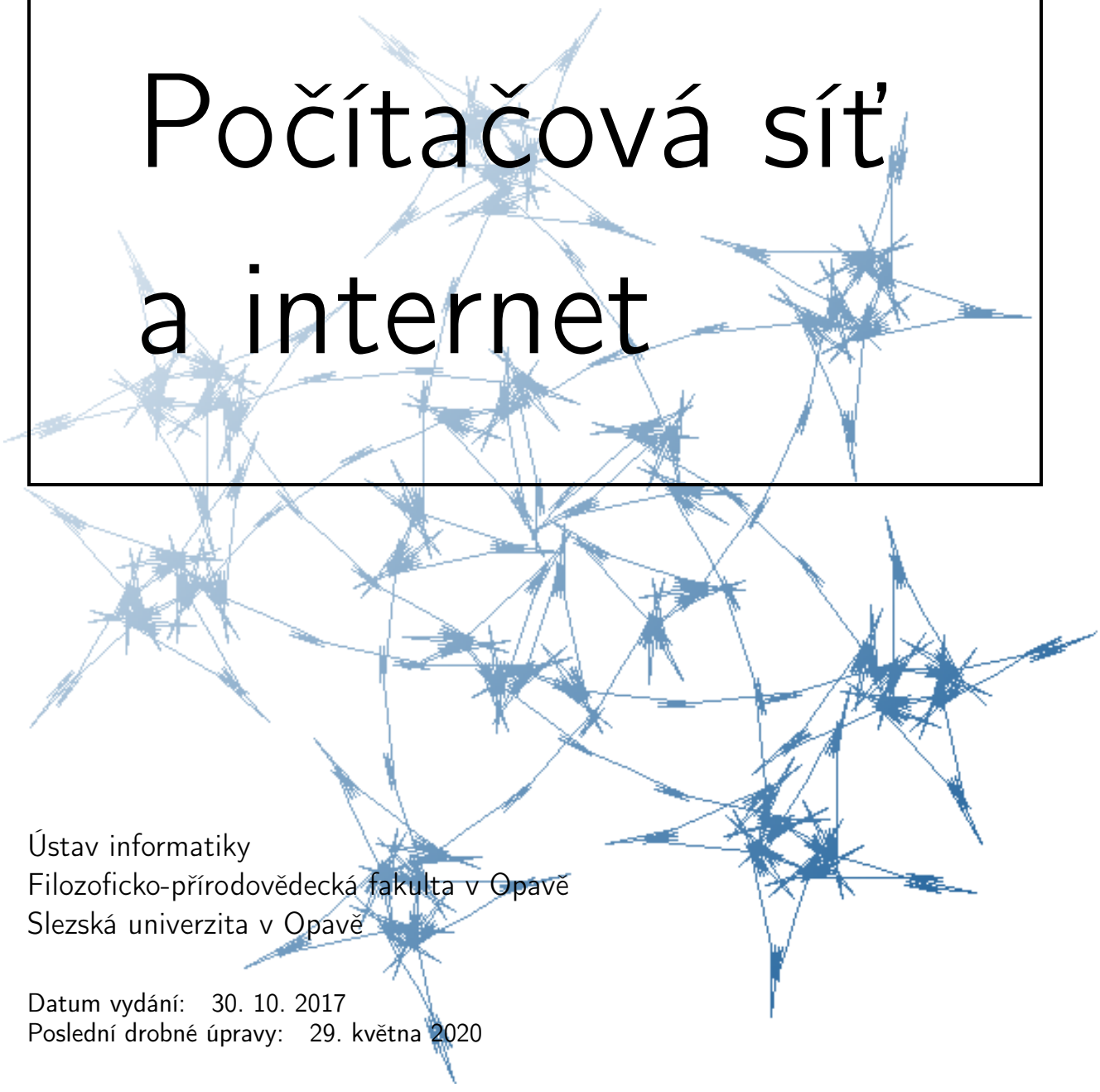
SLEZSKÁ UNIVERZITA
FILOZOFICKO-PŘÍRODOVĚDECKÁ FAKULTA V OPAVĚ

Šárka Vavrečková

Skripta do předmětu

Počítačová síť a internet



Ústav informatiky
Filozoficko-přírodovědecká fakulta v Opavě
Slezská univerzita v Opavě

Datum vydání: 30. 10. 2017
Poslední drobné úpravy: 29. května 2020

Anotace: Tato skripta jsou určena pro studenty předmětu *Počítačová síť a internet* na Ústavu informatiky Slezské univerzity v Opavě. Obsahují pouze učební látku pro přednášky, pro cvičení najdete další skripta. V předmětu se zabýváme prostředky a postupy používanými v počítačových sítích.

Počítačová síť a internet

RNDr. Šárka Vavrečková, Ph.D.

Dostupné na: http://vavreckova.zam.slu.cz/pocsit.html

Recenzenti: Ing. Magdalena Chmelařová, Ph.D.
Mgr. David Havrlant

Autorská práva: © RNDr. Šárka Vavrečková, Ph.D.

Vydavatel: © Filozoficko-přírodovědecká fakulta v Opavě,
Slezská univerzita v Opavě, 2017

ISBN: 978-80-7510-245-4

Sázeno v systému LaTeX

Předmluva

Co najdeme v těchto skriptech

Rychlý náhled: Tato skripta jsou určena pro studenty informatických předmětů na Slezské univerzitě v Opavě. Obsahují látku vyučovanou na přednáškách předmětu *Počítačová síť a Internet*, ve kterém se zabýváme (jak název napovídá) počítačovými sítěmi. Tato skripta obsahují převážně teorii (jak fungují zařízení připojená k síti, jakým způsobem je utvářena komunikace apod.), jejich doplněním jsou skripta pro cvičení z téhož předmětu (dostupná na stejném místě).

Budeme se zabývat především lokálními sítěmi (Ethernet, Wi-fi) a navazujícími mechanismy až po síťové služby jako je například DNS. Posledním probíraným tématem je problematika bezpečnosti, kde se seznámíme s principem VPN, síťovými analyzátory a procesem filtrování ve firewallu.

Některé oblasti jsou „navíc“ (jsou označeny ikonami fialové barvy), ty nejsou probírány a ani se neobjeví na zkoušce – jejich úkolem je motivovat k dalšímu samostatnému studiu či pokusům nebo pomáhat v budoucnu při získávání dalších informací. Pokud je fialová ikona před názvem kapitoly (sekce), platí pro vše, co se v dané kapitole či sekci nachází.

Značení

Ve skriptech se používají následující barevné ikony:

- *Rychlý náhled* (skript, kapitoly), ve kterém se dozvíme, o čem to bude.
- *Klíčová slova* kapitoly.
- *Cíle studia* pro kapitolu nám řeknou, co nového se v dané kapitole naučíme.
- Nové *pojmy*, značení apod. jsou značeny modrým symbolem, který vidíme zde vlevo.
- Konkrétní *postupy* a nástroje, způsoby řešení různých situací, do kterých se může správce počítačového vybavení dostat, atd. jsou značeny také modrou ikonou.
- Některé části textu jsou označeny fialovou ikonou, což znamená, že jde o *nepovinné úseky*, které nejsou probírány (většinou; studenti si je mohou podle zájmu vyžádat nebo sami

prostudovat). Jejich účelem je dobrovolné rozšíření znalostí studentů o pokročilá témata, na která obvykle při výuce nezbývá moc času.

- Žlutou ikonou jsou označeny odkazy, na kterých lze získat *další informace* o tématu. Nejčastěji u této ikony najdeme webové odkazy na stránky, kde se dané tématice jejich autoři věnují podrobněji.
- Červená je ikona pro *upozornění* a poznámky.

Pokud je množství textu patřícího k určité ikoně větší, je celý blok ohraničen prostředím s ikonami na začátku i konci, například pro definování nového pojmu:

Definice

V takovém prostředí definujeme pojem či vysvětlujeme sice relativně známý, ale komplexní pojem s více významy či vlastnostmi.

Podobně může vypadat prostředí pro delší postup nebo delší poznámku či více odkazů na další informace. Mohou být použita také jiná prostředí:

Příklad

Takto vypadá prostředí s příkladem, obvykle nějakého postupu. Příklady jsou obvykle komentovány, aby byl jasný postup jejich řešení.

Poznámka:

Poznámky zabírající více řádků jsou uzavřeny v tomto prostředí. Zde upozorňujeme na různé zajímavosti nebo zdůrazňujeme důležité fakty.

Obsah

Předmluva **iii**

1 Úvod do počítačových sítí **1**
1.1 Jak funguje počítačová síť . . . 1
1.1.1 Síťové služby a síťové technologie . . . 1
1.1.2 Co se nachází v počítačové síti . . . 2
1.1.3 Přenos dat . . . 4
1.1.4 Topologie . . . 6
1.2 Síťový hardware . . . 9
1.3 Co a jak přenášíme po počítačové síti . . . 11
1.3.1 Data a informace . . . 11
1.3.2 Protokoly: jak se elektroničtí cizinci navzájem domluví . . . 12
1.3.3 Vlastnosti protokolů . . . 13
1.4 Standardizace a modely počítačových sítí . . . 14
1.4.1 Standardizační instituce pro počítačové sítě . . . 15
1.4.2 Referenční model ISO/OSI . . . 18
1.4.3 Spolupráce protokolů . . . 22
1.4.4 Protokolové zásobníky . . . 25
1.4.5 Model TCP/IP . . . 26
1.5 Charakteristiky přenosu a přenosových cest . . . 29
1.5.1 Přenosový signál . . . 29
1.5.2 Přenos v základním a přeloženém pásmu . . . 30
1.5.3 Multiplex . . . 33

2 Lokální sítě – Ethernet **36**
2.1 Co je to Ethernet . . . 36
2.2 Komunikace v Ethernetu . . . 37
2.2.1 Typy zařízení v síti . . . 37
2.2.2 Propojení uzlů v síti . . . 37
2.2.3 Přístupová metoda . . . 38
2.3 Ethernet na linkové vrstvě . . . 40
2.3.1 Adresy na vrstvě L2 . . . 40
2.3.2 Podvrstvy L2 . . . 41

2.3.3 EtherType . . . 42
2.3.4 Formát ethernetového rámce . . . 43
2.3.5 Kolizní a všesměrová doména . . . 47
2.4 Ethernet na fyzické vrstvě . . . 48
2.4.1 Kabely . . . 48
2.4.2 Křížení a krimpování . . . 50
2.4.3 Parametry . . . 52
2.5 Průběh přenosu . . . 55
2.5.1 Přenos v poloviční duplexu . . . 55
2.5.2 Přenos v plném duplexu . . . 58
2.6 Technologie související s Ethernetem . . . 59
2.6.1 Autonegociace . . . 59
2.6.2 Napájení přes Ethernet . . . 59
2.6.3 Strukturovaná kabeláž . . . 60

3 Další témata k lokálním sítím 62
3.1 VLAN . . . 62
3.1.1 VLAN na jednom switchi . . . 63
3.1.2 VLAN rámce na cestě přes trunk . . . 64
3.1.3 Komunikace mezi různými VLAN sítěmi . . . 67
3.2 Switche a smyčky . . . 68
3.2.1 Protokol STP . . . 69
3.2.2 Konvergence sítě switchů . . . 71
3.2.3 Protokol STP a jeho varianty . . . 73
3.3 Přehled rodiny standardů IEEE 802 . . . 73

4 Síťová a transportní vrstva 76
4.1 Síťová vrstva a logické adresy . . . 76
4.2 Protokol IPv4 . . . 77
4.2.1 Adresy IPv4 . . . 77
4.2.2 Speciální adresy podle IPv4 . . . 79
4.2.3 IPv4 pakety . . . 80
4.2.4 Pole TTL a životnost paketu . . . 82
4.2.5 MTU a fragmentace IPv4 paketu . . . 82
4.3 Protokol IPv6 . . . 85
4.3.1 Adresy IPv6 . . . 86
4.3.2 Speciální adresy podle IPv6 . . . 87
4.3.3 IPv6 pakety . . . 88
4.4 Protokol ICMP a zprávy . . . 91
4.4.1 Účel protokolu ICMP . . . 91
4.4.2 ICMP verze 6 . . . 93
4.4.3 Testování dosažitelnosti . . . 94
4.5 Komunikace typu klient-server . . . 96
4.6 Transportní vrstva . . . 97
4.6.1 Čísla portů . . . 97

