajímavější, jelikož existuje celá řada programů pro realizaci.

Na základě získaných zkušeností jsem navrhl celkem 5 laboratorních úloh ve dvou operačních systémech. V laboratorní úloze zaměřené na přenosový protokol FTP se konfiguruje nezabezpečený i zabezpečený FTP server, aby bylo možné poukázat na nevhodnost použití nezabezpečené formy přenosu dat, ať už při přihlašování nebo přenosu samotných souborů. V další úloze jsem se zaměřil na konfiguraci DHCP serveru v protokolu IPv4 tak i IPv6, kvůli stále častějšímu pronikání tohoto „nového“ protokolu do běžného použití. Úloha se také zaměřuje na zachytávání paketů při DHCP komunikaci pro lepší pochopení principu fungování i formátu jednotlivých zpráv. Při návrhu laboratorní úlohy zaměřené na VPN spojení jsem v operačním systému Windows Server navrch konfiguraci PPTP serveru, ačkoliv nejde o nejbezpečnější protokol, ale je nejpoužívanějším typem VPN spojení v tomto systému. Druhou částí úlohy je vytvoření spojení založené na SSL/TLS šifrování. Při realizaci překladu síťových adres v systému Debian jsem zahrnul do návrhu i základní konfiguraci stavového firewallu. Největší problémy při návrhu úloh nastaly u SNMP protokolu, jelikož jsem nebyl schopen najít žádný zdarma dostupný software pro monitorování zařízení, který by splňoval moje požadavky.

LITERATURA

[1] *Windows Server 2008 R2 Editions* [online]. 2009 [cit. 2011-11-13]. Microsoft Server and Cloud Platform. Dostupné z: <http://www.microsoft.com/en-us/server-cloud/windows-server/2008-r2-editions.aspx>.

[2] STANEK, W. R. *Microsoft Windows Server 2008 – Kapesní rádce administrátora.* 1. vydání. Brno: Computer Press, a. s., 2008. 704 s. ISBN 978-80-251-1936-5.

[3] *DAQUAS* [online]. 2009 [cit. 2011-11-13]. Windows Server 2008 R2. Dostupné z: <http://www.daquas.cz/articles/463-windows-server-2008-r2>.

[4] KRČMÁŘ, Petr. *ROOT.CZ* [online]. 2010 [cit. 2011-11-13]. Historie operačního systému GNU/Linux. Dostupné z: <http://www.root.cz/texty/historie-operacniho-systemu-gnulinux/>.

[5] Debian. In *Wikipedia : the free encyclopedia* [online], poslední aktualizace 11. 9. 2011 [cit. 2011-11-13]. Dostupné z: <http://cs.wikipedia.org/wiki/Debian>.

[6] DOSTÁLEK, L., KABELOVÁ, A. *Velký průvodce protokoly TCP/IP a systémem DNS.* 5. aktualizované vydání. Brno: Computer Press, a. s., 2008. 488 s. ISBN 978-80-251-2236-5.

[7] PETERKA, Jiří. *eArchiv.cz* [online]. [cit. 2011-11-14]. Síťový model TCP/IP. Dostupné z: <http://www.earchiv.cz/a92/a231c110.php3>.

[8] PETERKA, Jiří. *eArchiv.cz* [online]. [cit. 2011-11-14]. Adresování v TCP/IP sítích - I. Dostupné z: <http://www.earchiv.cz/a92/a233c110.php3>.

[9] BOUŠKA, Petr. *SAMURAR-cz.com* [online]. 2010 [cit. 2011-11-15]. Adresování v IP sítích. Dostupné z: <http://www.samuraj-cz.com/clanek/adresovani-v-ip-sitich/>.

[10] SATRAPA, P. *IPv6* [online]. 3. vydání. Praha: CZ.NIC, z. s. p. o., 2011 [cit. 2011-04-13]. Dostupné z: <http://knihy.nic.cz/files/nic/edice/pavel_satrapa_ipv6_2011.pdf>. ISBN 978-80-904248-0-7.

[11] SCHRODER, C. *Linux – Kuchařka administrátora sítě.* 1. vydání. Brno: Computer Press, a. s., 2009. 608 s. ISBN 978-80-251-2407-9.

[12] ROUSEK, Petr. *Aktivní x Pasivní režim komunikace protokolu FTP* [online]. [cit. 2011-12-02]. Dostupné z: <http://www.peugeot-club.com/forum/files.php?pid=377013&aid=11397>.

[13] POUSEELE, Stefaan. *ISAserver.org* [online]. 2002 [cit. 2011-12-02]. How the FTP protocol Challenges Firewall Security. Dostupné z: <http://www.isaserver.org/articles/how_the_ftp_protocol_challenges_firewall_security.html>.

[14] *NurdleTech.com* [online]. [cit. 2011-12-02]. Securing FTP using SSH Dostupné z: <http://www.nurdletech.com/ftp.html>.

[15] DROMS, R., LEMON, T. *DHCP – Příručka administrátora.* 1. vydání. Brno: Computer Press, a. s., 2004. 490 s. ISBN 80-251-0130-4.

[16] BŘEZINA, Pavel. *pavka.shotzone.cz* [online]. [cit. 2011-11-14]. DHCP. Dostupné z: <http://pavka.shotzone.cz/dhcp/dhcp-uvod.html>.

[17] LUHOVÝ, Karel. Virtuální privátní sítě VPN. *Svět sítí* [online]. 2003 [cit. 2011-12-02]. Dostupné z: <http://www.svetsiti.cz/rubrika.asp?rid=17&tid=219>.

[18] PRŮCHA, Ondřej. Vše co jste chtěli vědět o VPN, ale báli jste se zeptat. *O VPN* [online]. 2005 [cit. 2011-12-02]. Dostupné z: <http://home.zcu.cz/~ondrous/index.php?vyber=0>.

[19] PUŽMANOVÁ, Rita. Virtuální privátní sítě pro vzdálený přístup. *DSL.cz: Telekomunikace* [online]. 2006 [cit. 2012-05-14]. Dostupné z: <http://www.dsl.cz/clanek/511-virtualni-privatni-site-pro-vzdaleny-pristup>.

[20] HABRMAN, Robert. Síťové protokoly (XXV. část): Protokol L2TP. *OWEBU.cz* [online]. 2008 [cit. 2012-05-18]. Dostupné z: <http://owebu.bloger.cz/PC-site/Sitove-protokoly-XXV-cast-Protokol-L2TP>.

[21] Mikulec, Martin. VPN - virtuální privátní sítě. *OWEBU.cz* [online]. 2009 [cit. 2012-05-08]. Dostupné z: <http://owebu.bloger.cz/stitek_?s=VPN+-+virtu%C3%A1ln%C3%AD+priv%C3%A1tn%C3%AD+s%C3%ADt%C4%9B>.

[22] Virtual Private Networking. *Technet.microsoft.com* [online]. 2001 [cit. 2012-05-18]. Dostupné z: <http://technet.microsoft.com/en-us/library/bb742566.aspx>.

[23] SATRAPA, Pavel. *LUPA.CZ* [online]. 2006 [cit. 2011-12-02]. NAT vesus NAP. Dostupné z: <http://www.lupa.cz/clanky/nat-vesus-nap/>.

[24] KRČMÁŘ, Petr. *ROOT.CZ* [online]. 2007 [cit. 2011-12-02]. Proč není NAT totéž co firewall. Dostupné z: <http://www.root.cz/clanky/proc-neni-nat-totez-co-firewall/>.

[25] TYSON, Jeff. How Network Address Translation Works. *HowStuffWorks.com* [online]. [cit. 2012-05-21]. Dostupné z: <http://computer.howstuffworks.com/nat.htm/printable>.

[26] ŠTĚPÁN, Jakub. Sledování sítě. *Archiv referátů FI MUNI* [online]. 2011 [cit. 2011-12-03]. Dostupné z: <http://www.fi.muni.cz/~kas/p090/referaty/2011-jaro/ut/snmp.html>.

[27] BOUŠKA, Petr. SNMP - Simple Network Management Protocol. *SAMURAR-cz.com* [online]. 2006 [cit. 2011-12-03]. Dostupné z: <http://www.samuraj-cz.com/clanek/snmp-simple-network-management-protocol/>.

[28] SNMP protokol a jeho využití. *HW.cz* [online]. 2003 [cit. 2012-05-22]. Dostupné z: <http://www.hw.cz/Produkty/ART957-SNMP-protokol-a-jeho-vyuziti.html>.

[29] KLAŠKA, Luboš. Vznik a principy SNMP. *Svetsiti.cz* [online]. 2000 [cit. 2012-05-22]. Dostupné z: <http://www.svetsiti.cz/clanek.asp?cid=Format-SNMP-zprav-1462000>.

[30] What is SNMP?. *Pulsewan.com* [online]. 2000 [cit. 2012-05-22]. Dostupné z: <http://www.pulsewan.com/data101/snmp_basics.htm>.

[31] *proftpd.org* [online]. [cit. 2011-12-11]. Configuration Directive List. Dostupné z: <http://www.proftpd.org/docs/directives/linked/by-name.html>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

DHCP Dynamická konfigurace klientů – Dynamic Host Configuration Protocol

DNAT Cílová překlad adres – Destination Network Address Translation

DNS Systém doménových jmen – Domain name system

DUID Identifikace v DHCPv6 – Demand Unique Identifier

FTP Protokol pro přenos souborů – File Transfer Protocol

GNOME Prostředí pracovní plochy – GNU Network Object Model Environment

GRE Tunelovací protokol – Generic Routing Encapsulation

IANA Autorita pro přidělování čísel na internetu – Internet Assigned Numbers Authority

IP Internetový protokol – Internet Protocol

L2TP Tunelovací protokol druhé vrstvy – Layer 2 Tunneling Protocol

MAC Identifikátor síťových karet – Media access control

NAT Překlad síťových adres – Network Address Translation

NTP Protokol pro synchronizaci vnitřních hodin – Network Time Protocol

PPTP Tunelovací protokol bod–bod – Point-to-Point Tunneling Protocol

SNAT Zdrojový překlad adres – Source Network Address Translation

SNMP Protokol pro monitorování zařízení – Simple Network Management Protocol

SOHO Segment malých a domácích sítí – Small Office/Home Office

SSL Vrstva bezpečných socketů – Secure Sockets Layer

TCP Jeden z internetových protokolů – Transmission Control Protocol

UDP Jeden z internetových protokolů – User Datagram Protocol

VPN Virtuální privátní sítě – Virtual private network

WINS Name server pro jména počítačů v síťi – Windows Internet Naming Service

SEZNAM PŘÍLOH

A REALIZACE LABORATORNÍCH ÚLOH

A.1 Konfigurace FTP(S) serveru v systému Windows Server a Debian

Cíl:

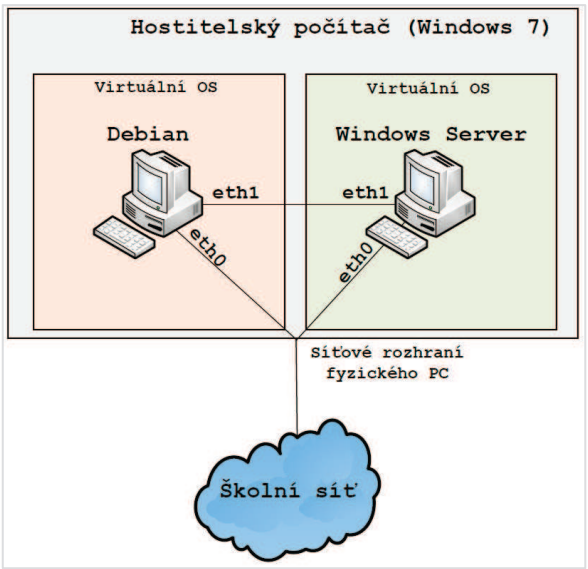
Seznámení se s přenosovým protokolem FTP a jeho možnostmi zabezpečení. Dále seznámení se s implementací protokolu v operačním systému Windows Server 2008 R2 a linuxové distribuce Debian.

Požadavky na vybavení pracoviště:

- PC se softwarem VirtualBox
- Virtualizovaný operační systém Microsoft Windows Server 2008 R2 SP1
- 2x Virtualizovaný operační systém Debian 6 (kernel 2.6.32-5-686)

Zapojení pracoviště

V každém virtuálním operačním systému jsou aktivována dvě síťová rozhraní, rozhraní **eth0** je přímo spojeno s učebnou (případně přes NAT) do tohoto rozhraní nebude nutné během úlohy zasahovat, obvykle bude sloužit jen jako „zdroj" internetu. Síťové rozhraní **eth1** slouží jako virtuální propojení běžících virtualizovaných operačních systému na jednotlivých stanicích. Rozhraní **eth1** má tedy planost pouze na konkrétním počítači a není možné se pomocí něho spojit z jiného počítače v učebně, na tomto rozhraní se budou tedy konfigurovat všechny parametry, aniž bychom zasáhli běh školní sítě.



Přihlašovací údaje

Windows Server 2008 R2 – účet: **Administrator**, heslo **Bars123456789**

Debian (klient i server) – účet: **bars**, heslo **bars** (učet bars je členem skupiny sudo, má tedy správcovská oprávnění)

Úkoly:

- Seznámit se protokolem FTP, s jeho režimy a s možnostmi zabezpečení
- Nakonfigurovat nezabezpečený FTP server v operačním systému Windows Server
- Nakonfigurovat zabezpečený FTPS server v prostředí Debian
- Ověřit funkčnost obou řešení pomocí FTP klienta
- Zachytit a rozpoznat přenášená data pomocí nástroje Wireshark

Teoretický úvod:

Viz kapitola 3.