4.6.2 Protokoly transportní vrstvy . . . 98
4.6.3 Protokol TCP . . . 99
4.6.4 Průběh TCP spojení . . . 101
4.6.5 Protokol UDP . . . 104

5 Aplikační protokoly 106
5.1 DNS aneb překlad adres . . . 106
5.1.1 Domény a jmenné adresy . . . 106
5.1.2 Zóny a DNS servery . . . 108
5.1.3 Tabulka hostitelů . . . 110
5.1.4 Protokol DNS a DNS paket . . . 110
5.1.5 Databáze WHOIS . . . 112
5.2 Služba WWW a protokol HTTP . . . 112
5.2.1 Adresace . . . 112
5.2.2 HTTP zprávy . . . 113
5.2.3 Komunikace podle HTTP . . . 114
5.2.4 Informační kódy HTTP . . . 116
5.3 Služby elektronické pošty . . . 117
5.3.1 Infrastruktura . . . 117
5.3.2 Protokoly . . . 118
5.3.3 Komunikace a nastavení . . . 119
5.3.4 MIME . . . 120
5.4 Souborové služby . . . 120
5.4.1 Protokol FTP . . . 120
5.4.2 Sdílení prostředků v lokální síti . . . 122
5.5 Přidělování IP adres a protokol DHCP . . . 122
5.6 Další typy serverů . . . 124
5.7 Vzdálená konfigurace . . . 125
5.7.1 Telnet . . . 125
5.7.2 SSH . . . 127
5.8 Správa sítě . . . 128
5.9 Přehled protokolů a portů . . . 128

6 Síťové adresy a směrování 129
6.1 Adresy v rámcích a paketech . . . 129
6.2 Objevování sousedů . . . 131
6.2.1 Tabulky sousedů . . . 131
6.2.2 K protokolům . . . 132
6.3 Adresní rozsah IPv4 . . . 133
6.3.1 Třídy . . . 133
6.3.2 Podsíťování . . . 135
6.3.3 VLSM . . . 136
6.3.4 CIDR . . . 137
6.4 Jak získat IP adresu . . . 139
6.4.1 Jak získat IPv4 adresu . . . 139

6.4.2 Jak získat IPv6 adresu . . . 139
6.5 NAT a soukromé adresy . . . 141
6.6 Směrování . . . 144
6.6.1 Jak směrujeme . . . 144
6.6.2 Směrování algoritmem vektoru vzdáleností . . . 145
6.6.3 Směrování algoritmem stavu spoje . . . 146
6.6.4 Protokol BGP . . . 147

7 Bezdrátové sítě **148**
7.1 Bezdrátové technologie . . . 148
7.2 Wi-fi . . . 149
7.2.1 Access point . . . 150
7.2.2 Wi-fi na vrstvě L2 . . . 151
7.2.3 Přístupová metoda . . . 152
7.2.4 Fyzická vrstva . . . 153
7.2.5 Signál a antény . . . 158
7.3 Zabezpečení bezdrátové komunikace . . . 160
7.3.1 AAA . . . 160
7.3.2 WPS . . . 163
7.3.3 Jak tedy zabezpečit bezdrátovou síť . . . 164

8 Rozlehlé sítě a přístupové sítě **165**
8.1 WAN sítě . . . 165
8.1.1 Struktura WAN sítí . . . 165
8.1.2 Protokoly vrstvy L2 pro WAN sítě . . . 166
8.2 Nejznámější WAN sítě . . . 167
8.2.1 Frame Relay . . . 167
8.2.2 ATM . . . 168
8.2.3 MPLS . . . 169
8.3 Telekomunikační sítě . . . 171
8.4 Přístupové sítě . . . 172
8.4.1 ADSL . . . 173
8.4.2 VDSL . . . 175

9 Bezpečnost **176**
9.1 Bezpečnost v sítích . . . 176
9.1.1 Typy útoků . . . 176
9.1.2 Zabezpečení na jednotlivých vrstvách ISO/OSI . . . 179
9.2 VPN . . . 180
9.3 Síťový analyzátor . . . 183
9.4 Firewall . . . 184
9.4.1 Princip firewallu . . . 184
9.4.2 Typy filtrování . . . 185

Seznam doporučené literatury **189**

Kapitola 1

Úvod do počítačových sítí

Rychlý náhled: V této kapitole si osvětlíme základní pojmy týkající se počítačových sítí, v dalších kapitolách pak již budeme na těchto pojmech stavět. Seznámíme se s pojmy, kterými je popisován přenos dat, s topologiemi, síťovým hardwarem, uvedeme si značení (například pro jednotlivé síťové prvky), které bude dále používáno.

Důležitou částí této kapitoly je sekce o síťových standardech – celou dobu se budeme bavit o komunikaci, a pro komunikaci je standardizace nepostradatelná: obě komunikující strany potřebují mít implementovány tytéž standardy, aby se mohly navzájem „domluvit“.

Poslední sekce této kapitoly je věnována fyzikálním parametrům přenosových cest. Seznámíme se se základy práce se signálem, principem modulace a multiplexování.

Klíčová slova: Síťový prvek, spoj, kanál, okruh, ISP, PAN, LAN, MAN, WAN, duplex, simplex, datagramová služba, virtuální okruh, přenos, best effort, topologie, point-to-point spoj, páteřní síť, repeater, hub, switch, bridge, router, gateway, data, informace, protokol, standard, norma, ETSI, ITU, ISO, IEEE, IEC, IETF, IANA, ANSI, TIA, EIA, protokolová datová jednotka (PDU), referenční model ISO/OSI, protokolová sada, protokolový zásobník, TCP/IP, přenosový signál, nosný signál, frekvence, vlnová délka, amplituda, fáze, baseband, broadband, modulace, multiplex

Cíle studia: Po prostudování této kapitoly získáte přehled v základních pojmech z oblasti počítačových sítí a budete připraveni nastudovat témata probíraná v následujících kapitolách. Naučíte se orientovat ve vrstvách referenčního modelu ISO/OSI a síťového modelu TCP/IP. Porozumíte v základu způsobu přenosu signálu.

1.1 Jak funguje počítačová síť

1.1.1 Síťové služby a síťové technologie

Zatímco služba nám říká, co je poskytováno, technologie nám říká, jakým způsobem to funguje. Tabulka 1.1 ukazuje vztah mezi některými známými síťovými službami a k nim příslušejícími technologiemi.

Služba	Technologie
přístup k internetu	xDSL, WiMAX, GPRS, EDGE, LTE,...
internetová telefonie	VoIP, VoLTE,...
e-mail	SMTP, POP3, IMAP,...
WWW	HTTP, HTTPS,...
přenos souborů	FTP, SFTP,...
vzdálená správa	rlogin, telnet, SSH,...

Tabulka 1.1: Síťové služby a síťové technologie

Poznámka:
Předchůdcem Internetu coby největšího zprostředkovatele síťových služeb byl ARPANET (Advanced Research Projects Agency Network) spuštěný roku 1969. Byl to projekt amerického ministerstva obrany, experimentální síť financovaná agenturou DARPA (Defense Advanced Research Projects Agency). Původně šlo o sesíťování výkonných počítačů, ke kterým se dalo přistupovat vzdáleně. Z evropských států se k ARPANETu jako první připojilo Norsko (r. 1973). Provoz ARPANETu byl ukončen roku 1990, kdy již plně fungoval jeho potomek – Internet.

1.1.2 Co se nachází v počítačové síti

Dále budeme používat pojem počítačová síť, ale to je nepřesné, ve skutečnosti zde ani zdaleka nepůjde pouze o počítače (nicméně ten pojem je poměrně vžitý).

Intuitivně určitě chápeme, že v počítačové síti jde o propojení určitých zařízení tak, aby tato zařízení spolu mohla komunikovat. Jaká zařízení to vlastně chceme propojovat? Určitě počítač, notebook, tablet, smartphone, server, případně chytrou televizi či jiné „chytré zařízení“ – dnes už je chytré kdeco, dokonce existují i kuchyňské spotřebiče, které mohou být zapojeny do počítačové sítě. Tyto typy uzlů (na začátku nebo konci komunikační cesty) souhrnně nazýváme *klientská zařízení*. Dále do počítačové sítě patří síťové prvky, přes které komunikace probíhá:

- *aktivní síťové prvky* – aktivně ovlivňují komunikaci: směrují, zesilují signál apod.,
- *pasivní síťové prvky* – pouze „pasivně přenesou“ data, například kabeláž.

Definice (Počítačová síť)
Počítačová síť je soustava *uzlů* (node) vzájemně propojených *přenosovými cestami* (transmission path). Uzel v počítačové síti je buď klientské zařízení nebo aktivní síťový prvek.

V definici vidíme dva důležité pojmy – uzel a přenosová cesta. Pojem uzlu jsme si už v základu osvětlili, ale co je to přenosová cesta? Může jít o jedno konkrétní *přenosové médium*, třeba některý pasivní síťový prvek, třeba metalický kabel, nebo to může být třeba vzduch (u Wi-fi). Nebo může být přenosová cesta zkombinována z více různých přenosových médií.

Definice (Spoj, přenosový kanál, přenosový okruh)

Spoj (link) je obecným (abstraktním) pojmem označujícím používanou přenosovou cestu mezi danými uzly sítě.

Přenosový kanál (channel) je jednosměrná přenosová cesta určená nejen uzly, které propojuje, ale také fyzikálními charakteristikami – přenosovou rychlostí, šířkou pásma, úrovní šumu apod.

Přenosový okruh (circuit) je obousměrná přenosová cesta, může jít o dvojici přenosových kanálů (pro každý směr jeden kanál).

Přenosové okruhy coby komunikační cesty pro data jsou obvykle zřizovány organizací, která je pronajímá svým zákazníkům. Takový okruh nazýváme *pronajatý* (leased circuit).

Definice (Poskytovatel síťové konektivity, ISP)

Poskytovatel síťové konektivity (Network Connectivity Provider) je organizace poskytující službu přístupu do dané sítě. *ISP* (Internet Service Provider) je organizace poskytující přístup do Internetu (buď přímo nebo zprostředkovává službu jiného – nadřízeného – ISP).

Přenosové okruhy existují v těchto variantách:

- *fyzický (pevný) okruh* (physical circuit) – určený přímo fyzickou přenosovou cestou, není přerušen žádným meziuzlem, dnes se moc nepoužívají (jen pro soukromé zabezpečené cesty),
- *virtuální okruh* (virtual circuit) – logický okruh vedoucí přes různé sdílené fyzické přenosové cesty, může být dočasný, při opakovaném vytvoření může vést pokaždé přes jiné fyzické meziuzly, v jedné fyzické přenosové cestě může vést více virtuálních okruhů, podle doby trvání rozlišujeme
  - *pevný virtuální okruh* (PVC, permanent virtual circuit) – existuje stabilně po celou dobu pronájmu/vlastnictví, dlouhodobě,
  - *přepínaný virtuální okruh* (SVC, switched virtual circuit) – vytváří se dynamicky při potřebě komunikace, existuje tedy jen krátkodobě.

Pevné virtuální okruhy jsou výrazně dražší, ale na druhou stranu obcházejí meziuzly sítě (což znamená rychlejší komunikaci) a jsou odolnější vzhledem k těm poruchám, které souvisejí se sdílením okruhů. V současných rozlehlých sítích se však více používají přepínané virtuální okruhy, protože jsou levnější, pružnější a zbytečně neblokují kapacitu přenosové cesty, když okruh zrovna není používán.

Poznámka:

Dříve se okruhy přepojovaly ručně (spojovatelka v ústředně musela dva kontakty propojit drátem), dnes už je tato operace dokonce i v tradicionalistických telekomunikacích automatizovaná.