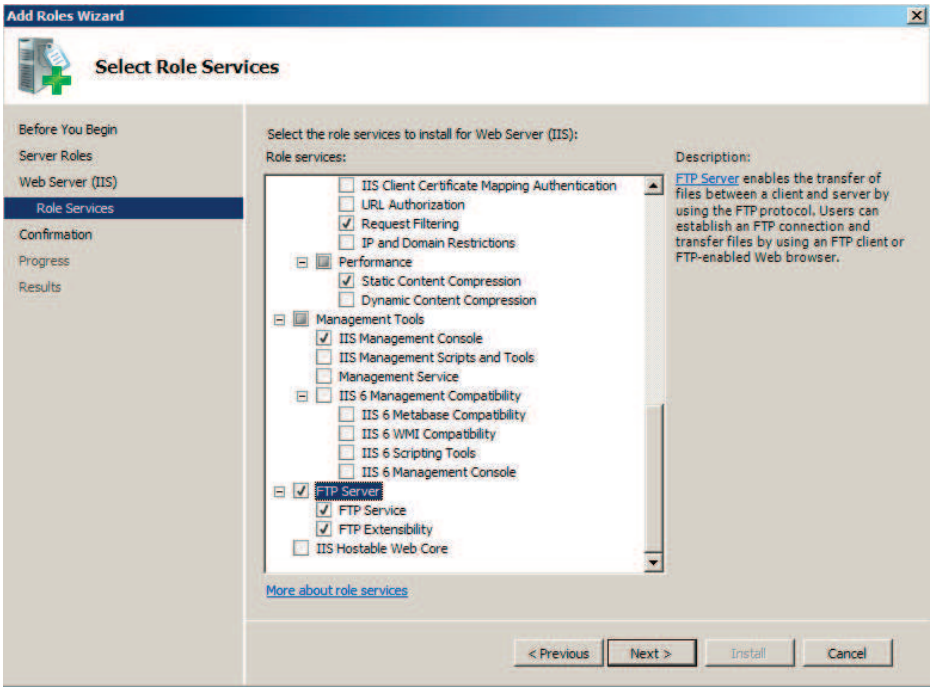
Postup práce:

Úkol č. 1:

V první části úlohy nakonfigurujeme jednoduchý FTP server v operačním systému Windows Server 2008 R2 SP1, pomocí nástroje IIS (Internet Information Services) který je součástí operačního systému. Ověření funkčnosti provedeme z druhého operačního systému Debian, FTP klientem FileZilla. Pomocí síťového analyzátoru Wireshark zachytíme průběh spojení, ze kterého budeme schopni zjistit obsah přenášeného souboru a přihlašovací údaje.

Instalace a konfigurace

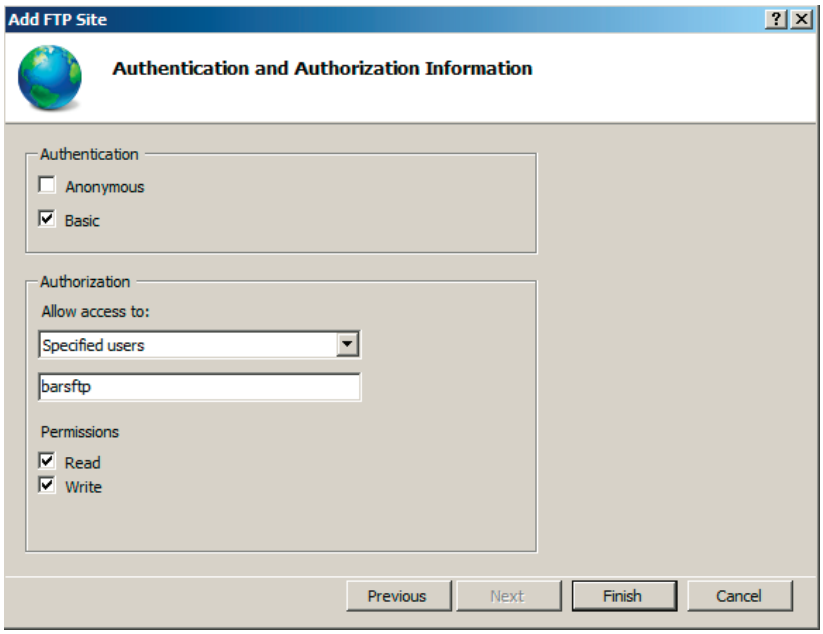
- Spusťte virtualizovaný operační systém Windows Server 2008 R2 SP1 a přihlaste se k němu (Heslo pro Administrátorský účet je **Bars123456789**).
- Před samotnou konfigurací FTP serveru je nejprve nutné vytvořit uživatelský účet, případně skupinu uživatelů, kteří se budou moci připojovat na FTP server. Otevřete tedy nástroj **Server Manager** (např. přes nabídku Start -> Administrative Tools -> Server Manager). V levém panelu rozklikněte nabídku *Configuration* a dále položku *Local Users and Groups***.** Zde vytvořte pod položkou *Users* nového uživatele nazvaného jako **barsftp** s heslem **Password123456** (heslo musí kvůli nastavení bezpečnostních politik splňovat určitou složitost) a **odškrtněte** položku *User must change password at next logon*, jinak nebude možné využít účet pro připojení.
- Zůstaňte v nástroji *Server Manager* a přidejte novou roli serveru. V levém panelu položka *Roles* a následně *Add Roles*. Zde vyberte roli **Web Server (IIS)** a pokračujte dále až k výběru jednotlivých služeb. Zde vyberte na konci seznamu položku **FTP server** včetně jeho služeb viz obr. 6. Pokračujte dále a dokončete instalaci tlačítkem *Install*.



obr. 6 Služby Web serveru

- Vše potřebné máme nainstalováno, vrhneme se tedy na samotnou konfiguraci FTP serveru. Otevřete nástroj **Internet Information Services (IIS) Manager** (Start -> Administrative Tools), rozklikněte náš server a přejděte na položku *Sites*. Zde v pravém menu vyberte položku *Add FTP site***,** tím pustíte průvodce pro vytvoření našeho FTP. Vyplňte název a vyberte složku, která bude sloužit jako FTP úložiště, např. **C:\FTP** (složku musíte vytvořit). Na další straně můžete vybrat, pro jaké rozhraní bude FTP fungovat a na jakém portu, nechejte výchozí nastavení, jako typ SSL

šifrování zvolte **No SSL**. V dalším kroku, jako typ ověřování zvolte **Basic** a v sekci autorizace vyberte možnost **Specified users** a vyplňte název vytvořeného účtu pro přístup na FTP server a povolte mu zápis a čtení viz obr. 7 a dokončete konfiguraci.



obr. 7 Průvodce přidání nového FTP

- Po instalaci se do firewallu automaticky přidají pravidla pro příchozí komunikaci na portu 21 a také se vytvoří pravidlo povolující rozsah portů nutný pro běh pasivního spojení FTP protokolu. Nicméně je ještě nutné přidat do příchozích pravidel pravidlo povolující spojení pro program **svchost.exe**, který je součástí Windows. Otevřete ovládací panely (Start -> Control Panel), zvolte kategorii **System and Security** a dále **Allow a program through Windows Firewall**. Zde přidejte nový záznam tlačítkem **Allow another program** a vyberte program **svchost.exe** (cesta k souboru je C:\Windows\System32) a přidejte ho do seznamu výjimek pro příchozí spojení. Tím je konfigurace FTP serveru kompletní.

Ověření konfigurace a zachytávání paketů

- Ve složce určené pro FTP, vytvořte obyčejný textový soubor s jednoduchým obsahem (vhodné volit text bez diakritiky), který budeme později zachytávat.
- Spusťte operační systém Debian (klient) a přihlaste se pod uživatelským jménem **bars** s heslem **bars**. Uživatelský účet *bars*, je členem skupiny sudo, je to tedy účet se zvýšenými oprávněními, která budeme při některých krocích potřebovat.
- Po přihlášení spusťte FTP klienta FileZilla (v terminálovém okně napište filezilla, případně přes ikonu na panelu), v hlavním okně vyplňte položky **Hostitel** (IP adresa FTP serveru), **Uživatelské jméno** a **heslo** a případně port (nevyplněno znamená výchozí port 21) a připojte se.
- Pokud se úspěšně připojíte, zapněte síťový analyzátor Wireshark přes konzoli, příkazem:

```
sudo wireshark
```

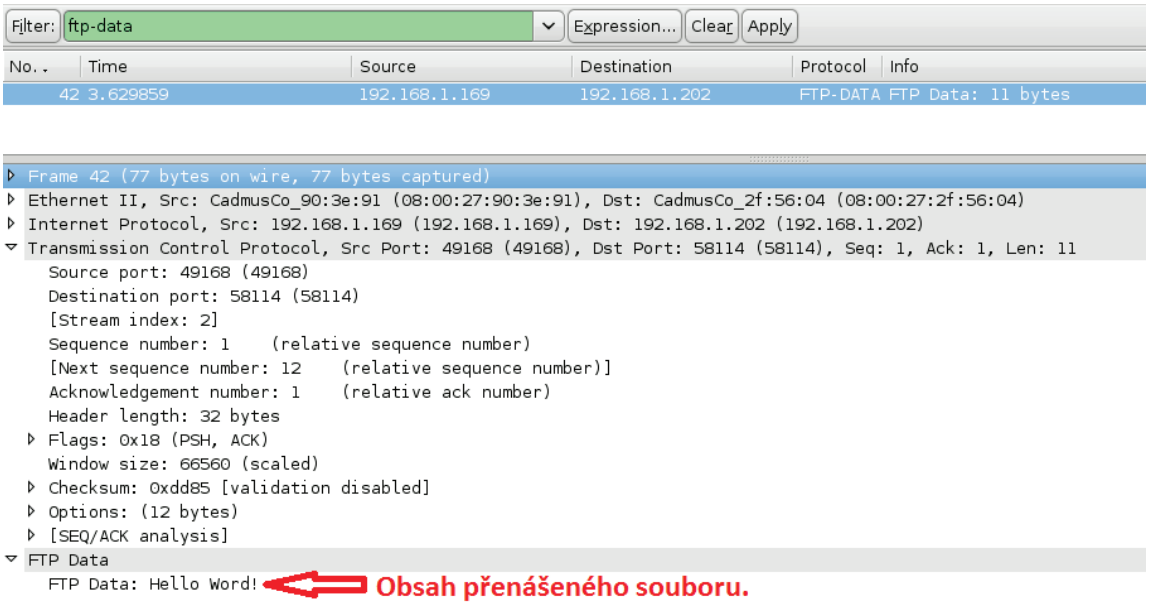
  nástroj je nutné spustit se zvýšenými právy.
- Spusťte zachytávání paketů (Capture -> Interface) pro ethernetové rozhraní **eth1.**
- Nyní se přepněte do nástroje FileZilla a opět se připojte na server a přeneste vámi vytvořený textový soubor do domovského adresáře **/home/bars** a ukončete spojení.
- Vraťte se zpět do programu Wireshark a zastavte zachytávání paketů (Capture -> Stop). Nyní přestoupíme k samotné analýze zachycených paketů.
- Díky filtrům, které aplikace umožňuje, se dá zachycená komunikace poměrně dost zpřehlednit.

Např. filtr nazvaný **ftp –** zobrazí všechny pakety FTP spojení

ftp-data – zobrazí pouze pakety přenesených souborů viz obr. 8

ftp contains "USER" – pakety při kterých se vyjednává přihlašovací jméno

Podívejte se, jaké informace je možné zachytit při použití „běžného“ nešifrovaného přenosu souborů pomocí protokolu FTP.



obr. 8 Zachycená data z nástroje Wireshark

Samostatná práce:

1. Pro vytvořený FTP server, definujte pravidlo povolující přístup pouze z jedné konkrétní IP adresy (Debian Klient).

Úkol č. 2:

V druhém úkolu vytvoříme FTP server s podporou šifrovaného přenosu v operačním systému Debian pomocí zdarma dostupného software *ProFTPd*. Certifikáty a šifrovací klíče budeme generovat nástrojem *openssl.* Ověření funkčnosti se opět provede pomocí nástroje FileZilla a také ověříme pomocí analyzátoru Wireshark, zda je komunikace skutečně šifrovaná.

Instalace a konfigurace

- Spusťte operační systém Debian (server) a přihlaste se (User **bars**, heslo **bars**).
- Po přihlášení se automaticky spustí Terminálové okno, pokud se tak nestane, zapněte jej ručně. Téměř celá úloha bude probíhat právě v terminálu zadáváním příkazů a editací konfiguračních souborů.
- Nainstalujte podpůrný software *ProFTPd* a *openssl* přes terminál:

```
sudo aptitude update
sudo aptitude install proftpd openssl
```

  po zadání příkazu budete dotázání na heslo, vyplňte jej a potvrďte instalaci.
- Při instalaci ProFTPd budete dotázání, jakým způsobem bude program spouštěn, zvolte možnost **Samostatně**.
- Během instalace se vytvoří uživatelský účet **ftp** s domovským adresářem */home/ftp*, který použijeme pro přihlašování na FTP server (účet ftp, má automaticky nastavený shell na /bin/false, není tedy možné se tímto účtem běžně přihlásit do systému).
- Účet *ftp* nemá přidělená žádné heslo, vytvořte tedy pro tento účet nějaké heslo.

```
sudo passwd ftp
```

- Nyní se vrhneme na samotnou konfiguraci FTP. Konfigurace se provádí editací konfiguračního souboru.

```
sudo gedit /etc/proftpd/proftpd.conf
```

  Soubor již obsahuje základní konfiguraci, nicméně některé poměrně důležité položky jsou zakomentovány, případně chybí.
- V souboru „odkomentujte“ (smažte znak #) u následujících položek:
  - **RequireValidShell off**
  - **DefaultRoot ~**

A dále přidejte řádky:

```
RootLogin  off
MaxLoginAttempts5
UserAlias pepik ftp
AllowStoreRestart    on
```

Význam a funkce jednotlivých řádků v konfiguračním souboru, jsou vysvětleny na konci laboratorní úlohy, případně na http://www.proftpd.org/docs/directives/linked/by-name.html .

- Po provedení všech úprav konfigurační soubor uložte a restartujte FTP server.

```
sudo /etc/init.d/proftpd restart
```

- Funkčnost této konfigurace ověřte z druhého systému nástrojem FileZilla, obdobně jako v Úkolu č. 1.

- Pokud vše proběhlo bez chyb, přejdeme k implementaci šifrování pro náš FTP server. Vygenerujte certifikát a šifrovací klíč příkazem:

```
sudo openssl req -new -x509 -days 365 -nodes
     -out /etc/proftpd/server.crt.pem
     -keyout /etc/proftpd/server.key.pem
```

Význam jednotlivých parametrů programu a podrobný popis je možné nalézt na http://www.openssl.org/docs/apps/openssl.html .

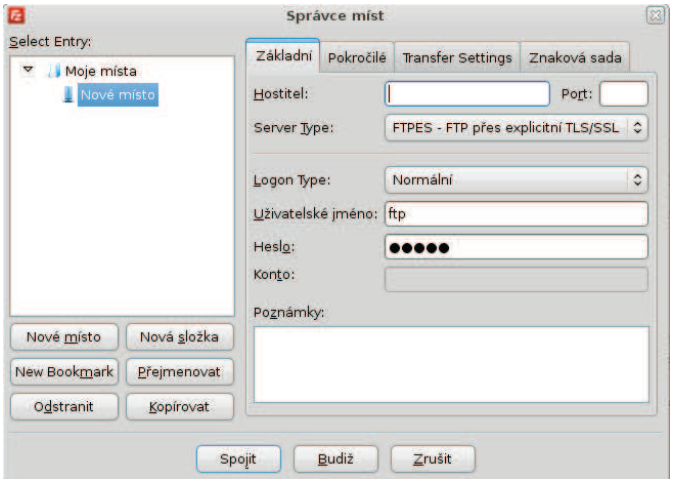
- Dále je nutné přidat následující řádky do konfiguračního souboru *proftpd.conf*, které povolují a dále upravují funkci šifrování přenosu dat.

```
<IfModule mod_tls.c>
  TLSEngine on
  TLSLog /var/log/proftpd-tls.log
  TLSProtocol TLSv1
  TLSRequired off
  TLSRSACertificateFile /etc/proftpd/server.crt.pem
  TLSRSACertificateKeyFile /etc/proftpd/server.key.pem
  TLSVerifyClient off
  TLSRenegotiate required off
  TLSOptions NoSessionReuseRequired
</IfModule>
```

- Po dokončení úpravy konfiguračního souboru, restartujte FTP server pro aplikaci provedených změn v konfiguraci.

Ověření konfigurace a zachytávání paketů

- V druhém operačním systému Debian spusťte FTP klient FileZilla a přidejte nové připojení přes **Správce míst** (Soubor -> Správce míst), vyplňte IP adresu FTP serveru**,** u položky Server Type, zvolte možnost **FTPES** a vyplňte přihlašovací údaje pro přístup na FTP viz obr. 9 a klikněte na tlačítko Spojit.
- Pokud připojení k serveru proběhne úspěšně, můžete se od něj odpojit.
- Zapněte paketový analyzátor Wireshark a spusťte zachytávání paketů na příslušném rozhraní a proveďte stejný proces jako při zachytávání paketů v prvním úkolu. (Připojit na FTP, přenést soubor, odpojit) Podařilo se vám i tentokrát zachytit přihlašovací jméno a heslo, případně obsah souboru?



obr. 9 Správce míst v nástroji FileZilla

Samostatná práce:

1. Upravte konfigurační soubor *proftpd.conf* tak, aby bylo možné přihlašovat se k FTP pouze a jedině přes účet *ftp*. (**Nápověda:** hledejte v sekci LIMIT v manuálových stránkách programu na http://www.proftpd.org/docs/howto/)
2. V konfiguračním souboru vytvořte alias s názvem *Franta* pro účet *ftp*. Přes vytvořený alias se zkuste přihlásit k FTP.

Kontrolní otázky:

1. K čemu slouží protokol FTP.
2. Na kterých portech FTP pracuje a jaké je jejich využití.
3. Jakým způsobem jsou zabezpečená data u základního protokolu FTP.
4. Hlavní rozdíl mezi aktivním a pasivním režimem FTP.
5. Jaké znáte možnosti zabezpečení přenosu pomocí protokolu FTP.
6. Jaký je rozdíl v protokolu TFTP oproti protokolu FTP?

Popis položek v souboru proftpd.conf

RequireValidShell	Požadavek na platný shell uvedený v souboru /etc/shells
DefaultRoot	Adresář, ve kterém bude uživatel „zamčen“ po připojení
RootLogin	Povolí/zakáže přihlašování přes uživatele root
MaxLoginAttempts	Nastavuje počet pokusů při zadávání chybného hesla, než nastane odpojení
UserAlias pepik ftp	Vytvoří „přezdívku“ pro účet ftp
AllowStoreRestart	Umožňuje znovunavázání při přerušeném nahrávání souboru

Literatura:

[1] POUSEELE, Stefaan. ISAserver.org [online]. 2002 [cit. 2011-12-02]. How the FTP protocol Challenges Firewall Security. Dostupné z WWW: <http://www.isaserver.org/articles/how_the_ftp_protocol_challenges_firewall_security.html>.

[2] ROUSEK, Petr. Aktivní x Pasivní režim komunikace protokolu FTP [online]. [cit. 2011-12-02]. Dostupné z WWW: <http://www.peugeot-club.com/forum/files.php?pid=377013&aid=11397>.

[3] NurdleTech.com [online]. [cit. 2011-12-02]. Securing FTP using SSH Dostupné Z WWW: <http://www.nurdletech.com/ftp.html>.

Manuály k použitým nástrojům

http://www.proftpd.org/docs/howto/
http://www.proftpd.org/docs/directives/linked/by-name.html
http://www.openssl.org/docs/HOWTO/
http://wiki.wireshark.org/
http://wiki.filezilla-project.org/Using

A.2 Konfigurace DHCP serveru v systému Windows Server a Debian

Cíl:

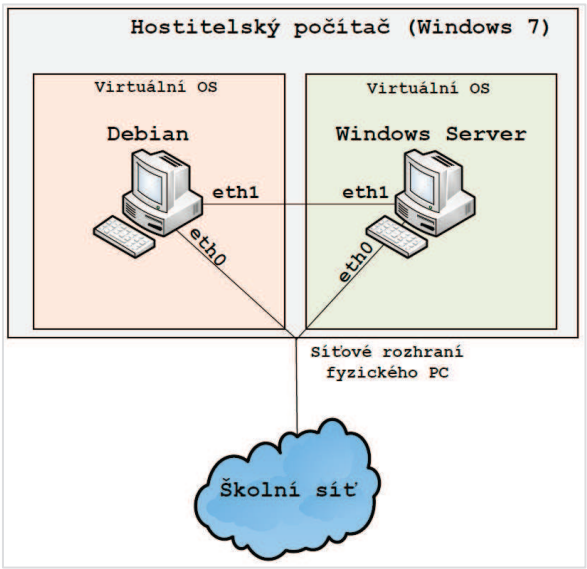
Cílem laboratorní úlohy je seznámit se s protokolem DHCP v protokolovém prostředí IPv4 a problémem automatické konfigurace v prostředí IPv6 a jejich konfiguraci v operačním systému Windows Server a linuxové distribuci Debian.

Požadavky na vybavení pracoviště:

- PC se softwarem VirtualBox
- Virtualizovaný operační systém Microsoft Windows Server 2008 R2 SP1
- 2x Virtualizovaný operační systém Debian 6 (kernel 2.6.32-5-686)

Zapojení pracoviště

V každém virtuálním operačním systému jsou aktivována dvě síťová rozhraní, rozhraní **eth0** je přímo spojeno s učebnou (případně přes NAT) do tohoto rozhraní nebude nutné během úlohy zasahovat, obvykle bude sloužit jen jako „zdroj“ internetu. Síťové rozhraní **eth1** slouží jako virtuální propojení běžících virtualizovaných operačních systému na jednotlivých stanicích. Rozhraní **eth1** má tedy planost pouze na konkrétním počítači a není možné se pomocí něho spojit z jiného počítače v učebně, na tomto rozhraní se budou tedy konfigurovat všechny parametry, aniž bychom zasáhli běh školní sítě.



Přihlašovací údaje

Windows Server 2008 R2 – účet: **Administrator**, heslo **Bars123456789**

Debian (klient i server) – účet: **bars**, heslo **bars** (učet bars je členem skupiny sudo, má tedy správcovská oprávnění)

Úkoly:

- Seznámit se s možnostmi automatické konfigurace v prostředí IPv4 a IPv6
- Nakonfigurovat DHCP server v operačním systému Windows Server
- Implementovat automatickou konfiguraci v IPv6 v operačním systému Debian
- Zachytit výměny zpráv DHCP a ověřit platnost poznatků z Teoretického úvodu

Teoretický úvod:

Viz kapitola 4.

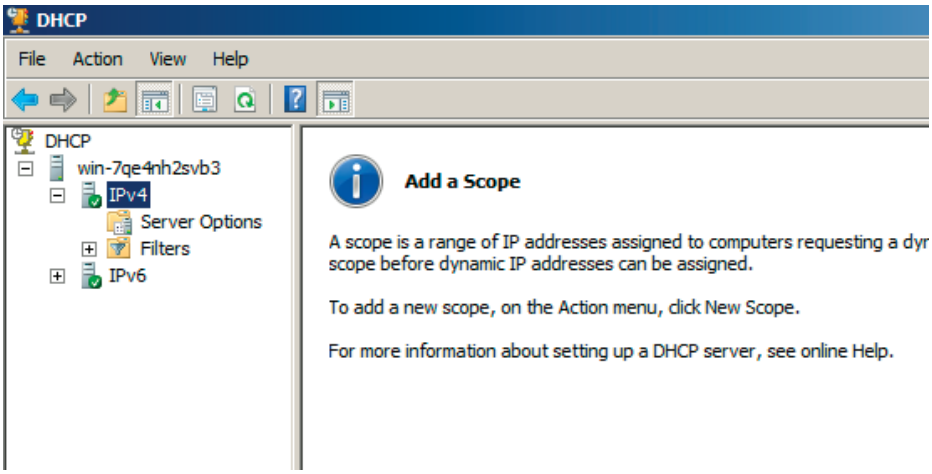
Postup práce:

Úkol č. 1:

V první části laboratorní úlohy se pokusíme nakonfigurovat DHCP server v operačním systému Windows Server 2008 R2 SP1. Dále se seznámíme s vytvářením filtrů a rezervací IP adres pro konkrétní MAC adresu. Ověření funkčnosti konfigurace provedeme v operačním systému Debian, ve kterém budeme také zachytávat pakety nástrojem Wireshark pro potvrzení znalostí z teoretického úvodu.

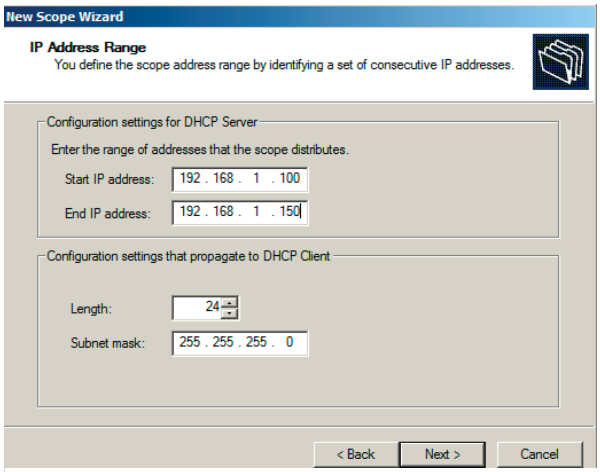
Instalace a konfigurace

- Spusťte operační systém Windows Server 2008 R2 SP1 a přihlaste se.
- Dříve než se pustíme do samotné instalace a konfigurace DHCP serveru, je nejprve nutné pro námi zvolené síťové rozhraní (**eth1**) definovat statickou IP adresu a masku, aby mohlo DHCP korektně fungovat. Otevřete Centrum síťových připojení a sdílení (*Network and Sharing Center*), např. přes ikonu na hlavním panelu nebo přes Ovládací panely. Zde klikněte na síťové rozhraní **eth1** a zvolte *Properties* a dále otevřete **Internet Protocol Version 4** a vyplňte IP adresu a masku (**192.168.1.1/24**) a konfiguraci potvrďte.
- Nyní již přejdeme k samotnému DHCP, spusťte nástroj **Server Manager** (např. přes nabídku Start -> Administrative Tools -> Server Manager) a přidejte novou roli serveru. V levém panelu položka *Roles* a následně *Add Roles*. Zde vyberte roli **DHCP Server** a pokračujte dále. Na dalším okně vybereme naše síťové rozhraní s IP adresou 192.168.1.1, dále vyplňte název *Parent domain* (např. bars.vutbr.cz) a pokračujte beze změny až na nabídku *DHCPv6 Stateless Mode*, kde zvolte *Disable DHCPv6 stateless…* (DHCPv6 není náplní této části úlohy). V okně *Confirmation* můžeme zkontrolovat naše nastavení a službu nainstalujte.
- Po instalaci můžete nástroj *Server Manager* zavřít a otevřete konzoli pro samotnou konfiguraci **DHCP** (Start -> Administrative Tools -> DHCP). „Rozklikněte“ náš server a označte protokol IPv4 a přes nabídku *Action* definujte nový rozsah (*New Scope*).

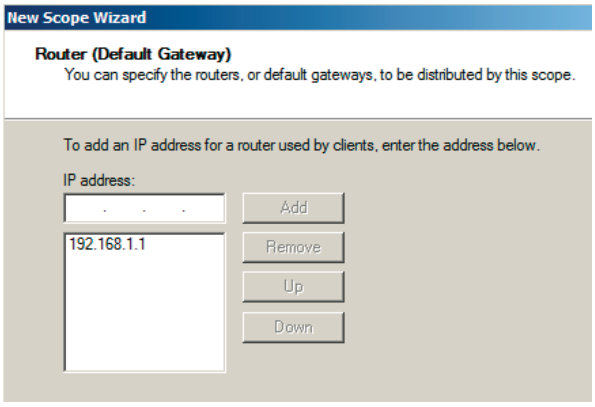


obr. 3 Konfigurační konzole DHCP

- Rozsah pojmenujte a na další stránce vyplňte rozsah, z jakého budou přidělovány IP adresy a masku sítě (192.168.1.100-192.168.1.150 s maskou 255.255.255.0) viz obr. 4.
- Na další stránce je možné definovat rozsah IP adres, které budou „vyloučené“ při přidělování, tuto nabídku nechejte nevyplněnou. Stejně tak i další stránku pro nastavení doby přidělení konfigurace nechte výchozí.
- V dalším kroku zvolte, že chcete konfigurovat další volby DHCP serveru. A přidejte IP adresu výchozí brány, která bude stejná jako IP serveru obr. 5.
- Další stránky pro konfiguraci DNS serverů a WINS serverů necháme ve výchozím nastavení.
- V posledním kroku zvolte, že chcete aktivovat konfigurovaný rozsah. Tím je základní nastavení DHCP serveru hotovo, nyní přistoupíme k ověření funkčnosti konfigurace.



obr. 4 Konfigurace rozsahu IP adres



obr. 5 Přidání výchozí brány

Ověření konfigurace a zachytávání paketů

- Spusťte operační systém Debian (klient) a přihlaste se. Zkontrolujte, zda je rozhraní **eth1** aktivované.
- Spusťte Terminál a ověřte, že síťové rozhraní **eth1** obdrželo IP adresu z nakonfigurovaného rozsahu příkazem:


```
sudo ifconfig
```

po zadání příkazu budete dotázání na heslo.