Podle rozlehlosti rozlišujeme tyto druhy počítačových sítí:

- PAN (Personal Area Network) – osobní síť na malém prostoru (například propojení zařízení pomocí Bluetooth, IrDA, USB),
- LAN (Local Area Network) – lokální sítě, rozlehlost v desítkách až stovkách metrů, obvykle v rámci jedné budovy (například Wi-fi, Ethernet),

- MAN (Metropolitan Area Network) – metropolitní sítě v rozsahu města nebo bloku budov (kilometry, desítky kilometrů), slouží k vzájemnému propojení různých LAN sítí (například WiMAX), často jde o přístupové sítě do Internetu,
- WAN (Wide Area Network) – rozlehlé sítě pokrývající region, stát, kontinent apod., slouží k vzájemnému propojení menších sítí (například různých LAN a MAN), plní podobnou roli jako WAN, ale na vyšší úrovni, Internet je také WAN sítí.

Jako přenosové médium je možné ve všech druzích používat metalické kabely nebo vzduch (rádiové spoje – elektromagnetické vlny), v LAN a rozlehlejších se můžeme setkat s optickými kabely (čím rozlehlejší, tím pravděpodobněji), infračervený přenos se používá v PAN sítích, v MAN sítích se můžeme setkat s laserovými spoji.

V tomto předmětu se budeme zabývat především LAN a WAN sítěmi.

1.1.3 Přenos dat

V síti nejde ani tak o samotné propojení uzlů, ale především o možnost přenosu dat mezi těmito uzly. Propojení (přenosová cesta, spoj) je pouze prostředkem, cílem je *přenos* (transmission): přenos probíhá vždy s využitím některého spoje.

Rozlišujeme přenos

- *simplexní* (simplex) – komunikace probíhá jen v jednom směru,
- *plně duplexní* (full duplex) – komunikace probíhá v obou směrech,
- *poloduplexní* (half duplex) – komunikace sice probíhá v obou směrech, ale ne zároveň (uzly se v komunikaci střídají, nemohou vysílat oba najednou).

Plně duplexní spoj může být realizován například dvěma nebo více simplexními spoji.

Většinou samozřejmě chceme komunikovat obousměrně, ale někdy to není technicky možné; některé přenosové cesty je třeba vyhrazovat, protože by při sdílení docházelo ke kolizím. U bezdrátových řešení, kdy je přenosovou cestou vzduch, se sdílení řeší multiplexováním: například přidělováním různých frekvencí nebo přidělováním času vysílání.

Přenos dále dělíme podle formy přenášených dat na

- *proudový* (stream) – přenáší se souvislý proud dat, který není třeba nijak vymezovat ani vnitřně členit,
- *paketový* (blokový, block, packet) – data jsou předem rozdělena (zabalena) do bloků o určité velikosti, opatřena „logistickými“ informacemi (adresy, typ obsahu, způsob zacházení po cestě apod.) a takto odesílána, v cíli jsou rozbalena a zkompletována do původní podoby.

Proudový přenos je typický například pro přenos hlasu po telekomunikační lince (po navázání spojení je přenášen audiosignál bez jakéhokoliv dělení) nebo v oblasti hardwaru například u HDMI (přenos multimediálního signálu). Paketový přenos je naopak typický pro počítačové sítě, v oblasti hardwaru u rozhraní DisplayPort.

O něco výše je definován pojem „přenosový okruh“. Proudový přenos úzce souvisí s okruhy – vždy probíhá tak, že je stanoven okruh, po kterém má přenos probíhat, případně je navázáno spojení přes tento okruh a data mohou začít „proudit“.

U paketového přenosu se nemusíme (ale můžeme) na konkrétní okruh vázat. Adresa nebo jiná identifikace cíle je součástí paketu, tedy pokud jsou data rozdělena do více paketů, může každý

tento paket jít jinou fyzickou cestou. Důležité je, že do cíle všechny pakety dorazí, také je stanoven postup, jak je seřadit (pakety mohou být doručeny v jiném pořadí než jak byly odeslány).

Z výše uvedeného vyplývá, že posíláme buď proud dat nebo jednotlivé pakety. Na hranici mezi dvěma proudy dat nebo dvěma pakety se provádí *přepojování* (přepínání; odeslali jsme jeden objekt, budeme odesílat další objekt). Rozlišujeme:

- *přepojování okruhů* (circuit switching) – celý proud dat je přenášen po tomtéž okruhu, který je třeba předem sestavit, data vždy dojdou ve správném pořadí,
- *přepojování paketů* (packet switching) – není nutné sestavovat přenosový okruh, pakety obsahují informace potřebné ke správnému nasměrování k cíli, paket jde tou cestou, která je v daném okamžiku považována za optimální, v cíli je nutné pakety správně seřadit.

Poznámka:

Okruhy jsou obdobou železniční sítě – lokomotiva i s vagóny jede po jedné konkrétní vyhrazené koleji, vagóny se cestou „nemíchají“. Přepojování *paketů* je jako silniční síť – větší skupina lidí může jet ve více automobilech, přičemž každý automobil může jet po jiné trase (každý řidič má vlastní představu o nejlepší trase), v cíli mohou být v jiném pořadí než v jakém vyjeli.

Zatímco telekomunikace zůstávají u „železnice“ (okruhy), počítačové sítě spoléhají spíše na „silnicích“ (pakety).

Přepojování paketů jako princip se poprvé objevilo v ARPANETu roku 1964 (Paul Baran), protože koncept přepojování okruhů běžný v telekomunikacích byl při pokusech uznán za nevhodný pro posílání dat: vyhrazení okruhu by znemožnilo tuto cestu používat komukoliv jinému a celkově by bylo používání okruhů nepružné. Ovšem samotný pojem „paket“ se používá až od roku 1965 (Donald Davies).

Podle toho, jestli je před samotným přenosem navazováno spojení, rozlišujeme přenos

- *spojovaný* (služba se spojením, connection-oriented) – nejdřív je navázáno spojení, následuje přenos, pak se ukončí spojení,
- *nespojovaný* (connectionless) – přímo před samotným přenosem se nenavazuje spojení.

Jaký je vztah mezi přepínáním okruhů/paketů a spojovanou/nespojovanou službou?

- Přepojování okruhů je vždy spojovaná služba. Vytvoření přenosového okruhu znamená vždy navázání spojení, zrušení okruhu po přenosu dat je ukončení spojení.
- Přepojování paketů může probíhat dvěma různými způsoby:
  - *datagramová služba* – nespojovaný způsob přenosu, samotný paket obsahuje plnou adresu příjemce,
  - *virtuální okruhy* – spojovaná služba, kdy jsou sice data rozdělena do paketů, ale předem je stanovena cesta pro jejich přenos, používají se přepínané virtuální okruhy (SVC).

Při využití virtuálních okruhů je v paketech jen krátká směrovací informace pro aktivní síťové prvky. Na síťových prvcích je pak ve speciální tabulce záznam o tom, kterým směrem má být paket s tou konkrétní informací poslán. Takže plná adresa příjemce vlastně nemusí být ani v paketu, ani na mezilehlých síťových prvcích (tam je jen lokální informace o tom, přes které síťové rozhraní vede k adresátovi cesta).

Během přenosu může dojít k poškození nebo ztrátě přenášených dat (například vlivem rušení, přeslechů, slabého signálu). S tím se běžně setkáváme například u Wi-fi: při zahlcení sítě se přenos zpomalí nejen tím, že síťové prvky mohou při detekování problémů snížit rychlost, ale i tím, že poškozené pakety je třeba poslat znovu.

Podle spolehlivosti členíme přenos na

- *spolehlivý* (reliable) – pokud je detekována chyba při přenosu (poškozený nebo chybějící paket), vyžádá se jeho opakování (například poškozený paket je znovu poslán),
- *nespolehlivý* (unreliable) – poškozený paket je jednoduše zahozen, chybějící ignorován.

Poznámka:

Pozor – nespolehlivá služba ještě neznamená, že data do cíle nemusejí vůbec dorazit! Znamená to jednoduše, že o ošetření chyb se prostě postará někdo jiný. Síťové technologie jsou typické tím, že jejich činnost je rozdělena do vrstev, kde každá vrstva plní určitý úkol, a pokud jedna vrstva poskytuje nespolehlivou službu, jiná vrstva může zajistit ošetření chyb.

Zajištění spolehlivé služby (s ošetřením chyb) je sice výhodné z pohledu výskytu chyb, ale na druhou stranu je taková služba časově náročná. Proto se nespolehlivý přenos používá všude tam, kde buď sem tam ztracený paket moc nevadí nebo se chybami zabývá jiná vrstva.

Definice (Princip Best Effort)

Princip *Best Effort* (nejlepší snaha) řeší nespolehlivost služby následovně:

- *je maximální snaha* paket doručit,
- *není zaručen výsledek* – když není dostatek zdrojů, přenosová kapacita apod., k doručení nedojde.

Maximální snaha znamená, že dokud je dostatek zdrojů, budou pakety doručovány, ale když zdroje dojdou, budou pakety bez rozlišování zahazovány, dokud se nenajdou nové zdroje. Tento princip se zdaleka netýká jen doručování paketů, ale obecně hospodaření s omezenou množinou zdrojů.

Podle množství adresátů rozlišujeme komunikaci

- *unicast* – posílaná data jsou určena jen jedinému adresátovi,
- *multicast* – data jsou určena *skupině* adresátů (všem patřícím do konkrétní skupiny),
- *broadcast* (všesměrová komunikace) – data jsou určena všem připojeným,
- *anycast* – data jsou určena jednomu (kterémukoliv) adresátovi z konkrétní skupiny.

1.1.4 Topologie

Pojem topologie obvykle souvisí s prostorem a vztahy (uspořádáním) mezi objekty v prostoru.

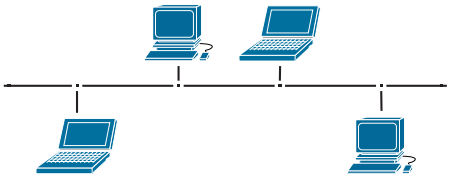
Definice (Fyzická a logická topologie sítě)

V počítačových sítích rozlišujeme fyzickou a logickou topologii:

- fyzická topologie určuje, jak jsou jednotlivé uzly sítě navzájem propojeny,
- logická topologie určuje, jak tyto uzly navzájem komunikují.

U topologie není důležitá délka cest, propustnost ani použitá technologie, jde o způsob fyzického nebo logického propojení uzlů.

V současné době už pojem topologie není až tak důležitý jako v minulosti, nicméně se s topologiemi občas setkáváme, proto si představíme několik nejběžnějších.

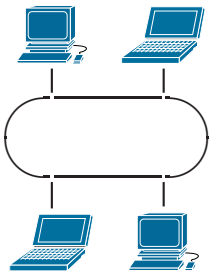


Topologie sběrnice. Přenosová cesta je sdílená všemi zapojenými zařízeními, nepotřebujeme centrální prvek. Hovoříme také o segmentu, případně páteři (to je kabel, ke kterému jsou všechna zařízení připojena). Signál vyslaný některým zařízením se šíří oběma směry, může být přijat kterýmkoliv jiným připojeným zařízením. Obvykle je třeba oba konce páteře ukončit *terminátorem*, který pohlcuje signály, aby nedocházelo ke zpětnému šíření a tím rušení další komunikace. Zařízení jsou k hlavnímu kabelu připojena pomocí *T-kusů* (T-spojek, „téček“).

Výhodou je jednoduchost a nižší spotřeba kabelu, stačí co nejefektivněji vést kabel mezi zařízeními. Pokud dojde k poškození kabelu, je síť rozdělena na dvě části, a pokud je dočasně vyřešeno případné zpětné šíření signálu od místa poškození, pak se v rámci dvou takto vzniklých segmentů dá dále komunikovat (ale ne mezi nimi).

Nevýhodou je příliš velké zatížení linky – všechna zařízení sdílejí tutéž přenosovou cestu a každý vyslaný signál se šíří po celé této cestě.