- Pokud máme přiřazenou IP adresu z vytvořeného DHCP serveru, přejdeme k zachytávání DHCP paketů pro ověření znalostí z teoretického úvodu.
- Zapněte paketový analyzátor Wireshark příkazem:

```
sudo wireshark &
```

analyzátor je nutné spustit se zvýšeným oprávněním. A spusťte zachytávání paketů pro rozhraní **eth1**.

- Nyní nejprve uvolníme přidělenou konfiguraci příkazem

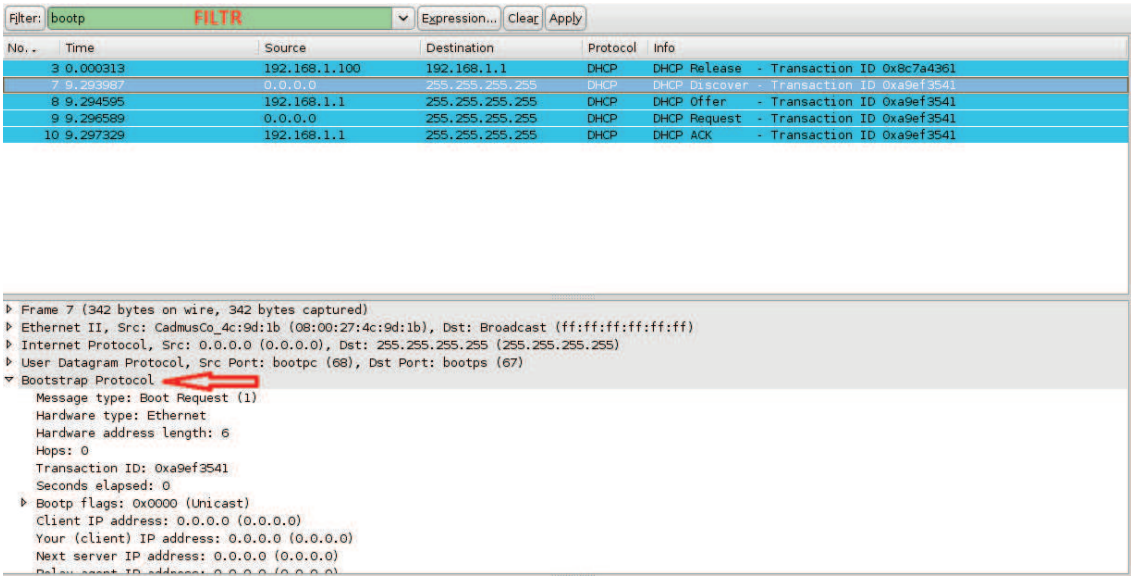
```
sudo dhclient -r
```

a následně zažádáme DHCP server o novou konfiguraci příkazem

```
sudo dhclient eth1
```

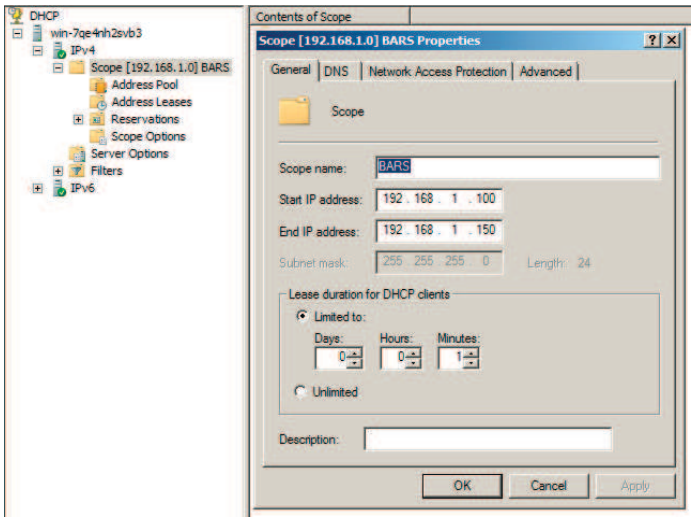
Po obnovení IP adresy se vraťte zpět do programu Wireshark a zastavte zachytávání paketů (Capture -> Stop). Nyní přestoupíme k samotné analýze zachycených paketů.

- Pro zobrazení pouze DHCP paketů slouží filtr s názvem **bootp**, aplikujte jej a prohlédněte si jaké typy DHCP paketů byly přenášeny a také se podívejte, které informace přenáší jednotlivé pakety, převážně se zaměřte na sekci **Bootstrap Protocol**.



obr. 6 Zachycená komunikace při obnovení IP adresy

- Dále změňte v nastavení DHCP serveru čas, po který bude přidělaná konfigurace platná (Lease Time). V konzoli DHCP klikněte pravým tlačítkem myši na vytvořený rozsah (Scope) a zvolte **Properties**. Zde nastavte čas na 1 minutu viz obr. 7.
- Zapněte zachytávání paketů, obnovte konfiguraci klienta podle příkazů výše a zjistěte, které DHCP pakety se vyměňují při znovu konfiguraci klienta po vypršení doby přidělení.



obr. 7 Změna doby platnosti konfigurace

Úkol č. 2:

V operačním systému Debian nakonfigurujeme DHCP server v protokolovém prostředí IPv6. Využijeme tzv. bezstavové DHCPv6, tj. IP adresu a informace o DNS bude klient přebírat od DHCPv6 a informace o výchozí bráně od Router Advertisement (RA). Jako DHCP server použijeme software od firmy ISC a router advertisement démona.

- Spusťte operační systém Debian (server) a přihlaste se.
- Stejně jako v první úkolu, nejprve přiřaďte rozhraní **eth1** statickou IPv6 adresu (fdfe::1 s maskou 64). To můžete udělat přes grafické rozhraní pro konfiguraci sítě (NetworkManager) nebo přes Terminál příkazem:

```
sudo ip addr add fdfe::1/64 dev eth1
```

a rozhraní zapněte

```
sudo ip link set eth1 up
```

- Nainstalujte potřebný software:

```
sudo aptitude update
sudo aptitude install isc-dhcp-server radvd
```

- Nyní již přejdeme k samotné konfiguraci, nejprve nakonfigurujte Router Advertisement démona. Vytvořte konfigurační soubor *radvd.conf*

```
sudo gedit /etc/radvd.conf
```

a vložte do něj následující obsah a soubor uložte:

```
##/etc/radvd.conf
interface eth1
{
   AdvSendAdvert on;
   AdvManagedFlag on;
   AdvOtherConfigFlag on;
   prefix fdfe::/64 {
      AdvOnLink on;
      AdvAutonomous off;
     };
};
```

Vysvětlení konfiguračního souboru: Položka `interface` udává, pro které rozhraní bude RA posíláno. `AdvSendAdvert` povoluje posílání periodických RA zpráv, `AdvManagedFlag` (příznak M) zapíná stavovou konfiguraci z DHCPv6, `AdvOtherConfigFlag` (příznak O) povolení dodatečných informací ze stavové konfigurace jako DNS, WINS atd., `AdvOnLink` povoluje použití daného prefixu, `AdvAutonomous` zakáže automatickou konfiguraci pomocí RA.

- Před samotným spuštěním RA démona je nutné v systému povolit předávání IPv6 paketů, to se provede příkazem:

```
sudo sysctl -w net.ipv6.conf.all.forwarding=1
```

- Nyní se již může RA spustit

```
sudo /etc/init.d/radvd start
```

- Další kroky budou věnovány konfiguraci DHCPv6. Po nainstalování ISC serveru se automaticky vytvoří soubor */etc/dhcp/dhcpd.conf*, který obsahuje vzorové nastavení pro konfiguraci DHCPv4, pro naše účely je tedy jeho obsah nepodstatný. Celý jeho obsah vymažte a nahraďte vlastní konfigurací určenou pro prostředí IPv6:

```
##/etc/dhcp/dhcpd.conf
default-lease-time 600;
max-lease-time 7200;

# Podsíť, kde se server nachazi
subnet6 fdfe::/64 {
      range6 fdfe::2 fdfe::255;
      option dhcp6.name-servers fec0:0:0:1::1;
      option dhcp6.domain-search "bars.vutbr.cz";
}
```

Vysvětlení konfiguračního souboru: Položka `default-lease-time` udává dobu, po kterou je přidělená konfigurace rezervovaná pro klienta, `max-lease-time` udává maximální dobu pronájmu konfigurace. `subnet6` definuje síť, pro kterou bude DHCPv6 přidělovat konfiguraci, `range6` je rozsah adres spravovaných serverem. A poslední dvě položky v souboru určují adresy DNS serverů a doménové jméno dané sítě.

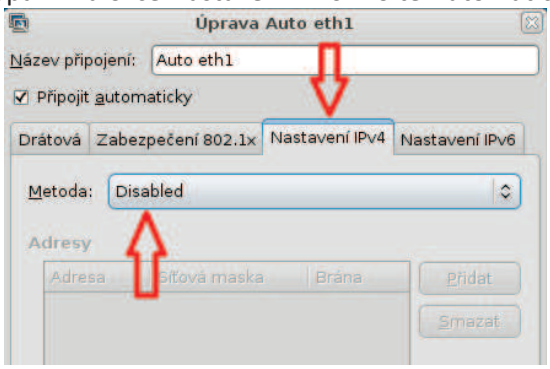
- Před spuštěním DHCP je ještě nutné vytvořit prázdný soubor, kdo kterého se budou ukládat informace o přidělených IP adresách, cesta k souboru je *„/var/lib/dhcp/dhcpd6.leases“*. Nyní můžete spustit samotnou službu DHCP serveru, která bude naslouchat na rozhraní **eth1.**

```
sudo dhcpd -6 eth1
```

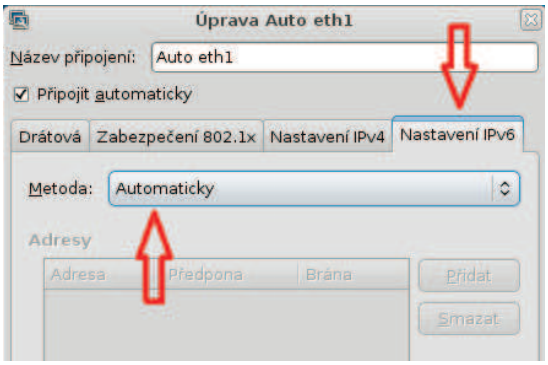
- Po spuštění DHCP serveru zapněte paketový analyzátor Wireshark (viz Úkol č. 1) a spusťte zachytávání pro rozhraní **eth1**.

Ověření konfigurace a zachytávání paketů

- Spusťte operační systém Debian (klient) a přes grafickou konzoli **Připojení k síti** (Systém -> Volby -> Připojení k síti) zakažte pro síťové rozhraní **eth1** přebírání konfigurace u protokolu IPv4 a naopak v záložce Nastavení IPv6 zvolte Automatickou metodu.



obr. 8 Konfigurace pro IPv4

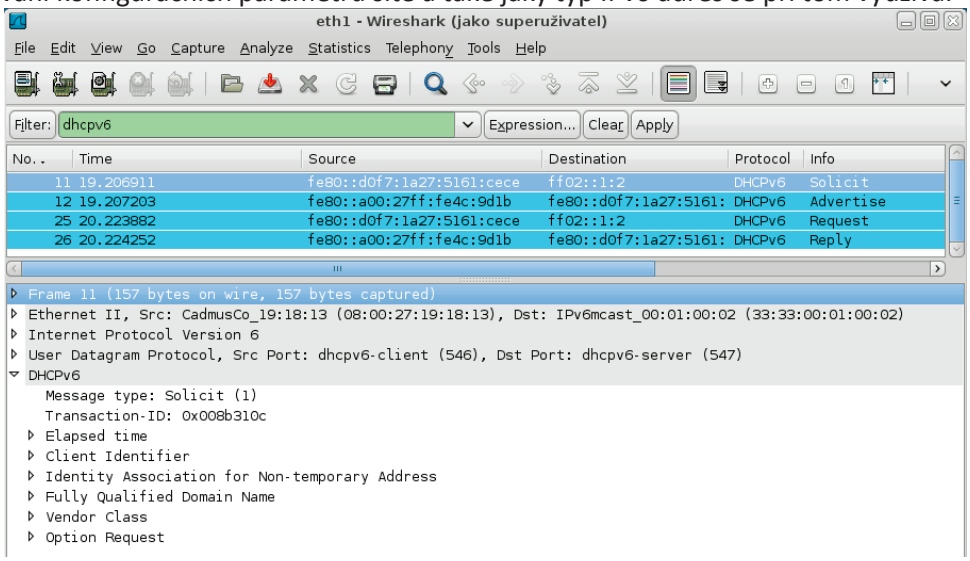


obr. 9 Konfigurace pro IPv6

- Spusťte Terminál a ověřte, že síťové rozhraní **eth1** obdrželo IPv6 adresu z nakonfigurovaného rozsahu příkazem:

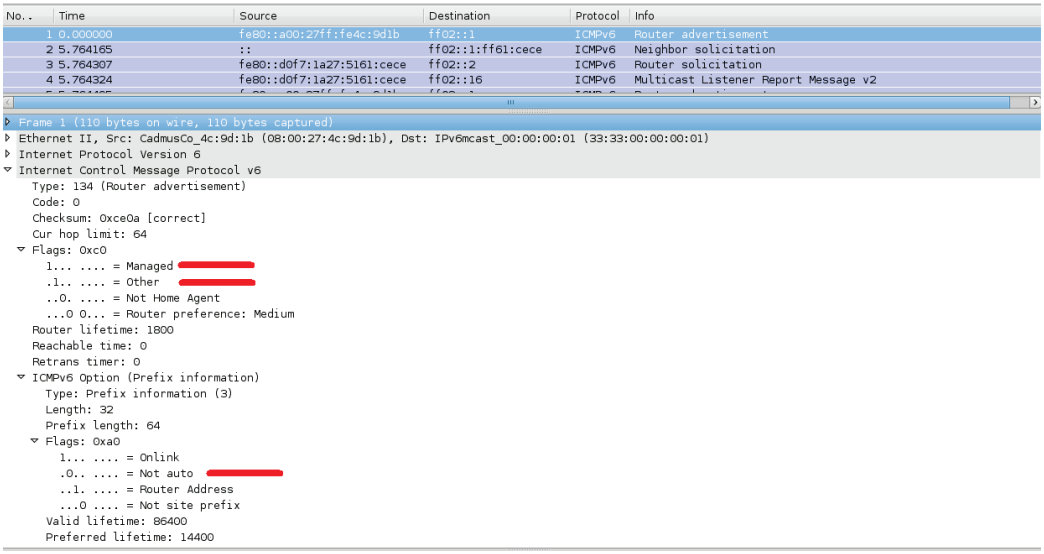
```
sudo ifconfig
```

- Pokud rozhraní obdrželo konfigurační parametry z DHCP serveru, přepněte se do operačního systému Debian (server) a zastavte zachytávání paketů.
- Na zachycené pakety aplikujte filtr **dhcpv6** a prohlédněte si, jaký typ paketů se vyměňuje při vyjednávání konfiguračních parametrů sítě a také jaký typ IPv6 adres se při tom využívá.



obr. 10 Zachycení průběhu přidělování adresy v prostředí IPv6

- V zachycených paketech by se měl nacházet i Router Advertisement pakety, které server posílá v pravidelných intervalech. RA pro svoji činnost využívá ICMPv6 protokol, pro jednodušší nalezení paketů aplikujte tento filtr. Po nalezení patřičného paketu analyzujte přenášené informace jako např. příznaky M a O.



obr. 11 Zachycený paket nesoucí RA

Kontrolní otázky:

1. K čemu slouží DHCP a jaké informace může poskytovat.
2. Jaké znáte metody přidělení IP adresy zařízení.
3. Popište proces přidělení parametrů sítě klientovi od DHCP serveru.
4. Jaké jsou typy automatické konfigurace v IPv6.
5. K čemu slouží tzv. Ohlášení směrovače (Router Advertisement).
6. K čemu slouží příznaky M a O v hlavičce RA zprávy.

Literatura:

[1] DROMS, R., LEMON, T. *DHCP – Příručka administrátora*. 1. vydání. Brno: Computer Press, a.s., 2004. 490 s. ISBN 80-251-0130-4.

[2] SATRAPA, P. *IPv6* [online]. 3. vydání. Praha: CZ.NIC, z.s.p.o., 2011 [cit. 2012-04-13]. Dostupné z WWW: < http://knihy.nic.cz/files/nic/edice/pavel_satrapa_ipv6_2011.pdf >.
ISBN 978-80-904248-0-7.

[3] BŘEZINA, Pavel. *pavka.shotzone.cz* [online]. [cit. 2012-04-15]. DHCP. Dostupné z WWW: < http://pavka.shotzone.cz/dhcp/dhcp-uvod.html >.

A.3 Konfigurace VPN spojení v systému Windows Server a Debian

Cíl:

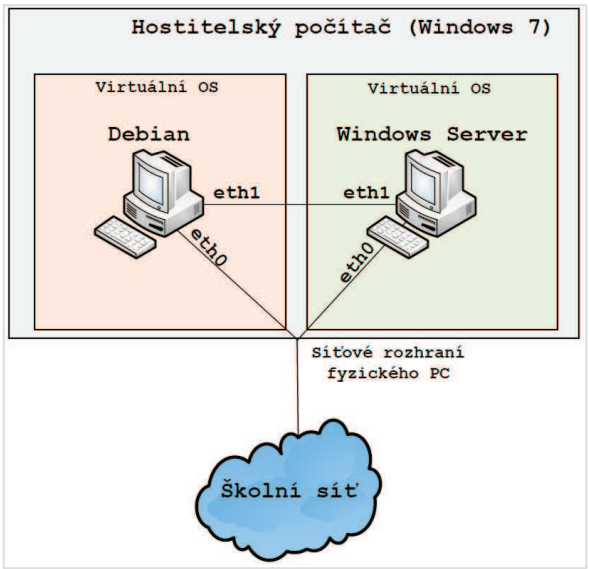
Cílem laboratorní úlohy je seznámit se s virtuálními privátními sítěmi, jejich typy, zabezpečením, tunelováním paketů a implementací v operačním systému Windows Server 2008 R2 a linuxové distribuci Debian.

Požadavky na vybavení pracoviště:

- PC se softwarem VirtualBox
- Virtualizovaný operační systém Microsoft Windows Server 2008 R2 SP1
- 2x Virtualizovaný operační systém Debian (kernel 2.6.32-5-686)

Zapojení pracoviště

V každém virtuálním operačním systému jsou aktivována dvě síťová rozhraní, rozhraní **eth0** je přímo spojeno s učebnou (případně přes NAT) do tohoto rozhraní nebude nutné během úlohy zasahovat, obvykle bude sloužit jen jako „zdroj“ internetu. Síťové rozhraní **eth1** slouží jako virtuální propojení běžících virtualizovaných operačních systému na jednotlivých stanicích. Rozhraní **eth1** má tedy planost pouze na konkrétním počítači a není možné se pomocí něho spojit z jiného počítače v učebně, na tomto rozhraní se budou tedy konfigurovat všechny parametry, aniž bychom zasáhli běh školní sítě.



Přihlašovací údaje

Windows Server 2008 R2 – účet: **Administrator**, heslo **Bars123456789**

Debian (klient i server) – účet: **bars**, heslo **bars** (učet bars je členem skupiny sudo, má tedy správcovská oprávnění)

Úkoly:

- Seznámit se druhy virtuálních privátních sítí a jejich zabezpečením
- V operačním systému Windows Server 2008 R2 nakonfigurovat PPTP VPN server
- Nakonfigurovat VPN spojení typu SSL/TLS v systému Debian

Teoretický úvod:

Viz kapitola 5.

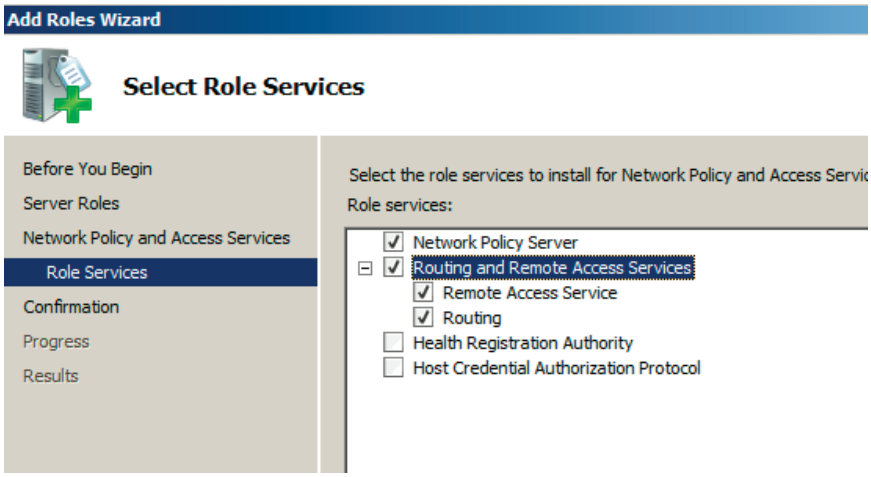
Postup práce

Úkol č. 1

V první části laboratorní úlohy nakonfigurujeme VPN spojení na bázi PPTP v operačním systému Windows Server R2 SP1, ověření spojení bude probíhat z operačního systému Debian. Naneštěstí oba operační systému jsou již propojeny v lokální síti, tudíž nemůžeme simulovat reálnou situaci, kdy se připojujeme na server z Internetu a až poté se budou nacházet oba systémy ve stejné privátní síti.

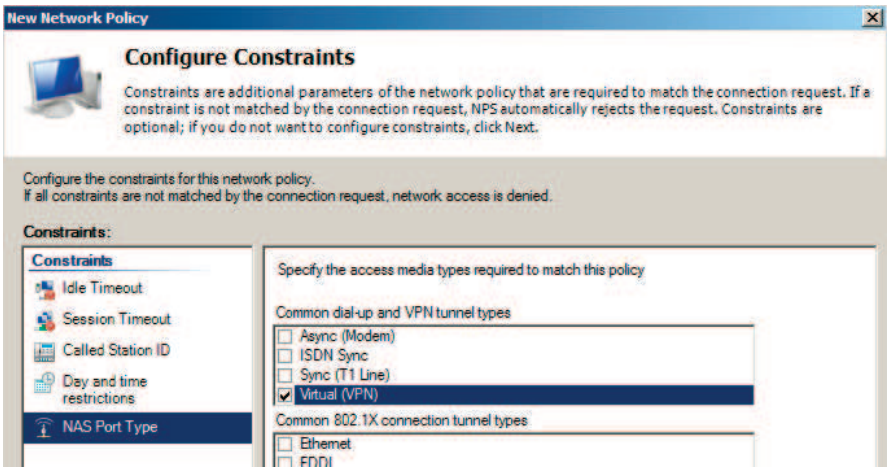
Instalace a konfigurace

- Spusťte virtualizovaný operační systém Windows Server 2008 R2 SP1 a přihlaste se k němu. (Heslo pro Administrátorský účet je Bars123456789)
- Nejprve je nutné vytvořit skupinu uživatelů, kteří se budou moci připojovat pomocí VPN spojení. Otevřete nástroj **Server Manager** (např. přes nabídku Start -> Administrative Tools -> Server Manager). V levém panelu rozklikněte nabídku *Configuration* a dále položku *Local Users and Groups***.** Pod položkou *Users* vytvořte nového uživatele nazvaného jako **bars** s heslem **Password123456** (heslo musí kvůli nastavení bezpečnostních politik splňovat určitou složitost) a **odškrtněte** položku *User must change password at next logon*, jinak nebude možné využít účet pro připojení. Také vytvořte novou skupinu (Groups) nazvanou **VPN** a přidejte do ní vámi vytvořený účet.
- Zůstaňte v nástroji *Server Manager* a přidejte novou roli serveru. V levém panelu položka *Roles* a následně *Add Roles*. Vyberte roli **Network Policy and Access Services** a pokračujte dále, v okně výběru služeb označte položky **Network Policy Server** a **Routing and Remote Access Services** viz obr. 8 a dokončete instalaci.



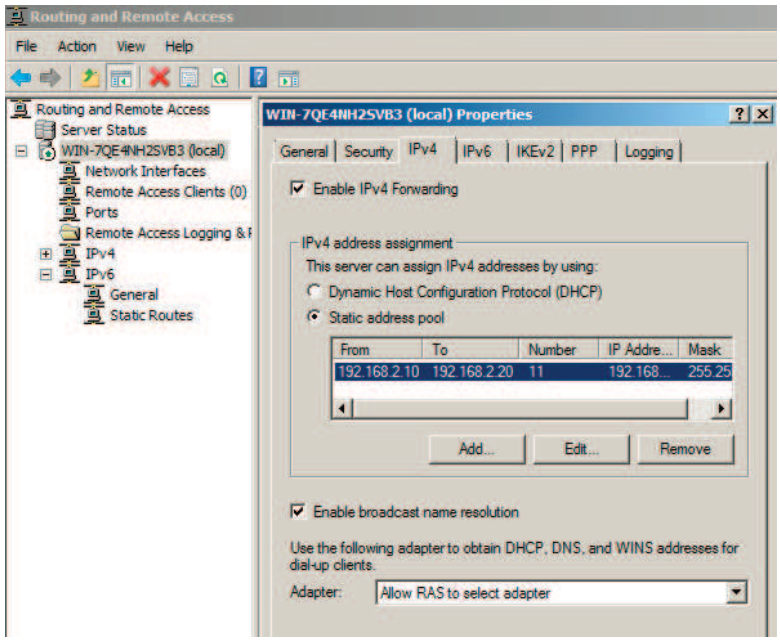
obr. 8 Výběr jednotlivých služeb

- Po dokončení instalace vytvořte pravidlo povolující přístup na VPN server. Otevřete konzoli **Network Policy Server (NAP)** (Start -> Administrative Tools), rozklikněte položku *Policies* a v *Network Policies* vytvořte nové pravidlo. Pravidlo pojmenujte jako VPN a jako typ serveru pro přístup k síti zvolte **Remote Access Server (VPN-Dial up).** Na další stránce přidejte skupinu uživatelů (User Groups), kterou jste vytvořili na začátku úlohy a v dalším kroku udělte přístup (Access granted) pro toto pravidlo.
- V okně metod ověřování **přidejte** jako typ protokolů EAP položku **Microsoft: Secured Password (EAP-MSCHAP v2)**. V dalším okně pod položkou *NAS Port Type,* vyberte jako typ tunelu **Virtual (VPN)**, toto okno slouží k vytváření časových pravidel pro VPN spojení, je zde možné definovat, v jakých časech bude možné se k VPN připojovat, po jaké době nečinnosti se spojení zruší atd., pro naše účely necháme výchozí hodnoty.



obr. 9 Konfigurace pravidla pro přístup k VPN

- V dalším kroku nastavení můžeme definovat další pravidla, např. filtry IP adres, RADIUS server a další, opět pro naše účely není třeba nic měnit a na poslední stránce zkontrolujte nastavení a dokončete konfiguraci pravidla.
- Nyní již přistoupíme k vytvoření samotného VPN spojení. Otevřete konzoli **Routing and Remote Access**, označte náš server a přes nabídku *Action* zvolte *Configure and Enable*… Zvolte vlastní konfiguraci (**Custom configuration**), dále vyberte typ služby *VPN access* a dokončete konfiguraci a spusťte službu. Server automaticky povolí ve firewallu potřebné protokoly (GRE) a porty pro připojení vzdálených klientů.
- Nyní ještě nadefinujeme rozsah IP adres, které se budou přidělovat připojovaným klientům. Označte náš lokální server a přes nabídku *Action* zvolte **Properties**. Přejděte na záložku IPv4, zde přepněte na **Static address pool** a přidejte nový rozsah (192.168.2.10-192.168.2.20).



obr. 10 Přidání rozsahu přidělovaných IP adres

- Tímto je konfigurace kompletní a zbývá funkčnost konfigurace ověřit.

Ověření spojení

- Spusťte operační systém Debian (klient) a přihlaste se pod uživatelským jménem **bars** s heslem **bars**.
- Přes nástroj **NetworkManager** odpojte rozhraní **eth0** od sítě. Zůstane nám tedy jen vnitřní spojení mezi systémy skrze rozhraní eth1.

- Vytvořte nové VPN připojení, jako typ připojení vyberte **PPTP**. Vyplňte IP adresu brány (IP adresa serveru na rozhraní eth1), uživatelské jméno a heslo. V nabídce *Pokročilé* zaškrtněte položku Použít Point-to-Point šifrování (MPPE) a nastavení potvrďte.
- Připojte se k vytvořenému VPN spojení, po úspěšném navázání spojení se u ikony **NetworkManageru** objeví zámek.
- Dále spusťte Terminál, a ověřte konfiguraci IP adres pro rozhraní **ppp0**.

```
sudo ifconfig
```

Úkol č. 2

V druhém úkolu nakonfigurujeme VPN spojení typu SSL/TLS v operačním systému Debian. Využijeme k tomu multiplatformní, zdarma dostupný software OpenVPN. Spojení bude realizováno co možná nejbezpečněji, k tomu využijeme infrastrukturu veřejného klíče (PKI). Ověření konfigurace bude poté probíhat z operačního systému Debian (klient).