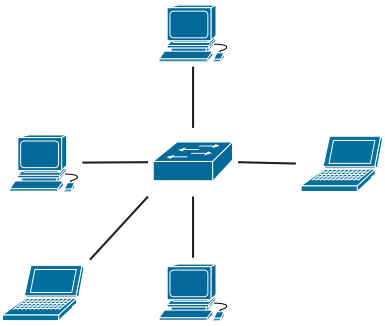
V současné době se tato topologie jako fyzická v počítačových sítích ani nepoužívá, setkáme se s ní spíše u kabelové televize, na ni navázaného „kabelového internetu“ a u některých hardwarových rozhraních v PAN sítích (SCSI). Dříve se používala u Ethernetu (na koaxiálním kabelu). Jako logická topologie se používá u starších pomalejších generací Ethernetu.



Kruhová topologie. je vpodstatě obměnou topologie sběrnice se spojenými konci, vlastnosti této topologie jako fyzické jsou podobné jako vlastnosti sběrnice, včetně způsobu šíření signálu, ale jako logická topologie funguje trochu jinak – může být například využit princip „peška“ jako speciálního paketu povolujícího vysílání, obíhajícího v kruhu, čímž je jednoznačně určeno, které připojené zařízení může v daném okamžiku vysílat.

Výhody a nevýhody jsou podobné jako u sběrnice.

Dříve se tato topologie používala u sítí Token Ring (lokální) a FDDI (WAN).



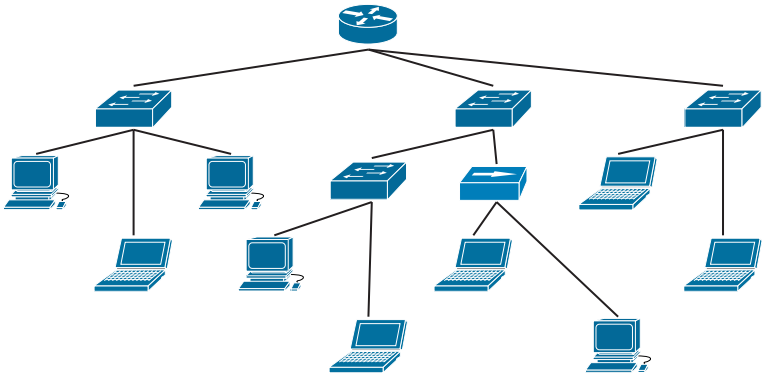
Topologie hvězda. V síti je jeden centrální prvek, k němuž jsou připojeny ostatní uzly sítě. Veškerá komunikace probíhá přes tento centrální prvek.

Výhodou je, že pokud je poškozen spoj mezi některým uzlem a centrálním prvkem, síť funguje dál (jen ten jeden uzel nemůže komunikovat). Další výhodou je možnost oddělení komunikačních cest mezi jednotlivými dvojicemi uzlů (oddělení zajišťuje na logické úrovni centrální prvek).

Nevýhodou je, že centrální prvek je „úzkým hrdlem komunikace“ (tj. jakákoliv komunikace musí jít přes něj, je nejvíc vytěžován), dále poškození nebo přetížení centrálního prvku znamená kolaps celé sítě. Nevýhodou je také větší spotřeba kabeláže (ve srovnání s topologií sběrnice).

Tato (fyzická) topologie je základem současných lokálních sítí.

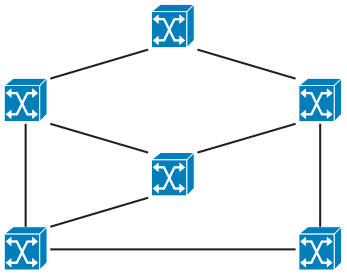
Stromová topologie. Je to zobecnění topologie hvězda. Zařízení jsou rozdělena do úrovní a počítá se s oddělením provozu v jednotlivých částech sítě; to neznamená, že by uzly v různých částech nemohly spolu komunikovat, jen se v případě, že zrovna spolu komunikovat nechtějí, navzájem neruší.



Výhodou je usnadnění správy sítě a malé riziko zahlcení spoje, nevýhodou je vyšší spotřeba kabelů (především pokud se budování sítě provádí živelně bez plánu). Při výpadku některého z vnitřních uzlů se jeho podstrom rozpadne.

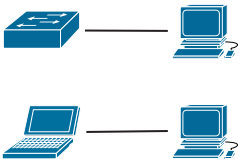
Se stromovou topologií se dnes běžně setkáváme ve středních a velkých lokálních sítích, je základem *strukturované kabeláže*. Typickým zástupcem je Ethernet.

Topologie mesh (smíšená). Pro tuto topologii jsou typické *redundantní spoje* (tj. mezi dvěma uzly může vést více než jedna cesta). Uzly v síti jsou víceméně rovnocenné (jen na okrajích mohou být jiné typy uzlů, které zprostředkovávají komunikaci vnitřních uzlů se „zbytkem světa“). Pro tyto sítě je typická nejen redundance, ale také velké množství uzlů, proto je hledání cesty v síti náročnější.



Výhodou je vysoká odolnost proti výpadku, nevýhodou vysoká spotřeba kabeláže.

Tento typ topologie je typický pro současné WAN sítě (MPLS apod.). V nich je třeba zajistit spolehlivost cest, přičemž provoz je v průměru velký a proměnlivý, tedy redundance je nezbytná.



Point-to-point spoj. Jednoduché propojení dvou zařízení. Jako fyzická topologie je použitelná v případě, že opravdu potřebujeme propojit pouze dvě zařízení, ale jako logická topologie je v současné době hodně používaná – pokud například u fyzické topologie hvězda použijeme jako centrální prvek takové zařízení, které dokáže na logické úrovni oddělit komunikaci s jednotlivými připojenými zařízeními, pak je logickou topologií množina point-to-point spojů (každé koncové zařízení „vidí“ jen centrální prvek).

Páteřní vedení (backbone). Páteř je část sítě propojující její jednotlivé části. Předpokládá se, že přímo na páteř jsou napojeny aktivní prvky sítě schopné oddělovat provoz (poznají jestli mají dotyčný paket poslat do „své“ sítě nebo pryč na páteř). U větších sítí je páteř víceúrovňová (jednotlivé úrovně – tier – jsou hierarchicky uspořádány). S páteří jsme se ve skutečnosti setkali u stromové topologie.

Poznámka:

Také Internet má svou páteř. Ze začátku ji tvořil její předchůdce ARPANET, v druhé fázi se páteří stalo několik sítí komerčních ISP. Dnes je struktura Internetu postavena na třech úrovních:

- Tier 1 (Backbone Providers) – ISP organizace většinou nadnárodního dosahu, které vlastní nejdůležitější části internetové páteře a mají pod kontrolou směrování provozu na této páteři,

- Tier 2 (National Providers) – ISP národního dosahu, kteří provozují svou národní část páteře Internetu, platí některému ISP z Tier 1 za přístup na hlavní páteř a provozují přístupové body na tuto páteř (POP – Point of Presence),
- Tier 3 (Local Providers) – v rámci země je (mělo by být) více těchto ISP, kteří si navzájem konkurují, nakupují konektivitu od svého národního poskytovatele z Tier 2 a nabízejí služby (přístup k Internetu) koncovým zákazníkům (firmám, běžným uživatelům, státním institucím apod.). Pro své zákazníky provozují přístupovou síť, což je třetí úroveň páteře Internetu.

Další informace:
http://www.slideshare.net/abdurrehmanabdurrehman391/iap-03-isp-hierarchy-types-of-delays

1.2 Síťový hardware

Každé zařízení připojené do sítě musí mít nejméně jedno *síťové rozhraní*, tedy komponentu, přes kterou dokáže komunikovat se sítí (přijímat pakety nebo je odesílat). U běžného počítače, notebooku apod. jde jednoduše o *síťovou kartu*.

Síťové rozhraní může být integrováno na základní desce zařízení (v poslední době to je poměrně běžné, zejména pro malá zařízení) nebo to může být dedikovaná karta (pro zasunutí do slotu PCI nebo PCIe) nebo externí (připojujeme většinou přes USB).

Síťová karta (NIC, Network Interface Card) je rozšiřující karta poskytující síťové rozhraní.

Obrázek 1.1: Síťové karty – dvě pro Etnernet, dvě pro Wi-fi (Zdroj: Heureka.cz)

Na obrázku 1.1 jsou čtyři různé síťové karty pro zapojení běžného počítače do lokální sítě. První karta je dedikovaná do slotu PCIe a umožňuje komunikovat přes síť Ethernet, druhá je externí pro USB a také je pro síť Ethernet. Zbývající dvě síťové karty umožňují komunikovat přes bezdrátovou síť Wi-fi, a také je jedna dedikovaná do PCIe a druhá externí do USB.

První dvě karty na obrázku 1.1 mají fyzický *port*, do kterého zasouváme konektor kabelu, protože Ethernet je síť využívající kabely. Další dvě karty žádný fyzický port nemají, protože Wi-fi je bezdrátová síť, tedy nepotřebujeme žádné kabely připojovat. Pokud je síťová karta integrovaná a potřebuje fyzické porty, najdeme je v případě klasického počítače na back panelu (vzadu), u notebooků taky vzadu nebo na boku.

Servery mají podobné síťové karty, jen obvykle s více než jedním portem (buď záložní spoj, nebo paralelní komunikace nebo pro rozšíření komunikačního pásma), taky se u nich můžeme setkat i s jinými plugy/rozhraními než u běžných počítačů (například Fibre Channel).

Aktivní síťové prvky, tedy na taková zařízení, která nejsou koncová (komunikace jde přes ně, nikoliv pouze od nich nebo jen k nim), samozřejmě taky potřebují síťová rozhraní. Obvykle jich však mívají mnohem víc než běžné počítače nebo servery, protože pracují jako zprostředkovatelé komunikace, mezilehlá zařízení propojující jednotky, desítky,... jiných zařízení. Protože práce se sítí je hlavní náplní jejich činnosti, jsou jejich síťová rozhraní typicky integrovaná.

Existují různé aktivní síťové prvky, které se liší už svými základními vlastnostmi. Na obrázcích v sekci s topologiemi jste si určitě všimli, že pro mezilehlá zařízení je používáno několik různých ikon. Ty nejdůležitější aktivní síťové prvky si dále představíme.

Repeater (opakovač) je jednoduchý aktivní síťový prvek se dvěma porty. Příchozí signál zesílí (případně znovu vygeneruje) a pošle dál. Dnes se s repeatery setkáváme spíše u bezdrátových lokálních sítí, ale také např. HDMI, kde slouží ke zvýšení dosahu signálu.

Hub (rozbočovač) je jednoduchý aktivní síťový prvek, který má typicky více než dva porty. Cokoliv přijde na některý port, jen přepošle na všechny ostatní porty (tento signál zesílí nebo znovu vygeneruje). Stejně jako repeater, i hub pracuje pouze se signálem, nedokáže se podívat „dovnitř“ paketu. Každý port se taky chová jako repeater – zesílí nebo znovu vygeneruje signál.

Jeho výhodou je rychlost (nic složitého nedělá), nevýhodou je, že zbytečně zahlcuje síť (paket pošle všem kromě odesílatele). V běžných počítačových sítích se už dnes s huby moc nesetkáváme, používají se ale například v některých PAN sítích nebo u USB.

Switch (přepínač) je již trochu inteligentnější aktivní síťový prvek, který si vede tabulku adres uzlů (ke každému uzlu, který „zná“, má poznamenáno, přes který port je tento uzel dosažitelný). U příchozího paketu zjistí adresáta, podle tabulky určí port, který k němu vede, a paket odešle jen na tento port. Pokud adresáta „nezná“ (nemá ho v tabulce) nebo když je paket určen všem (broadcast), odešle ho na všechny porty. Říkáme, že směrovač rozděluje síť na segmenty – pokud mezi sebou komunikují uzly ze stejného segmentu, pak taková komunikace nepronikne do jiného segmentu.

Výhodou je, že tolik nezahlcuje síť, také je možné implementovat správní a bezpečnostní funkce. Nevýhodou je výpočetně náročnější provoz (což je technicky řešitelné). Se switchi se běžně setkáváme v LAN, MAN i WAN sítích.

Bridge (most) je zařízení navzájem oddělující dva segmenty sítě, podobně jako switch. S „čistým“ mostem se však moc nesetkáváme – běžnější jsou switche, které propojují více než jen dva segmenty sítě, navíc dražší switche mívají rozšířenou funkcionalitu (například některé funkce typické pro routery).