Instalace a konfigurace

- Spusťte operační systém Debian (server) a přihlaste se.
- Kromě samotného OpenVPN je nutné nainstalovat ještě *open source* implementaci protokolů SSL a TLS s názvem OpenSSL. Instalace obou nástrojů se provede příkazem v Terminálu (avšak software je již nainstalován):

```
sudo aptitude install openvpn openssl
```

- Jako další je nutné vytvořit infrastrukturu veřejného klíče (PKI), což znamená vytvoření vlastní certifikační autority (CA), klíčů a certifikátů serveru i klienta. OpenVPN obsahuje sadu skriptů, které správu PKI značně zjednodušují.
- Přejděte do složky s umístěnými skripty

```
cd /usr/share/doc/openvpn/examples/easy-rsa/2.0/
```

- Pro zjednodušení práce a zkrácení zadávaných příkazů, se přihlásíme k Terminálu jako **root** (heslo **barsbars**) příkazem:

```
su root
```

- Otevřete soubor *vars* (**gedit vars**) ve kterém nadefinujeme některé parametry, které se poté promítnou do námi vytvářené CA a všech certifikátů. V souboru upravte řádky definující zemi, ve které certifikáty vytváříme, kraj, město, organizaci a email (tyto parametry se nachází na konci souboru). Po dokončení soubor uložte.

```
export KEY_COUNTRY="CZ"
export KEY_PROVINCE="JM"
export KEY_CITY="Brno"
export KEY_ORG="VUTBR"
export KEY_EMAIL="me@myhost.mydomain"
```

- Pro aplikaci změn soubor spusťte.

```
source ./vars
```

- Nyní přistoupíme k vytvoření všech potřebných certifikátů a klíčů. Při vytváření není potřeba nic vyplňovat, použijí se parametry ze souboru *vars*, pouze je nutné potvrdit podpis certifikátů klávesou **y**. **Postupně** vkládejte do Terminálu následující řádky:

```
./clean-all
./build-ca
./build-key-server server
./build-key klient
./build-dh
cd keys/
openvpn --genkey --secret ta.key
```

Vysvětlení jednotlivých příkazů: `./clean-all` – slouží ke smazání obsahu složky, do které se budou generovat certifikáty. `./build-ca` – vytvoří certifikát a klíč certifikační autority. `./build-key-server server` – vygeneruje certifikát a privátní klíč pro server (server.key, server.crt). `./build-key klient` – certifikát a klíč pro klienta, `./build-dh` – Diffie-Hellmanův parametr, umožňuje přes nezabezpečený kanál vytvořit mezi komunikujícími stranami šifrované spojení, bez předchozího dohodnutí šifrovacího klíče. Příkaz `openvpn --genkey --secret ta.key` vygeneruje klíč pro TLS autentikaci.

- Vytvořené certifikáty překopírujte do pracovní složky nástroje openvpn (*/etc/openvpn/keys*):

```
cp -r /usr/share/doc/openvpn/examples/easy-rsa/2.0/keys/ /etc/openvpn/keys/
```

- Vytvořte soubor */etc/openvpn/vpn_server.conf*, nakopírujte do něj následující obsah a restartujte službu openvpn (**/etc/init.d/openvpn restart**):

```
mode server #režim činnosti
tls-server #režim činnosti
dev tap0 #režim typ virtuálního rozhraní
port 1194 #port, na kterém server naslouchá
proto udp #typ spojení

ifconfig 10.0.1.1 255.255.255.0 #definice IP konfigurace serveru
ifconfig-pool 10.0.1.100 10.0.1.200 255.255.255.0
duplicate-cn #povoluje současné přihlášení více klientů se stejným certifikátem
#klíče systemu
ca /etc/openvpn/keys/ca.crt
cert /etc/openvpn/keys/server.crt
key /etc/openvpn/keys/server.key
dh /etc/openvpn/keys/dh1024.pem
tls-auth /etc/openvpn/keys/ta.key 0 #TLS autentizace před samotným
spojením

log-append /var/log/openvpn #umisteni logování
status /tmp/vpn.status 10 #zpusob logovani
mute 20
comp-lzo #Nastaveni komprese toku dat
verb 3 #Míra podrobnosti logu
keepalive 1 120
```

Ověření konfigurace

- Spusťte operační systém Debian (klient) a přihlaste se.
- I zde je OpenVPN s OpenSSL již nainstalováno, přejdeme rovnou k přemístění certifikátů a klíčů potřebných po vytvoření spojení.
- U klienta vytvořte složku **keys** v pracovním adresáři OpenVPN (**sudo mkdir /etc/openvpn/keys**)
- Nyní je nutné **bezpečně** přenést certifikáty a klíče (hlavně klíče se nesmí dostat do rukou cizích osob), ideální scénář je zajít si pro certifikát osobní cestou, případně využitím různých smart karet atd. My použijeme překopírování souborů pomocí šifrovaného spojení **scp**.
- Přepněte se na Debian (server), přejděte do složky (**cd /etc/openvpn/keys/**) a následujícím příkazem (nutno doplnit IP adresu klienta) zkopírujte certifikát a klíč klienta, certifikát certifikační autority, a klíč pro TLS autentizaci.

```
scp klient.crt klient.key ta.key ca.crt  root@[IP_adresa_klienta]:/etc/openvpn/keys/
```

- Po překopírování se vraťte zpět ke klientovi a vytvořte soubor **/etc/openvpn/vpn_client.conf** a naplňte ho následujícím obsahem:

```
remote 192.168.1.2 #IP adresa OpenVPN serveru
tls-client
dev tap
pull #povoluje stažení konfigurace ze serveru

ca /etc/openvpn/keys/ca.crt
cert /etc/openvpn/keys/klient.crt
key /etc/openvpn/keys/klient.key
tls-auth /etc/openvpn/keys/ta.key 1

mute 20
comp-lzo
verb 3
ns-cert-type server
```

- Spuštění OpenVPN klienta s naší konfigurací a připojení k serveru se provede:

```
sudo openvpn /etc/openvpn/vpn_client.conf
```

- Spusťte nové terminálové okno, a ověřte IP konfiguraci pro rozhraní **tap0.**

Kontrolní otázky:

1. K čemu slouží virtuální privátní sítě (VPN)
2. Co je to tunelování paketů, a popište obecný princip tunelování.
3. Popište zapouzdření protokolu PPTP.
4. Které bezpečnostní služby (mechanizmy) může zajišťovat protokol IPSec.
5. Jaký je rozdíl mezi transportním a tunelovacím módem, který má vyšší nároky na přenosové pásmo.
6. Kde se nejčastěji setkáte se SSL protokolem.

Literatura

[1] PUŽMANOVÁ, Rita. Virtuální privátní sítě pro vzdálený přístup. *DSL.cz: Telekomunikace* [online]. 2006 [cit. 2012-05-14]. Dostupné z: < http://www.dsl.cz/clanek/511-virtualni-privatni-site-pro-vzdaleny-pristup >.

[2] HABRMAN, Robert. Síťové protokoly (XXV. část): Protokol L2TP. *OWEBU.cz* [online]. 2008 [cit. 2012-05-18]. Dostupné z: http://owebu.bloger.cz/PC-site/Sitove-protokoly-XXV-cast-Protokol-L2TP.

[3] MIKULEC, Martin. VPN - virtuální privátní sítě. *OWEBU.cz* [online]. 2009 [cit. 2012-05-18]. Dostupné z: <http://owebu.bloger.cz/stitek_?s=VPN+-+virtu%C3%A1ln%C3%AD+priv%C3%A1tn%C3%AD+s%C3%ADt%C4%9B >.

[4] LUHOVÝ, Karel. Virtuální privátní sítě VPN. *Svět sítí* [online]. 2003 [cit. 2011-12-02]. Dostupné z WWW: < http://www.svetsiti.cz/rubrika.asp?rid=17&tid=219 >.

[5] PRŮCHA, Ondřej. Vše co jste chtěli vědět o VPN, ale báli jste se zeptat. *O VPN* [online]. 2005 [cit. 2011-12-02]. Dostupné z WWW: < http://home.zcu.cz/~ondrous/index.php?vyber=0 >.

[6] Virtual Private Networking. *Technet.microsoft.com* [online]. 2001 [cit. 2012-05-18]. Dostupné z: < http://technet.microsoft.com/en-us/library/bb742566.aspx >.

A.4 Konfigurace NAT a firewallu v systému Windows Server a Debian

Cíl:

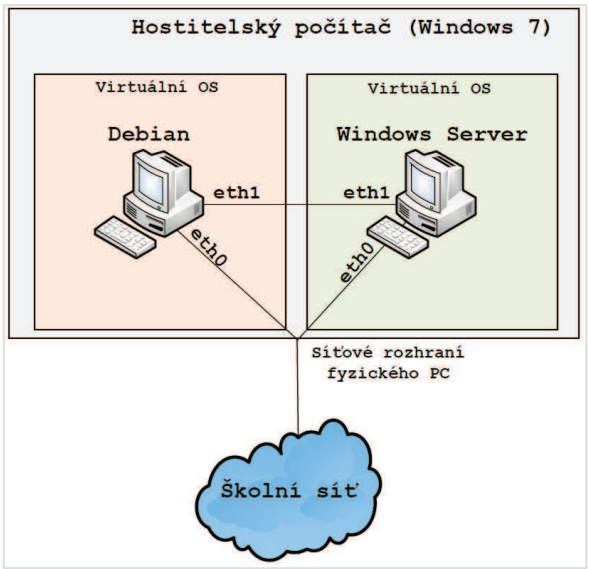
Seznámení se s principem a možnostmi překladu síťových adres, jeho konfiguraci v operačním systému Windows Server 2008 R2 a linuxové distribuci Debian. Dále také seznámení s konfigurací firewallu pomocí nástroje *iptables* v Debianu.

Požadavky na vybavení pracoviště:

- PC se softwarem VirtualBox
- Virtualizovaný operační systém Microsoft Windows Server 2008 R2 SP1
- 2x Virtualizovaný operační systém Debian 6 (kernel 2.6.32-5-686)

Zapojení pracoviště

V každém virtuálním operačním systému jsou aktivována dvě síťová rozhraní, rozhraní **eth0** je přímo spojeno s učebnou (případně přes NAT) do tohoto rozhraní nebude nutné během úlohy zasahovat, obvykle bude sloužit jen jako „zdroj“ internetu. Síťové rozhraní **eth1** slouží jako virtuální propojení běžících virtualizovaných operačních systému na jednotlivých stanicích. Rozhraní **eth1** má tedy planost pouze na konkrétním počítači a není možné se pomocí něho spojit z jiného počítače v učebně, na tomto rozhraní se budou tedy konfigurovat všechny parametry, aniž bychom zasáhli běh školní sítě.



Přihlašovací údaje

Windows Server 2008 R2 – účet: **Administrator**, heslo **Bars123456789**

Debian (klient i server) – účet: **bars**, heslo **bars** (učet bars je členem skupiny sudo, má tedy správcovská oprávnění)

Úkoly:

- Seznámit se principem NAT a konfigurací firewallu pomocí nástroje *iptables*
- Nakonfigurovat NAT a směrování portů v systému Windows Server
- V systému Debian nakonfigurovat překlad adres včetně nastavení firewallu

Teoretický úvod:

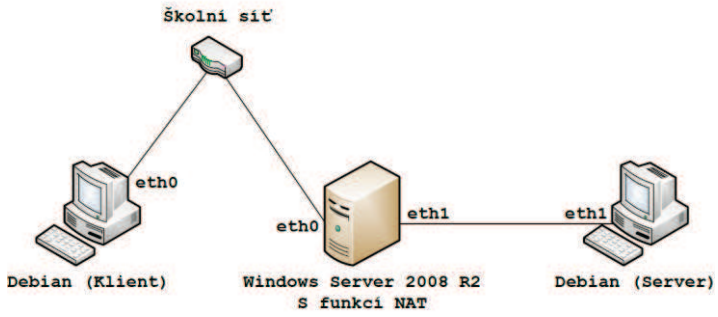
Viz kapitola 6.

Postup práce:

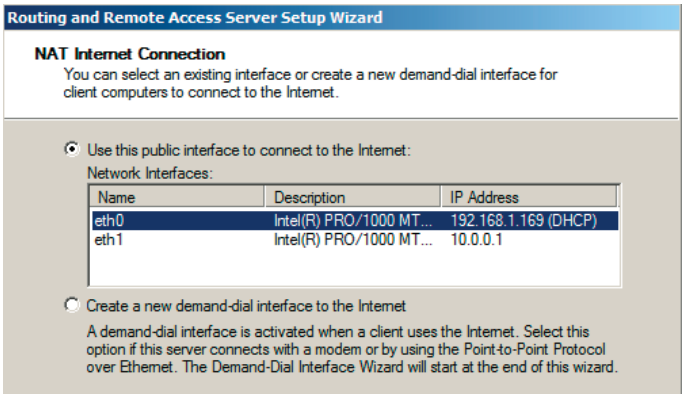
Úkol č. 1:

V prvním úkolu nakonfigurujeme NAT v operačním systému Windows Server 2008 R2 a pokusíme se přesměrovat porty pro SSH spojení. V této úloze bude nutné spouštět všechny 3 virtualizované operační systémy, abychom byli schopni ověřit funkčnost konfigurace.

- Spusťte operační systém Debian (klient) a přihlaste se k němu. Přes nástroj NetworkManager (ikona sítě vedle data a času) **odpojte** síťové rozhraní **eth1**.
- Dále spusťte operační systém Debian (Server) a odpojte síťové rozhraní **eth0** a pro síťové rozhraní **eth1** nastavte statickou konfiguraci IP adres (IP adresa 10.0.0.10, maska 255.0.0.0, brána 10.0.0.1, DNS 8.8.8.8).
- Zapněte také Windows Server 2008 R2 a nastavte statickou adresu pro rozraní **eth1** (IP adresa 10.0.0.1, maska 255.0.0.0, brána ani DNS není nutné vyplňovat).
- Po nastavení bude naše síť vypadat následovně:

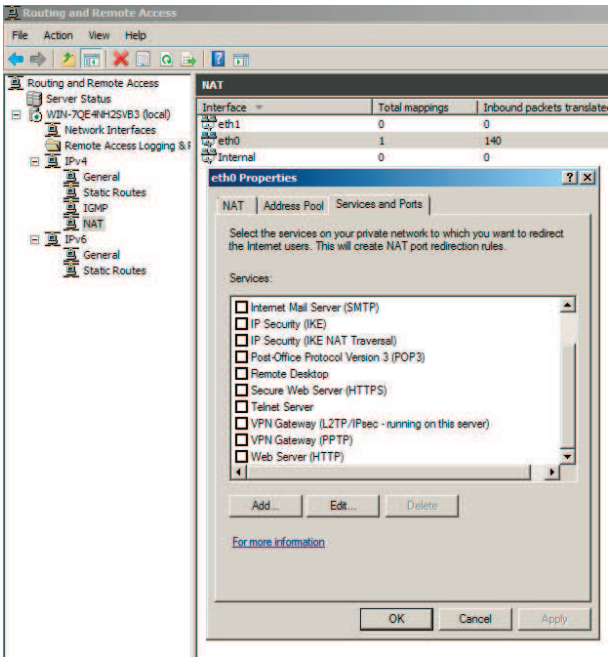


- Nyní když ověříme funkčnost internetu na Debian Serveru, zjistíme, že nefunguje. Pro zprovoznění je nutné na Windows Serveru nakonfigurovat NAT, kdy **eth0** na Windows bude sloužit jako zdroj internetu a **eth1** je rozhraní do vnitřní sítě.
- Přepněte se do operačního systému Windows Server a spusťte nástroj **Server Manager** (např. přes nabídku Start -> Administrative Tools -> Server Manager) a přidejte novou roli serveru. V levém panelu položka *Roles* a následně *Add Roles*. Vyberte roli **Network Policy and Access Services** a pokračujte dále, v okně výběru služeb označte položku **Routing and Remote Access Services** a dokončete instalaci.
- Otevřete konzoli **Routing and Remote Access** (Start -> Administrative Tools), označte náš server a přes nabídku *Action* zvolte *Configure and Enable Routing and Remote Access.* V konfiguračním okně vyberte položku **Network address translation (NAT)** a pokračujte na další stránku. Zde vyberte rozhraní, které bude sloužit jako zdroj internetu, v našem případě jde o rozhraní **eth0** a dokončete konfiguraci.



obr. 8 Výběr rozhraní připojeného do internetu

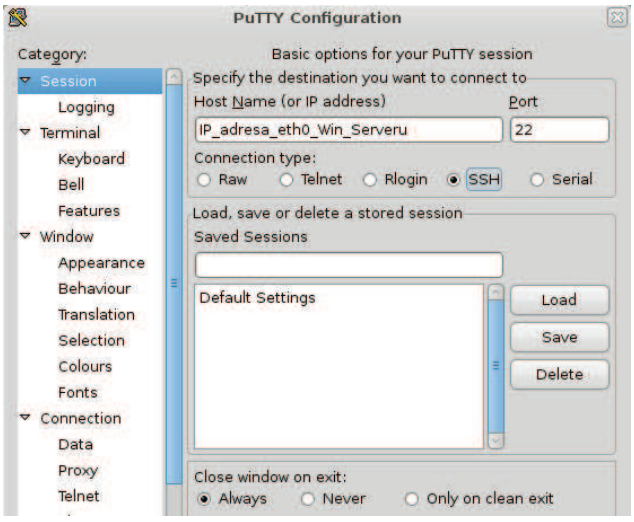
- Tím je základní konfigurace překladu adres hotová, a nyní by měl v Debian Serveru fungovat internet (spojení do vnější sítě). Teď je tedy celá síť 10.0.0.0/8 „skryta“ za IP adresu Windows Serveru přiřazené na rozhraní eth0.
- Nyní přesměrujeme veškerou SSH komunikaci na portu 22 přicházející na ethernetové rozhraní z vnější sítě na IP adresu Debian serveru ve vnitřní síti. Požadavek na připojení bude tedy mít jako cílovou IP adresu, adresu rozhraní **eth0** a po průchodu NATem bude směrován na adresu 10.0.0.10 ve vnitřní síti.
- V konzoli **Routing and Remote Access** v levém panelu rozklikněte strom IPv4 a přejděte do položky NAT, kde se nachází rozhraní, mezi kterými pracuje překlad adres. Poklepejte na rozhraní **eth0** a přejděte do záložky **Services and Ports**, kde vytvořte novou službu s názvem např. SSH, jako protokol zvolte **TCP**, příchozí a odchozí port nastavte **22** a privátní adresu nastavte adresu Debian Serveru.



obr. 9 Konfigurace přesměrování portů

Ověření konfigurace

- V operačním systému Debian (Server) běží SSH server, na který je směrován veškerý provoz na portu 22 z vnější sítě. Je tedy možné se k němu připojit z vnější sítě ze systému Debian (klient).
- Přepněte se na operační systém Debian Klient a spusťte nástroj **Putty** (Aplikace -> Internet). Vyplňte IP adresu Windows Serveru a připojte se.
- Jako přihlašovací jméno a heslo použijte **bars bars**. Pokud se připojíte k serveru, ověříte tím funkčnost překladu adres i přesměrovaného portu do vnitřní sítě.

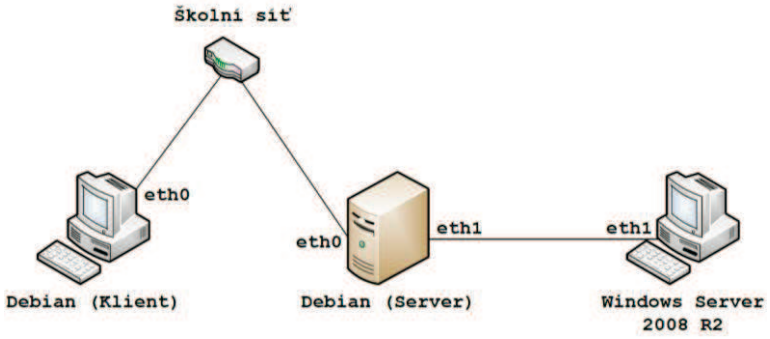


Úkol č. 2:

V druhé polovině úlohy nakonfigurujeme překlad adres v operačním systému Debian pomocí nástroje iptables a zabezpečíme vnitřní síť a Debian (Server) firewallem. Taktéž se pokusíme přesměrovat porty z vnější sítě do vnitřní pro konkrétní službu.

Před začátkem druhé části vypněte a znovu zapněte operační systém Windows Server 2008 R2, tím zrušíte všechny dříve aplikované změny v tomto systému.

- Jako první krok je potřeba změnit zapojení naší sítě z předchozího úkolu. Operační systém Debian (klient) zůstane stále ve vnější síti, není tedy nutné u něj nic měnit.
- V systému Debian (Server) je nutné aktivovat ethernetové rozhraní **eth0** (např. přes NetworkManager) a také je nutné upravit statickou konfiguraci adres pro rozhraní **eth1** (IP adresa 10.0.0.1, maska 255.0.0.0, brána ani DNS není nutné vyplňovat)
- U Windows Serveru zakažte (deaktivujte) rozhraní **eth0** a pro **eth1** nastavte následující parametry, IP adresa 10.0.0.10, maska 255.0.0.0, výchozí brána 10.0.0.1 a DNS 8.8.8.8 .
- Po provedení všech úprav bude síť zapojená jako na obr. 10. Pokud zkusíme z každého systému funkčnosti internetu, zjistíme, že z Windows Serveru se nedostaneme z vnitřní sítě, ve vnitřní síti bude internet fungovat až po zprovoznění NATu.



obr. 10 Zapojení sítě pro Úkol č. 2

- Přepněte se do systému Debian (Server) a spusťte Terminál. Nejprve je nutné zapnout předávání paketů mezi síťovými rozhraními, to se provede příkazem:

```
sudo sysctl -w net.ipv4.conf.all.forwarding=1
```

- Nyní se již dostáváme ke konfiguraci NATu pomocí nástroje iptables. Předtím než začneme vytvářet nové řetězce pravidel, pro jistotu zadejte příkazy, které všechna uživatelsky definovaná pravidla vymažou.

```
sudo iptables -F
sudo iptables -F -t mangle
sudo iptables -X -t mangle
sudo iptables -F -t nat
sudo iptables -X -t nat
sudo iptables -F -t filter
sudo iptables -X -t filter
```

- Změňte výchozí nastavení politik firewallu tak, aby veškeré příchozí a směrované pakety byly zahozeny, pouze odchozí pakety budou povoleny. Tzn., že všechny příchozí a směrované pakety, pro které nebude vytvořené povolující pravidlo firewallem neprojdou, vše co není povoleno, bude zahozeno.

```
sudo iptables -P INPUT DROP
sudo iptables -P OUTPUT ACCEPT
sudo iptables -P FORWARD DROP
```

- Nyní vytvořte pravidlo, které bude překládat vnitřní rozsah adres (10.0.0.0/8) za vnější IP adresu Debian (serveru). Jedná se tedy o tzv. IP maškarádu. A také vytvořte pravidla pro předávání paketů mezi rozhraními vnitřní a vnější sítě.

```
sudo iptables -A POSTROUTING -t nat -o eth0 -s 10.0.0.0/24 -j MASQUERADE
sudo iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
sudo iptables -A FORWARD -i eth0 -o eth1 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Vysvětlení: První řádek je pravidlo, které vytváří samotný SNAT (maškaradu), parametr *–A* značí přidání nového pravidla, *POSTROUTING* slouží k úpravě odchozích paketů viz Teoretický úvod, *-t nat* značí, že se pravidlo uloží do tabulky NAT, *-o* udává výstupní rozhraní, *-s* definuje zdrojovou adresu a parametr *–j* je pokyn, pro naložení s paketem. Druhý řádek vytváří pravidlo typu FORWARD, které povolí předávání všech paketů z vnitřního rozhraní **eth1** na rozhraní **eth0**. Poslední pravidlo povoluje předávání paketů z vnější sítě do vnitřní sítě, pro již navázaná spojení, tudíž nově vytvořená spojení z vnější sítě do vnitřní z bezpečnostních důvodu neprojdou.

- Nyní by měl začít fungovat internet na Windows Serveru, ale např. ping na Debian (Server) nepůjde, jelikož firewall je nastavený aby zahazoval veškerou příchozí komunikaci. Nynější nastavení firewallu může být poměrně omezující, např. pokud by nám na Debian (Serveru) běžel FTP server, nebylo by možné se k němu připojit ani z vnější ani z vnitřní sítě. Taktéž pokud bychom chtěli provozovat nějakou veřejně dostupnou službu na Windows Serveru ve vnitřní síti, měli bychom smůlu.
- Vytvořte nová pravidla povolující přístup na server z lokální smyčky (lo), vnitřní sítě a povolení průchodu paketů z připojení navázaných serverem. Nyní je Debian (Server) plně přístupný z vnitřní důvěryhodné sítě.

```
sudo iptables -A INPUT -i lo -j ACCEPT
sudo iptables -A INPUT -i eth1 -j ACCEPT
sudo iptables -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

- V dalším kroku vytvořte pravidlo povolující ping z vnitřní a vnější sítě, které bude mít limit maximálně 10 zpracovaných icmp paketů za 1 sekundu. Omezení počtu pingů je vhodné, abychom se vyvarovali nadměrné zátěži serveru a sítě při odpovídání na icmp dotazy.

```
sudo iptables -A INPUT -p icmp --icmp-type echo-request \
-m limit --limit 1/s --limit-burst 10 -j ACCEPT
```

- Na Debian (Serveru) beží SSHd server, vytvořte pravidlo povolující průchod nově příchozího spojení na portu 22 využívající protokol TCP.

```
sudo iptables -A INPUT -p tcp --dport 22 -m state --state NEW -j ACCEPT
```

- Ve vnitřní síti, na Windows Serveru je spuštěná služba pro připojení ke vzdálené ploše (port 3398). Aby bylo možné se k ploše připojit z vnější sítě, je nutné vytvořit pravidlo pro předávání paketů pro nově příchozí i navázané spojení s cílovou adresou Windows Serveru a cílovým portem 3398. Dále je nutné vytvořit pravidlo DNAT pro změnu cílové IP adresy.

```
sudo iptables -A FORWARD -m state -p tcp -d 10.0.0.10 --dport 3389 \
--state NEW,ESTABLISHED,RELATED -j ACCEPT
sudo iptables -t nat -A PREROUTING -p tcp --dport 3389 -j DNAT \
--to-destination 10.0.0.10:3389
```

Ověření funkčnosti konfigurace

- Ověřte funkčnost připojení na SSH server, který běží na Debian (Serveru), musíte se připojovat na IP adresu pro rozhraní **eth0**.
- Dále ověřte správné fungování směrování portu 3389 na vnitřní adresu Windows Serveru. Zapněte nástroj **Remmina**, jděte do nabídky Akce -> Rychlé připojení, jako protokol zvolte RDP, vyplňte IP adresu a přihlašovací údaje na Windows Server (Opět se musíme připojovat na IP adresu Debian (Serveru) na rozhraní **eth0**!!!)

Kontrolní otázky:

1. Co je to NAT a k čemu slouží.
2. Popište, jak se mění IP adresy odesílatele a příjemce při průchodu NATem.
3. Rozdíl mezi statickým a dynamickým překladem adres.
4. Co je to IP maškaráda a kde se nejčastěji využívá.
5. Co je to firewall a jaké typy znáte.

Literatura

[1] BAČA, Radim. Semestrální projekt do předmětu Technologie počítačových sítí. *cs.vsb.cz* [online]. [cit. 2012-05-21]. Dostupné z: < http://www.cs.vsb.cz/grygarek/TPS/projekty/0405Z/NAT/Nat.htm >.

[2] PETŘÍČEK, Miroslav. Stavíme firewall. *Root.cz* [online]. 2001 [cit. 2012-05-21]. Dostupné z: < http://www.root.cz/serialy/stavime-firewall/ >.

[3] TYSON, Jeff. How Network Address Translation Works. *HowStuffWorks.com* [online]. [cit. 2012-05-21]. Dostupné z: < http://computer.howstuffworks.com/nat.htm/printable >.

[4] SATRAPA, Pavel. NAT vesus NAP. *Lupa.cz* [online]. 2005 [cit. 2012-05-21]. Dostupné z: < http://www.lupa.cz/clanky/nat-vesus-nap/ >.