Router (směrovač) je ještě složitější aktivní síťový prvek. Taky si vede tabulku (říkáme jí *směrovací tabulka*), v ní však obvykle nemá adresy konkrétních uzlů, ale spíše adresy sítí, ve kterých se nacházejí uzly. Příchozí paket „pitvá“ o něco důkladněji než switch (o tom podrobněji dále), přičemž podle cílové adresy zjistí, do které sítě patří adresát, ve směrovací tabulce ověří, který port je cestou do této sítě, a na ten port paket odešle. Taky má stanovený port, na který

odesílá uzly patřící do neznámých sítí. Broadcast pakety, tedy pakety adresované všem, zahazuje (přijme, ale nikam nepošle, ignoruje).

Zatímco switch odděluje segmenty jedné sítě, router odděluje různé sítě (a taky samozřejmě umožňuje jejich vzájemnou komunikaci, přičemž se navzájem nezahlcují).

Výhodou routeru je možnost implementace pokročilých správných a bezpečnostních funkcí a schopnost oddělit komunikaci v různých sítích. Nevýhodou je ještě více výpočetně náročný provoz než u switche, proto typicky routery mají méně portů než switche, výkonnější procesor a více paměti, aby vyšší výpočetní zátěž unesly.

Brána (gateway) slouží k propojení dvou různých typů sítí, resp. slouží jako spojovací bod a zároveň „překladatel“. Například pokud máme v domácnosti VDSL router, pravděpodobně má vestavěnou bránu pro komunikaci mezi naší lokální sítí a VDSL sítí poskytovatele internetu.

Obrázek 1.2: Příklady switchů (D-Link a Cisco) a routerů (Cisco a HP, zadní strany) (Zdroj: Heureka.cz)

1.3 Co a jak přenášíme po počítačové síti

1.3.1 Data a informace

Lidé většinou počítají v dekadické (desítkové) soustavě, protože mají deset prstů, ale pro počítače je přirozenější počítat spíše v binární (dvojkové) soustavě, protože elektronické součástky se mohou vyskytovat v jednom ze dvou stavů – vypnuto/zapnuto, neprochází proud/prochází proud. Také data, která posíláme počítačovou sítí, jsou ve skutečnosti v binárním formátu.

Zatím jsme pro to, co je přenášeno sítí, používali pojem data, což je ve většině případů v pořádku. Konkrétnějším pojmem je však informace.

Definice (Informace)

Informace jsou data, která snižují nebo odstraňují neurčitost systému. Informace lze přijímat, odesílat, uchovávat a zpracovávat.

Množstvím informace rozumíme rozdíl mezi stavem neurčitosti systému před a po přijetí dané informace.

To, do jaké míry jsou data informací, je relativní – záleží na posuzovateli a na tom, jak moc jsou konkrétně pro něj dotyčná data „nová“, „užitečná“.

Jednotkou množství informace je *bit*, který může nabývat jedné ze dvou hodnot – 0 nebo 1. Odvozenou jednotkou je *Byte*, který obvykle zabírá 8 bitů.

Také se používají předpony určující násobné množství – buď podle soustavy SI (násobnost $1000 = 10^3$) nebo binární násobky (násobnost $2^{10} = 1024$). Binární násobky jsou standardizovány jako IEC 60027-2, tento standard je převzat jako česká norma ČSN IEC 60027-2.

Název	Značka	Hodnota v B
Kibibyte	KiB	$1\,\text{KiB} = 2^{10}\,\text{B} = 1024\,\text{B}$
Mebibyte	MiB	$1\,\text{MiB} = 2^{10}\,\text{KiB} = 1024\,\text{KiB} = 2^{20}\,\text{B}$
Gibibyte	GiB	$1\,\text{GiB} = 2^{10}\,\text{MiB} = 1024\,\text{MiB} = 2^{20}\,\text{KiB} = 2^{30}\,\text{B}$
Tebibyte	TiB	$1\,\text{TiB} = 2^{10}\,\text{GiB} = 1024\,\text{GiB} = 2^{20}\,\text{MiB} = 2^{30}\,\text{KiB} = 2^{40}\,\text{B}$
atd.		

Tabulka 1.2: Binární násobky

V tabulce 1.2 je ukázán význam přípon používaných pro násobnost (v případě jednotky Byte, pro bit by to bylo samozřejmě podobné). Na rozdíl mezi násobností podle SI a binární je třeba si dávat pozor – například:

- $1\,\text{MB} = 1\,000\,000\,\text{B}$
- $1\,\text{MiB} = 1\,048\,576\,\text{B}$

To znamená, že 1 MiB je o 48 576 Bytů větší, což je rozdíl téměř 5 %.

Poznámka:

Jak je výše uvedeno, Byte mívá *obvykle* 8 bitů. Proč *obvykle*? Protože záleží na konkrétní hardwarové architektuře. Jeden Byte může mít 6, 7, 8 nebo 9 bitů. Většina těchto výjimek se vztahuje k hlubové historii, ale například dodnes se můžeme setkat se 7bitovým kódováním v některých e-mailech psaných v anglicky mluvících zemích.

V informatice je preferován jeden důležitý princip – jednoznačnost. Potřebujeme se vyjadřovat jednoznačně, přesně, tak, aby bylo vždy jasné, o čem a o jakém množství mluvíme. Proto kromě pojmu Byte existuje i jeho jednoznačnější ekvivalent: oktet.

Oktet je jednotka množství informace představující 8 bitů. Význam tohoto pojmu se promítl i do jeho názvu – v latině se „osm“ řekne „octo“.

Ve většině případů jsou tedy termíny Byte a oktet synonyma, ale pojem oktet je vždy jednoznačný, proto je především v oblasti počítačových sítí používán více. Dokonce je možné k němu přidružovat předpony znamenající násobky – například kilooktet (ko) nebo kibioktet (Kio), ale s tím se už zdaleka tolik nesetkáváme, ve zkratkovém zápisu a s násobnostmi se používá spíše pojem Byte.

1.3.2 Protokoly: jak se elektroničtí cizinci navzájem domluví

V počítačové síti můžeme propojit i poměrně hodně odlišná zařízení – nejde jen o to, že na jednom počítači běží Windows v určité verzi, na další Windows v jiné verzi, pak je tu nějaký Linux, server se Solarisem, MacOS X apod., na aktivních síťových prvcích často máme Linux (opravdu!), IOS (ne ten od Applu, ale od Cisca) nebo něco úplně jiného… Zařízení v síti se mnohem více než softwarem liší svým hardwarem. Jak se vlastně takoví „cizinci“ dokážou navzájem domluvit?

Ve světě lidí se dorozumíváme některým jazykem – česky, anglicky, německy, čínsky,…, a dokážeme si popovídat s každým, kdo buď ovládá některý jazyk, který ovládáme i my, anebo existuje ochotný překladatel ovládající některý náš jazyk a některý jazyk našeho partnera pro komunikaci.

Jazyk určuje, jak je co pojmenováno, jak máme pozdravit na začátku komunikace, jakým způsobem máme druhé straně sdělit určitou informaci, jak nám na toto sdělení druhá strana odpoví, jak se dozvíme, že nám ten druhý nerozumí, jak máme rozhovor ukončit a rozloučit se.

Ve světě elektroniky jsou místo jazyků *protokoly*. Protokol určuje, jakým způsobem má komunikace začít, případně jak se dohodnout na určitých parametrech komunikace, jak sdělit druhé straně určitý typ informace, jak má druhá strana potvrdit příjem nebo sdělit, že je třeba přenos opakovat, jak se komunikace ukončuje, atd.

Definice (Protokol a jeho implementace)

Protokol je konvence, podle které probíhá určitý typ (většinou elektronické) komunikace. Definuje pravidla určující syntaxi (jak jsou které signály poskládány), sémantiku (význam) a synchronizaci komunikace.

Protokol je pouze předpis, který je třeba implementovat (naprogramovat, aby mohl být v praxi používán). Implementace může být softwarová, hardwarová nebo kombinace softwarové a hardwarové.

Například protokol HTTP (kromě jiného) určuje, jak se má „bavit“ webový prohlížeč s WWW serverem. Je implementován (naprogramován) v operačním systému na počítači, u kterého sedíte, a taky na druhé straně – v operačním systému WWW serveru, se kterým komunikujete.

Poznámka:

Pro návrh protokolu platí jedno důležité pravidlo (vlastně je zachováváno i jinde, například u systému UNIX): protokol by měl být jednoduchý, krátký, jednoznačně implementovatelný. Neměl by být moc složitý, protože čím větší složitost, tím větší pravděpodobnost chyb. Proto žádný protokol není moc univerzální – umí jednu konkrétní věc, a umí ji dobře. V angličtině se to vyjadřuje zkratkou KISS (Keep it Simple, Stupid – ať je to jednoduché, hloupé).

Aby tento princip mohl být dodržován, existují určité konvence pro spolupráci protokolů: to, co protokol neumí, předá ke zpracování jinému protokolu.

Princip protokolů není ani zdaleka jen záležitostí počítačových sítí – protokoly se používají v telekomunikacích, spotřební elektronice, strojírenství, ale například i „uvnitř“ počítače. Každý se setkal třeba s protokolem USB, SATA,...

1.3.3 Vlastnosti protokolů

Jak víme, běžně používané protokoly mají obvykle (kromě jiných) tyto tři důležité vlastnosti:

- jsou všeobecně známé a (téměř) kdokoliv je může implementovat,
- jsou spíše jednoduché, nepříliš komplexní,
- mohou spolupracovat s jinými protokoly.

Nejdřív se zaměříme na první vlastnost. Proč je důležitá? Představme si situaci, kdy výrobce síťového hardwaru přijde s novým zařízením a rozhodne se, že toto zařízení bude komunikovat podle nového protokolu, jehož specifikaci nikomu jinému neposkytne. Toto zařízení si ovšem bude

„rozumět“ jen s takovými zařízeními, která budou podporovat tentýž protokol, takže nedokáže komunikovat se zařízeními od jiných výrobců. Pro dotyčného výrobce by to snad mohlo být výhodné, pokud by dokázal své potenciální zákazníky přesvědčit, aby kupovali jenom od něj, ale realita bývá jiná – zákazníci by radši kupovali taková zařízení, která by mohli bez problémů zařadit do své sítě, ve které už pravděpodobně mají nějaká zařízení od jiných výrobců.

Proto je většina protokolů buď volně dostupná (specifikace je zcela volně k nahlédnutí, implementovat může kdokoliv) nebo alespoň dostupná jiným způsobem (specifikace může být poskytnuta za poplatek, udělení licence), případně je specifikace sice dostupná, ale za její použití (implementaci) je třeba platit formou nákupu licence.

Otevřený protokol je protokol zcela volně dostupný, naopak *proprietární protokol* je takový protokol, jehož specifikace není nikde zveřejněna a jeho tvůrce si ji buď nechává jen pro sebe nebo licencuje vybraným obchodním partnerům za poplatek.

Poznámka:
O rozšířenosti volných specifikací mluví například to, že momentálně je na routerech nejběžnějším směrovacím protokolem otevřený protokol OSPF. Naopak IGRP (proprietární směrovací protokol společnosti Cisco) se téměř nepoužívá dokonce ani na zařízeních od samotného Cisca.

Na druhou stranu – Skype je taky proprietární, ale to na jeho popularitě neubírá (nicméně se nejedná o protokol, na němž by stálo fungování sítí).

Pokud síťové zařízení podporuje některý proprietární protokol, obvykle pro danou funkci taky podporuje některý alternativní protokol s dostupnější specifikací, aby si mohlo „popovídat“ se zařízeními jiných výrobců.

Zaměřme se teď na druhou a třetí vlastnost – jednoduchost a schopnost spolupráce s jinými protokoly. O jakémkoliv produktu (ať už hardwarovém nebo softwarovém) platí, že čím je složitější, tím větší je pravděpodobnost chyb a tím náročnější je takový produkt vytvořit. V oblasti technologií jde vývoj velmi svižně kupředu, takže i časové hledisko vytváření nebo aktualizace produktů je důležité, o bezpečnosti nemluvě. Navíc – různé typy zařízení mají určitou množinu funkcí společnou, liší se jen v něčem. Proto je běžné vytvářet menší úseky kódu (programů, protokolů, případně u hardwaru kusy součástek) a pak je vhodně skládat do požadovaného celku.