Manuály k použitým nástrojům

http://linux.die.net/man/8/iptables

A.5 Konfigurace SNMP v systému Windows Server a Debian

Cíl:

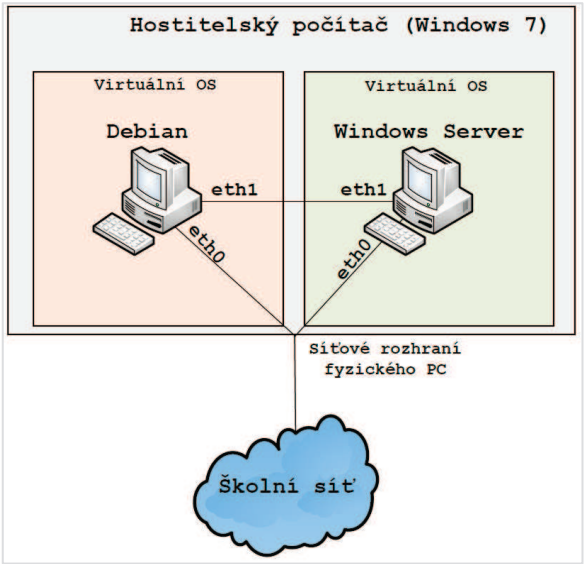
Cílem úlohy je seznámit se s protokolem SNMP, typy operací a formátem zpráv. Dále pak nakonfigurovat monitorovací nástroj *Cacti* v systému Debian.

Požadavky na vybavení pracoviště:

- PC se softwarem VirtualBox
- Virtualizovaný operační systém Microsoft Windows Server 2008 R2 SP1
- 2x Virtualizovaný operační systém Debian 6 (kernel 2.6.32-5-686)

Zapojení pracoviště

V každém virtuálním operačním systému jsou aktivována dvě síťová rozhraní, rozhraní **eth0** je přímo spojeno s učebnou (případně přes NAT) do tohoto rozhraní nebude nutné během úlohy zasahovat, obvykle bude sloužit jen jako „zdroj“ internetu. Síťové rozhraní **eth1** slouží jako virtuální propojení běžících virtualizovaných operačních systému na jednotlivých stanicích. Rozhraní **eth1** má tedy planost pouze na konkrétním počítači a není možné se pomocí něho spojit z jiného počítače v učebně, na tomto rozhraní se budou tedy konfigurovat všechny parametry, aniž bychom zasáhli běh školní sítě.



Přihlašovací údaje

Windows Server 2008 R2 – účet: **Administrator**, heslo **Bars123456789**

Debian (klient i server) – účet: **bars**, heslo **bars** (učet bars je členem skupiny sudo, má tedy správcovská oprávnění)

Úkoly:

- Seznámit se s protokolem SNMP
- Nakonfigurovat SNMP agenta v operačním systému Windows Server
- Monitorovat zařízení pomocí nástroje Cacti

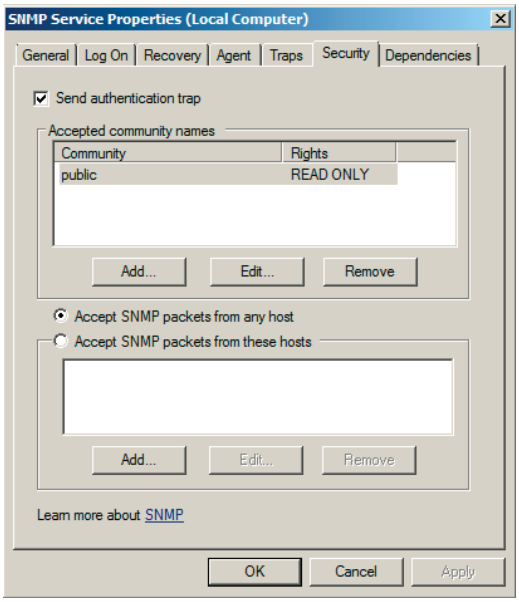
Teoretický úvod:

Viz kapitola 7.

Postup práce

V laboratorní úloze nakonfigurujeme SNMP agenta v operačním systému Windows Server, jako manažer nám bude sloužit nástroj Cacti nainstalovaný v systému Debian. Cacti je nástroj pro tvoření grafů vytížení CPU, využití pásma ethernetových rozhraní, pro monitoring volného místa na harddisku nebo počtu dotazů za sekundu na MySQL server a mnoho dalšího.

- Spusťte operační systém Windows Server 2008 R2 SP1 a přihlaste se k němu (Heslo pro Administrátorský účet je **Bars123456789**).
- Jako první přidejte podporu pro SNMP služby, otevřete nástroj **Server Manager** (např. přes nabídku Start -> Administrative Tools -> Server Manager). V levém panelu klikněte na položku *Features* a přidejte novou vlastnost (*Add Features*). Ve výběru vlastností vyberte **SNMP Services** včetně přidružených položek a dokončete instalaci.
- Konfigurace SNMP agenta se provádí přes konzoli *Services* (Start -> Administrative Tools -> Services), zde vyhledejte službu *SNMP Service* a otevřete ji. V záložce *Agent* můžete vyplnit kontaktní údaje (email) na osobu zodpovědnou na server a umístění serveru. Na záložce *Security* přidejte novou komunitu s názvem *public* s právy číst a dále zaškrtněte volbu *Accept SNMP packets from any host* viz obr. 5.



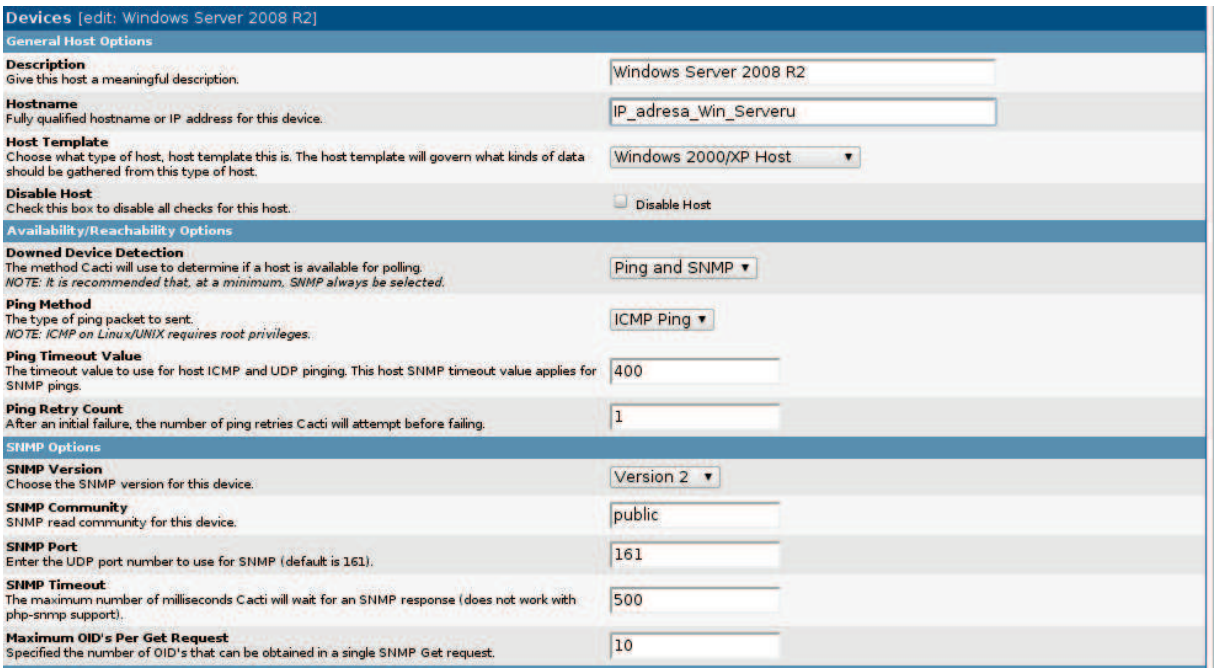
obr. 5 Nastavení SNMP

- Tím se konfigurace SNMP agenta hotová a přejdeme k instalaci nástroje Cacti.
- Spusťte operační systém Debian (Server) a přihlaste se.
- Přes Terminál proveďte instalaci nástroje Cacti příkazy:

```
sudo aptitude update && sudo aptitude install cacti
```

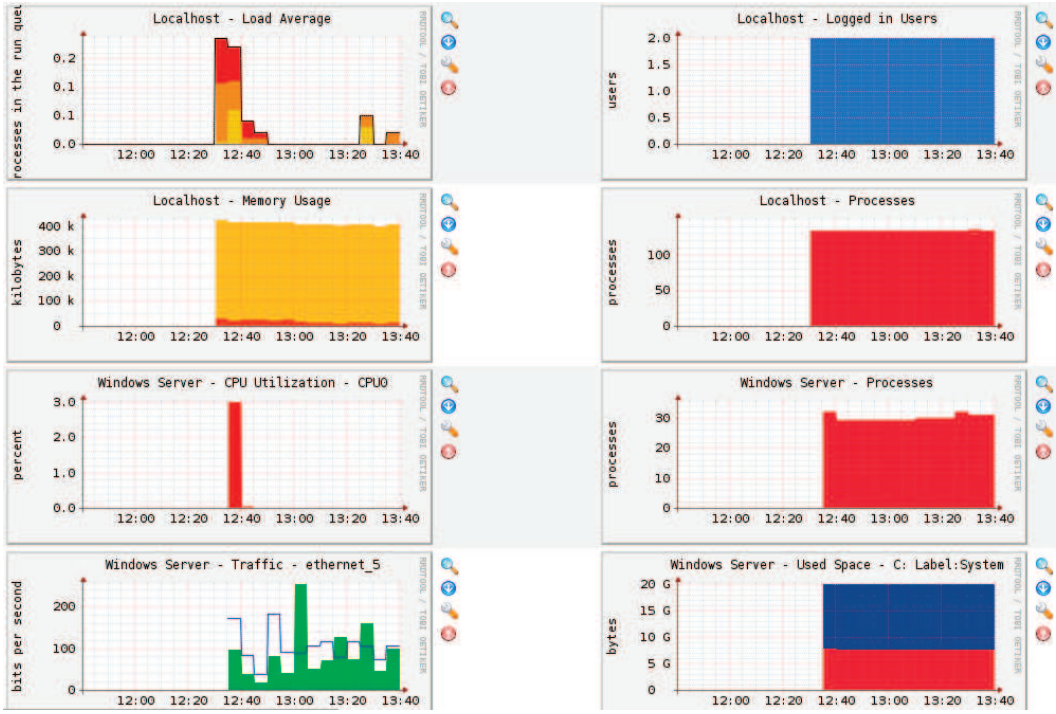
  Systém si sám vyřeší všechny závislosti balíčku a automaticky nainstaluje další potřebné balíčky jako apache2, mysql, php atd.
- Při instalaci budete dotázáni na heslo pro mysql roota, pro jednoduchost zvolte **bars**, jako typ webového serveru zvolte **Apache2**, při dotazu zda chcete nastavit databázi pomocí **dbconfig-common** zvolte Ano. Při dalším dotazu na heslo administrátora databáze volte opět bars.
- Po instalaci spusťte webový prohlížeč Chromium a přejděte na adresu **http://127.0.0.1/cacti** kde dokončíte instalaci.
- Po dokončení instalace se ocitnete na přihlašovací stránce do nástroje Cacti, výchozí přihlašovací jméno a heslo je admin/admin, heslo musí být z bezpečnostních důvodů při prvním přihlášení změněno (volte opět bars).

- Prvním krokem při přidání nového monitorovaného zařízení, je vytvoření nového zařízení v nabídce *Devices*. Přejděte tedy do *Devices* a tlačítkem Add (v pravo nahoře) přidejte nové zařízení, proveďte nastavení, viz obr. 6.



obr. 6 Přidání monitorovaného zařízení

- Při vybrání šablony Windows 2000/XP Host, se automaticky přidají i šablony pro zachytávaná data ze SNMP agenta. Nyní je potřeba z těchto dat vytvořit grafy, pro lepší přehled o zařízení. Ve vytvořeném zařízení klikněte na položku *Create Graphs for this Host*, na další stránce vyberte požadované grafy (např. grafy počtu přihlášených uživatelů, běžících procesů, vytížení procesoru a statistik z ethernetových rozhraní) „zatržítkem“ na pravé straně a klikněte na tlačítko *Create*.
- Nyní je ještě nutné přidat vytvořené grafy do tzv. ***Graph Tree,*** otevřete tedy tuto položku, klikněte na *Default Tree* a přidejte novou položku, jako *Tree Item type* zvolte *Host* a vyberte naše zařízení Windows Server 2008 a klikněte na *Create.*
- Přepněte se na záložku **graphs** a změňte položku Presents (časová osa grafů) na půl hodiny. A zhruba po 5-10 minutách se začnou generovat první statistiky do grafů, viz obr. 7.
- Během čekání na vygenerování grafu, zapněte nástroj Wireshark (**sudo wireshark**) a aplikujte filtr **snmp** a prohlédněte si formát vyměňovaných zpráv.



obr. 7 Grafy z monitorovacího nástroje Cacti

Kontrolní otázky:

1. K čemu slouží SNMP protokol.
2. Co je to OID identifikátor.
3. K čemu slouží MIB databáze.
4. K čemu slouží operace Get a GetBulk.
5. Popište formát SNMP zpráv.

Literatura

[1] ŠTĚPÁNEK, Jakub. Sledování sítě. Fi.muni.cz [online]. [cit. 2012-05-22]. Dostupné z: http://www.fi.muni.cz/~kas/p090/referaty/2011-jaro/ut/snmp.html

[2] LICEHAMMER, Slávek. SNMP - Simple Network Management Protocol. Fi.muni.cz [online]. [cit. 2012-05-22]. Dostupné z: http://www.fi.muni.cz/~kas/p090/referaty/2010-podzim/po/snmp.html

[3] BOUŠEK, Petr. SNMP - Simple Network Management Protocol. Samuraj.cz [online]. 2006 [cit. 2012-05-22]. Dostupné z: http://www.samuraj-cz.com/clanek/snmp-simple-network-management-protocol/

[4] SNMP protokol a jeho využití. Hw.cz [online]. 2003 [cit. 2012-05-22]. Dostupné z: http://www.hw.cz/Produkty/ART957-SNMP-protokol-a-jeho-vyuziti.html

[5] KLAŠKA, Luboš. Vznik a principy SNMP. Svetsiti.cz [online]. 2000 [cit. 2012-05-22]. Dostupné z: http://www.svetsiti.cz/clanek.asp?cid=Format-SNMP-zprav-1462000

[6] What is SNMP?. *Pulsewan.com* [online]. [cit. 2012-05-22]. Dostupné z: http://www.pulsewan.com/data101/snmp_basics.htm