Dále se k principu otevřenosti, jednoduchosti a spolupráce budeme často vracet.

1.4 Standardizace a modely počítačových sítí

Definice (Standard, norma a standardizace)
Standard (v oblasti technologií) je požadavek na splnění určitých konkrétních vlastností pro určitý typ hardwaru, softwaru apod. Je to dokument popisující požadavky, specifikace, návody a popisy pro daný produkt, proces či službu. Rozlišujeme standardy

- *normativní (de iure, podle zákona)* – jejich dodržování je vyžadováno, obvykle je stanoví příslušná státní instituce,
- *popisné (de facto)* – jejich dodržování sice není striktně vyžadováno, ale je v zájmu výrobce.

V oblasti technologií obvykle používáme pojem *norma* pro normativní standardy, jinak prostě použijeme pojem *standard*, třebaže se taky často mluví o *doporučeních* (recommendation).

Standardizace je proces sjednocení vlastností a funkcí daného produktu pomocí stanovení standardu.

1.4.1 Standardizační instituce pro počítačové sítě

Následuje přehled nejdůležitějších standardizačních organizací a institucí, které vydávají standardy související s počítačovými sítěmi a technologiemi obecně. Většinou se jedná o standardy de facto (nejsou závazné, ale běžně bývají dodržovány), samozřejmě až náš vlastní (a částečně i evropský) normalizační institut.

Situace se mírně komplikuje tím, že některé standardy vydané jednou organizací bývají často adaptovány jednou či dokonce více dalšími organizacemi (například je běžná adaptace standardů českým normalizačním úřadem). Původce standardu obvykle poznáme podle začátku označení tohoto standardu, kombinaci více organizací pak díky kombinaci jejich označení.

V České republice se tvorbou českých norem zabývá *Úřad pro technickou normalizaci, metrologii a státní zkušebnictví*. Normy vydané tímto ústavem poznáme podle toho, že začínají písmeny ČSN, za nimi následuje číslo normy. V současné době ve značné míře zabývá adaptací (přejmutím a přizpůsobením) norem vydaných Evropskou unií nebo některou standardizační organizací. Tyto normy poznáme podle toho, že za ČSN následuje zkratka původní organizace či instituce, například ČSN EN... jsou normy přejaté od Evropské unie.

Další informace:

Normy vydané Úřadem pro technickou normalizaci, metrologii a státní zkušebnictví jsou k nahlédnutí na http://www.unmz.cz/urad/csn-online (jen náhledy, za celé znění se platí).

ETSI (European Telecommunications Standard Institute) je sice původně evropská organizace, ale ve skutečnosti jsou její členové ze všech obydlených kontinentů (státy, významní výrobci komunikačních zařízení, poskytovatelé síťových služeb, výzkumné organizace, atd.). Náměty na nové standardy nebo změny stávajících vycházejí ze tří zdrojů – buď se domluví nejméně čtyři členové ETSI, nebo je přijat podnět od Evropské komise (EC, European Commission) nebo od EFTA (European Free Trade Association).

Rozlišuje se několik typů ETSI standardů – například EN (European Standard), ES (ETSI Standard), TS (ETSI Technical Specification) a další. K nejznámějším patří standardy související s mobilními sítěmi (především GSM, 3G, 4G sítě), chytrými sítěmi, komunikací machine-to-machine, ICT ve zdravotnictví, atd. ETSI standardy jsou dostupné volně bez poplatků.

Další informace:

https://portal.etsi.org

ITU (International Telecommunications Union, Mezinárodní telekomunikační unie) je celosvětová organizace pro informační a komunikační technologie. Je součástí OSN a sídlí ve Švýcarsku.

Má tři části:

- ITU-R: radiokomunikační sektor,
- ITU-T: sektor pro standardizaci telekomunikací (původně CCITT),
- ITU-D: sektor rozvoje telekomunikací.

ITU-T je součást ITU pro standardizaci telekomunikací, její činnost souvisí i s počítačovými sítěmi. Členy s hlasovacím právem jsou zainteresované země, členem bez hlasovacího práva může být kterákoliv organizace. Členství je placené.

ITU-T se dále člení do *studijních skupin* (SG, Study Group), které mají každá své vlastní zaměření. Například SG13 v poslední době vydala několik standardů o cloud computingu, SG16 se zabývá multimédii (přenos videa, obrázky, zvuk apod.), SG17 bezpečností, SG20 chytrými sítěmi (Internet of Things,...). Standardy vytvořené ITU-T jsou otevřené, volně dostupné.

Se standardy ITU-T se setkáváme poměrně běžně: například má „na svědomí" protokol H.323 používaný v multimediálních přenosech a internetové telefonii, protokoly G.992.1 and G.992.2 pro přístupové sítě ADSL, standard X.509 používaný při šifrování a další.

Další informace:
http://www.itu.int/en/ITU-T/Pages/default.aspx

ISO (International Organization for Standardization) je další nezávislá organizace vydávající standardy pro komunikační technologie. Jejími členy jsou standardizační (normalizační) instituce z různých zemí (každá země tam má jedno zastoupení), naopak organizace ISO je členem ITU. Sídlo ISO je také ve Švýcarsku.

ISO má hierarchickou strukturu. Člení se na 200 TC (Technical Committee, technický výbor), každý technický výbor má svou oblast působnosti. Například ISO/TC 47 se zabývá chemií, ISO/TC 20 letadly a vesmírnými vozidly, ISO/TC 272 forenzními vědami, ISO/TC 299 robotikou, ISO/IEC JTC 1 informačními technologiemi.

Každý TC se člení na subkomise (SC, Subcommittee) a každá subkomise se člení na pracovní skupiny (WG, Working Group). Návrhy na standardy nebo jejich změnu jdou vždy zespoda, od pracovních skupin, postupně se přepracovávají a „probojovávají" nahoru až ke schválení standardu. Pracovní skupina vytvoří *pracovní verzi* (Working Paper), ten je přepracován na *koncept* výboru (Committee Draft), následuje *koncept mezinárodního standardu* (Draft International Standard), posledním stupněm je *mezinárodní standard* (International Standard).

Nejznámějším standardem, se kterým se seznámíme už na následujících stranách, je referenční model ISO/OSI (standard ISO 7498 a hodně dalších s ním souvisejících), dále se ISO zabývá sítěmi senzorů, vzdáleným přístupem, UPnP, webovými službami a dalšími tématy.

Další informace:
http://www.iso.org/iso/home/standards_development/list_of_iso_technical_committees.htm

IEEE (Institute of Electrical and Electronics Engineers, čteme anglicky „Eye-triple-E", tedy [aj tripl i:]) je největší světová organizace sdružující odborníky z oblasti elektroniky, elektrotechniky, informatiky a souvisejících oborů, je to tedy profesní sdružení. Nezabývá se jen standardy, ale

pořádá různá setkání, konference, vzdělávací akce, vydává odborné časopisy a vyvíjí další aktivity. Sídlo IEEE je v USA. Jednou z aktivních sekcí IEEE je i ta česká.

Sdružení IEEE stojí především za skupinou standardů IEEE 802 pro lokální a částečně i rozlehlé sítě, do které patří například IEEE 802.11 pro bezdrátové lokální sítě (Wi-fi) nebo IEEE 802.3 pro Ethernet. Přístup k plnému znění standardů je placený.

Další informace:

http://www.ieee.org

IEC (International Electrotechnical Commission) se z oblasti počítačových sítí zabývá především standardy pro elektrická a elektronická zařízení, spolupracuje především s organizací ISO a hodně standardů nese označení „ISO IEC“.

Další informace:

http://www.iec.ch/

IETF (The Internet Engineering Task Force) se zabývá především standardy souvisejícími s Internetem, včetně například protokolů TCP/IP. Velmi úzce spolupracuje s dalšími organizacemi, například IANA (The Internet Assigned Numbers Authority), ISOC (Internet Society), IAB (Internet Architecture Board, Rada pro architekturu Internetu), W3C (World Wide Web Consortium, standardy pro web) a dalšími.

Standardy vydané IETF obvykle začínají zkratkou RFC (Request for Comments) a následuje číslo. Pokud je třeba standard aktualizovat, nemění se původní znění, ale vytvoří se RFC dokument (obsahující popis standardu) s novým číslem. Například pro výše zmíněný otevřený směrovací protokol OSPF bylo takto postupně vytvořeno několik dokumentů (verzí), z nichž jsou momentálně aktuální RFC 2328 (OSPF verze 2) a RFC 5340 (OSPF verze 3).

Další informace:

Oficiální stránky organizace jsou na https://www.ietf.org/. Všechny dokumenty RFC jsou volně dostupné, dokonce existuje řada webových stránek, které je zpřístupňují, například:

- https://tools.ietf.org/html/
- https://datatracker.ietf.org/doc/
- http://www.rfc-base.org/
- https://www.rfc-editor.org/

ANSI (American National Standards Institute) je normalizační úřad USA. Sdružuje státní instituce, komerční společnosti, členy z akademické sféry a další. Kromě vyvíjení vlastních standardů také reprezentuje USA v ISO, IEC a dalších nadnárodních organizacích. Standardy jsou dostupné za poplatek.

ANSI stojí například za standardem pro znakovou sadu ASCII, známý je také standard pro programovací jazyk C (ANSI C), v oblasti sítí například FDDI, CDMA, Frame Relay.

Další informace:

https://www.ansi.org/

TIA (Telecommunication Industries Association) a *EIA* (Electronic Industries Alliance) jsou americké organizace zaměřující se především na fyzickou úroveň komunikace mezi zařízeními, v čemž hodně spolupracují také s ANSI.

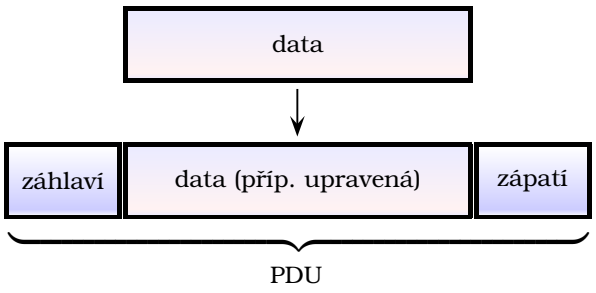
TIA se zabývá standardy pro nejrůznější typy kabelů a konektorů, připojení antén, mobilní sítě, ICT ve zdravotnictví, chytré sítě. EIA stojí například za starým známým konektorem SCART, dále v sítích se setkáváme s konektorem RS-232. Existují standardy vzniklé při spolupráci těchto společností, například TIA/EIA 568, se kterým se setkáme v kapitole o Ethernetu.

Další informace:

- http://www.tiaonline.org/
- https://www.eia.gov/about/eia_standards.cfm

1.4.2 Referenční model ISO/OSI

Protokolová datová jednotka (PDU, Protocol Data Unit) je sekvence dat opatřená metadaty (informacemi o datech) vztahující se ke konkrétnímu protokolu. Protokoly obvykle obdrží data, podle potřeby je určitým způsobem zpracují (strukturují, rozdělí na menší části, zašifrují, komprimují, přeloží, určí adresu příjemce apod.) a přidají před ně *záhlaví* (header) se souvztažnými informacemi (například délka dat, použitý šifrovací algoritmus, adresa odesílatele a příjemce, atd.). Některé protokoly také přidávají za data *zápatí* (trailer) obsahující například kontrolní součet. Data přenášená v datové jednotce (tedy mezi záhlavím a zápatím) taky nazýváme anglickým termínem *payload* (užitné zatížení).

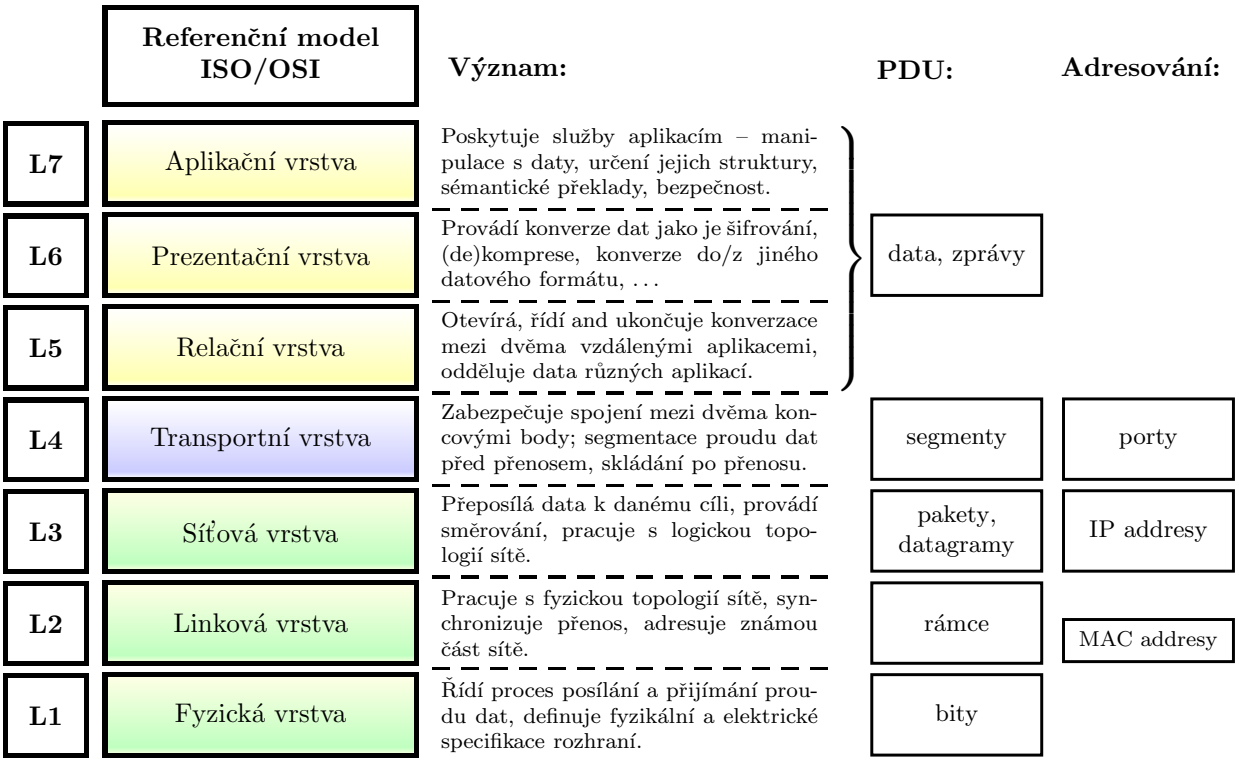


Obrázek 1.3: Datová jednotka (PDU)

K nejdůležitějším standardům z oblasti počítačových sítí patří skupina standardů popisujících *referenční model OSI* (Open Systems Interconnection) publikovaných organizací ISO, proto se označuje RM ISO/OSI. Původní označení standardu je ISO/IEC 7498-1, dnes je dostupný jako ITU-T X.200 na webu http://www.itu.int/rec/T-REC-X.200-199407-I.

OSI definuje sedm vrstev. Každá vrstva plní v komunikaci přes počítačovou síť konkrétní funkci a je přesně stanoveno, co se na dané vrstvě může dít, jedná se tedy o konceptuální model (tj. popisuje logiku návrhu, vztahy mezi součástmi).

Pořadí vrstev a stručný popis najdeme na obrázku 1.4. Vrstvy zobrazené zeleně (L1 až L3) jsou závislé na přenosovém médiu, vrstva L4 (modrá) je přechodová, vrstvy zobrazené žlutě (L5 až L7) jsou nezávislé na přenosovém médiu a na samotném přenosu se podílejí jen nepřímo (přípravou dat a komunikací s aplikacemi).



Obrázek 1.4: referenční model ISO/OSI

Účelem jakéhokoliv konceptuálního návrhu je rozdělit celek na menší části, které se snadněji popisují, a určit vztahy mezi těmito částmi. V našem případě jsou tyto části vrstvy a je stanoveno, jak lze mezi vrstvami komunikovat. V předchozích sekcích jsme se bavili o tom, že komunikace probíhá vždy podle určitých protokolů. Právě protokoly jsou členěny do vrstev modelu OSI (většinou patří konkrétní protokol na jednu konkrétní vrstvu, podle toho, co je jeho účelem).

Dále se budeme věnovat jednotlivým vrstvám. Ke každé potřebujeme znát její účel, typické protokoly, které na této vrstvě pracují a používané datové jednotky.

Fyzická vrstva (L1, Physical Layer). Vrstva L1 je odpovědná za samotný fyzický přenos dat. Definuje přenosové médium (kabely, rádiové vlny apod.), jak je reprezentován bit o hodnotě 0 nebo 1 (kódování), síťové rozhraní (porty, konektory), k čemu konkrétně je používán který vodič v kabelu, a obecně vše, co je potřeba pro konverzi sekvence bitů do formy přenášeného signálu.

Na této vrstvě se nevyskytují žádné datové jednotky, vrstva pouze přijímá proud bitů a přetváří je na signál, který odesílá.

Fyzickou vrstvu mají implementovánu všechna zařízení v počítačové síti, tedy všechna, která jsou opatřena nějakým síťovým rozhraním.

Existují také zařízení, která mají implementovanou pouze fyzickou vrstvu: ze zařízení jmenovaných v sekci o síťovém hardwaru jde o huby (rozbočovače) a repeatery (opakovače). V obou případech to naprosto stačí – hub pouze přijme posloupnost bitů z jednoho portu a přepošle ji na všechny ostatní porty, repeater vezme signál s posloupností bitů z jednoho portu, zesílí ho a odešle na druhý port, žádné z těchto zařízení „vyšší inteligenci“ nepotřebuje. Výhodou těchto zařízení je jednoduchost a rychlost, nevýhodou je nemožnost implementovat pokročilejší funkce.

✎ **Linková vrstva (L2, Data-link Layer, spojová).** Na této vrstvě se určuje vztah mezi příchozím proudem bitů (který vidí fyzická vrstva) a konkrétním uzlem v síti. Zařízení s implementovanou vrstvou L2 má přehled o zařízeních připojených do místní sítě (minimálně „vidí“ ta zařízení, ke kterým je přímo připojeno), vede si tabulku fyzických adres těchto zařízení (nazývá se obvykle MAC tabulka, CAM tabulka apod., podle konkrétního protokolu a konkrétního zařízení). Ke každé adrese v tabulce máme i port, přes který je dotyčné zařízení dosažitelné.

Na linkové vrstvě jsou také zajištěny funkce, které se sice vztahují k přenosovému médiu, ale zároveň vyžadují práci s datovými jednotkami. Například se zde určuje rychlost přenosu, protože tu je třeba řídit i podle toho, v jakém stavu je příjem datových jednotek. Pokud se sem tam nějaká datová jednotka ztratí (což by fyzická vrstva nepoznala), pak zřejmě celý mechanismus „nestíhá“ a je třeba snížit rychlost přenosu. K funkcím vrstvy L2 tedy patří i detekce přenosových chyb.

Datové jednotky, se kterými pracují protokoly této vrstvy, se obvykle nazývají *rámce*. Každý rámec především jednoznačně označuje začátek a konec dat a obsahuje (kromě jiného) fyzickou adresu příjemce (uzlu, kterému mají být data doručena) a fyzickou adresu odesílatele (kdo data poslal).

Vrstvu L2 implementují ta zařízení v síti, která potřebují přehled o uzlech v místní síti a pracují s adresami těchto uzlů, nestačí jim pouze vědět, že „přišly nějaké bity“, takže vlastně skoro všechna (výjimkou jsou například huby a repeatery).

Switche (přepínače) a bridge (mosty) implementují vždy vrstvy L1 a L2, přičemž na vrstvě L2 pracuje vždy některý protokol, který si vede tabulku fyzických adres. Ovšem existují i switche implementující vrstvu L3, ale to už je dodatečná funkcionalita.

Pokud se jedná o aktivní síťový prvek, pak takové zařízení dokáže *oddělovat segmenty v síti*. Například pokud hub (který nemá implementovanou vrstvu L2) přijme data, musí je odeslat na všechny porty (kromě toho, ze kterého data přišla), tedy do všech připojených segmentů, protože cestu nedokáže určit. Ale switch (má implementovánu vrstvu L2) se při přijetí dat podívá se do záhlaví rámce, ve kterém jsou data uložena, najde tam adresu příjemce, v tabulce fyzických adres zjistí port, přes který je příjemce dosažitelný (přesněji – segment sítě, ve kterém se příjemce nachází), a rámec odešle jen na tento jeden port.

Výjimkou jsou *všesměrové (broadcastové) adresy*. Takový rámec je určen „všem v síti“, například zjišťovací rámec s dotazem „kdo má adresu xxx?“, když se vytváří tabulka fyzických adres. Rámec s adresou příjemce všesměrovou je poslán na všechny porty kromě toho, ze kterého přišel.

✎ **Síťová vrstva (L3, Network Layer).** Zatímco protokoly linkové vrstvy mají přehled o fyzické topologii sítě, protokoly síťové vrstvy pracují s logickou topologií sítě a „vidí“ i za hranice lokální sítě. Úkolem síťových protokolů je stanovit skutečnou cestu (nebo úsek cesty, za který je dotyčné zařízení zodpovědné), tedy určit adresu vyšší úrovně a *směrovat*.

Datové jednotky na síťové vrstvě jsou označovány jako *pakety* nebo *datagramy* (záleží na konkrétním protokolu, a taky na dotyčném zdroji informace). Jaký je mezi nimi rozdíl?

- *Datagram* je datová jednotka posílaná v rámci nespojované (datagramové) služby, viz str. 5.
- *Paket* je datová jednotka posílaná v rámci spojované služby, ale často se používá i v obecnějším významu (prostě jako datová jednotka).

Síťová vrstva je implementovaná na koncových zařízeních a dále na těch aktivních síťových prvcích, které zajišťují směrování, pracují s logickou topologií sítě, propojují nejen segmenty, ale také celé

sítě, což jsou routery (směrovače). Router tedy implementuje vrstvy L1, částečně L2 (jen to, co je nezbytné k propojení s vyšší vrstvou) a L3.

Také na síťové vrstvě je vedena (minimálně jedna) tabulka s adresami, ale tentokrát jde o *směrovací tabulku* (routing table). Ve směrovací tabulce je informace o tom, kam poslat datovou jednotku patřící do určité (pod)sítě – můžeme si to představit tak, že na řádku je adresa cílové (pod)sítě, brána (zde ve smyslu blízkého zařízení, přes které se dá do té (pod)sítě dostat, tedy určení dalšího kroku na cestě), síťové rozhraní, přes které mají data vyjít z tohoto zařízení a další informace.

Takže pokud směrovač dostane data k přeposlání, najde v záhlaví síťové vrstvy logickou adresu příjemce a postupně prochází jednotlivé řádky směrovací tabulky – zastaví se na prvním řádku takovém, že adresa příjemce patří do (pod)sítě na tomto řádku uvedené. V řádku si přečte směr, kterým má data poslat.

Transportní vrstva (L4, Transport Layer). Tato vrstva je jakýmsi přechodem mezi vrstvami orientovanými na proces přenosu (L1–L3) a vrstvami orientovanými na aplikace a tedy nezávislými na procesu přenosu (L5–L7). Takže směrem nahoru potřebuje vazbu na konkrétní protokol protokol vyšší vrstvy (této vazbě, číslu, které je obdobou adresy, říkáme *port*) a směrem dolů uplatňuje funkce související s přenosem dat.

Poznámka:

Pozor – pojem „port“ se v počítačových sítích používá ve dvou velmi odlišných významech:

- (fyzický) port jako součást síťového rozhraní, jak je vysvětleno na straně 9,
- port (určený číslem) v záhlaví datové jednotky transportní vrstvy určující, se kterým protokolem vyšší vrstvy se právě komunikuje.

Například port číslo 80 znamená komunikaci s protokolem HTTP.

Datovou jednotkou transportní vrstvy je *segment*, a pokud jde o nespojovanou službu, může se použít i pojem *datagram*. Hlavním úkolem protokolů transportní vrstvy (tedy kromě práce s číslem portu) je *segmentace dat* z vyšší vrstvy na dostatečně malé úseky, které bude možné přes síť přepravit. Data jsou rozdělena na úseky o stanovené délce, ke každému je přidáno záhlaví s údaji transportní vrstvy, čímž je vytvořen segment.

Na transportní vrstvě se rozhoduje, zda bude přenos realizován formou služby se spojením nebo služby bez spojení. U služby se spojením zajišťuje vrstva L4 navázání spojení (handshake), řídí průběh spojení, potvrzování doručených segmentů, v případě ztráty či poškození dat zajistí opakování přenosu segmentu (tedy potvrzovaná, spolehlivá služba) a podle potřeby zajistí úpravu parametrů spojení, a pak ukončí spojení.

Transportní vrstva je implementovaná obvykle jen na koncových zařízeních.

Relační vrstva (L5, Session Layer). Na této vrstvě jsou oddělena data patřící různým aplikacím. Každá aplikace komunikující se sítí přes tuto vrstvu navazuje *relaci* (session) s nějakou aplikací na jiném systému, a v rámci této relace se například přenášejí data. Z toho vyplývá, že relace je logické spojení mezi dvěma aplikacemi navázané (většinou) za účelem výměny dat, které může vést přes síť.

Poznámka:

Jaký je rozdíl mezi spojením na transportní vrstvě a relací na relační vrstě? Zatímco spojení se navazuje mezi dvěma zařízeními, relace se navazuje mezi dvěma aplikacemi na těchto zařízeních. Jak bylo výše uvedeno – spodní vrstva poskytuje služby vyšší vrstvě, a tedy pokud relační vrstva chce pro určitou aplikaci navázat relaci s aplikací na jiném systému, potřebuje k tomu kromě jiného také spojení, které jí zajistí transportní vrstva.

Relační vrstva, stejně jako všechny vyšší vrstvy, je implementovaná na koncových zařízeních.

Prezentační vrstva (L6, Presentation Layer). Vrstva L6 je zodpovědná za provádění konverzí dat (žádná z nižších vrstev do dat nezasahuje), například úpravy kódování textu (ASCII, EBCDIC apod.), komprese a dekomprese, šifrování a dešifrování, některé úkoly související se zpracováním multimédií.

Proč se tato vrstva vlastně nazývá prezentační? Protože má za úkol data prezentovat vyšším vrstvám v takové formě, které tyto vrstvy rozumí.

Aplikační vrstva (L7, Application Layer). Na této vrstvě pracují protokoly, které jsou využívány aplikacemi (od toho název). Například aplikace webový prohlížeč využívá (kromě jiného) aplikační protokol HTTP. Funkcí této vrstvy je při odesílání dat převzat data od aplikace a předat nižším vrstvám; naopak při přijímání dat jsou tato data přijata od nižších vrstev a předána příslušné aplikaci.

Poznámka:

Pozor, na aplikační vrstvě jsou aplikační protokoly, nikoliv aplikace!

Nadřízená vrstva vytvoří svou PDU (například paket) a odešle podřízené vrstvě. Pro podřízenou vrstvu je to, co takto obdržela, označováno jako *SDU* (Service Data Unit). Celý postup odesílání dat je takový, že každá vrstva obdrží od nadřízené vrstvy SDU a přidáním záhlaví (a případně i zápatí) z ní vytvoří PDU a předá nižší vrstvě. To platí i pro aplikační vrstvu – nad ní sice žádná vrstva není, ale sekvenci dat, kterou obdrží od některé aplikace, je pro ni SDU.

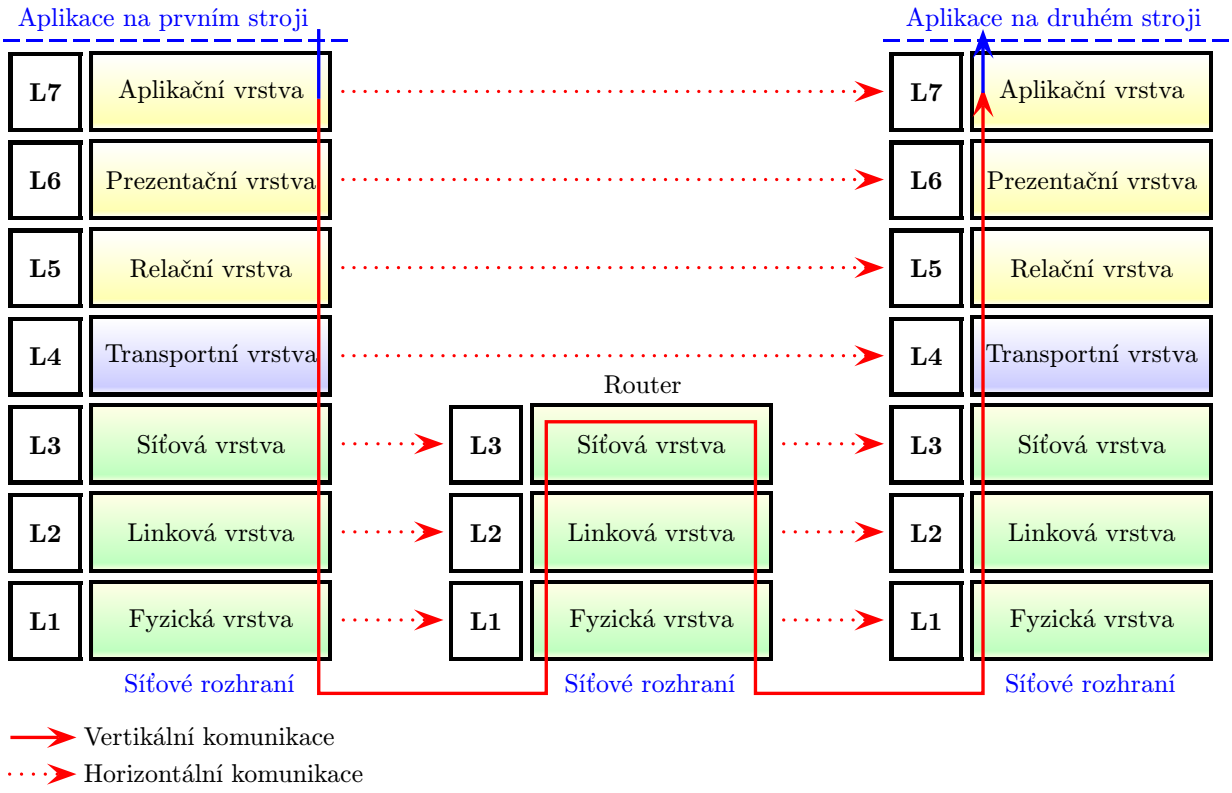
1.4.3 Spolupráce protokolů

Víme, že protokoly nemají být moc komplexní, a tedy potřebují spolupracovat s jinými protokoly. Spolupráce může probíhat buď v rámci jedné vrstvy, nebo mezi sousedními vrstvami, přičemž platí, že spodní vrstva poskytuje služby horní vrstvě.

Entita je aktivní prvek na určité vrstvě v modelu ISO/OSI, který má definováno rozhraní – sadu služeb, které může využívat entita z bezprostředně nadřízené vrstvy. Entitu vrstvy n označujeme *n-entita*. Na n-té vrstvě ISO/OSI je tedy sada n-entit.

Rozhraní mezi komunikujícími entitami (a tedy mezi vrstvami) se nazývá *SAP* (Service Access Point, přístupový bod služby). SAP tedy propojuje dvě entity v sousedních vrstvách – uživatele služby (service user) a poskytovatele služby (service provider).

Způsob komunikace v rámci jednoho systému (řetěz střídajících se entit a SAP) se v ISO/OSI nazývá *vertikální komunikace*. *Horizontální komunikace* v ISO/OSI je komunikace mezi dvěma



Obrázek 1.5: Horizontální a vertikální komunikace v ISO/OSI

stejnými vrstami umístěnými na různých strojích, a to na logické úrovni. Oba způsoby komunikace jsou naznačeny na obrázku 1.5.

Pod pojmem entita si můžeme představit (spuštěnou) instanci některého konkrétního protokolu nebo jejich sady. Jeden SAP může být v jednom okamžiku využíván pouze jedním uživatelem a jedním poskytovatelem, ale jedna entita může zároveň používat více SAPů.

Protokol tedy při odesílání dat (podle obrázku 1.5 na stroji vlevo)

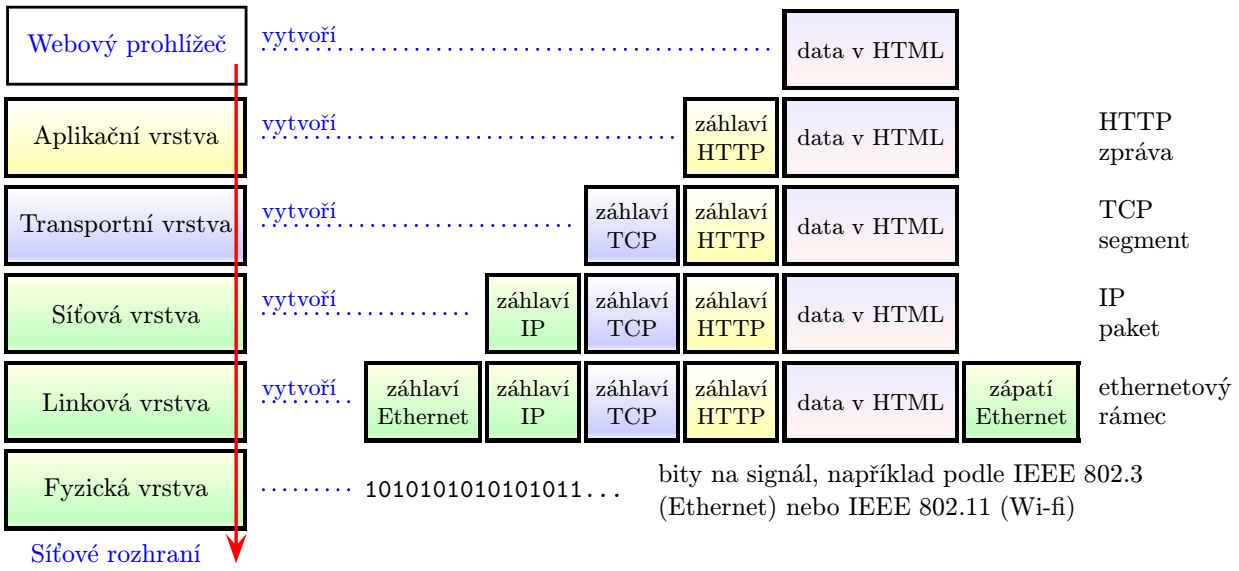
- obdrží data přes SAP od vyšší vrstvy,
- pokud je to nutné, stanoveným způsobem je zpracuje či rozdělí na menší bloky,
- stanoví příslušné metainformace (tj. informace o informacích), například adresy, velikost dat, informace o tom, jak se má po cestě s daty zacházet, apod.,
- přidá záhlaví (header) s metainformacemi a pokud je to třeba, pak i zápatí (trailer), přidá k datům, čímž data „zabalí“ do PDU (paket, rámec, datagram apod.),
- předá PDU přes SAP nižší vrstvě.

Jestliže jsou data naopak přijímána (podle obrázku 1.5 na stroji zcela vpravo), pak protokol

- přijme PDU přes SAP od nižší vrstvy,
- oddělí od PDU záhlaví a zápatí (pokud tam ovšem zápatí je),
- analyzuje metainformace v záhlaví a zápatí, stanoveným způsobem je zpracuje,
- pokud byla data před odesláním rozdělena na více bloků a jedná se o vrstvu, kde jsou bloky kompletovány (pozná se ze záhlaví), postupně shromáždí všechny bloky a zkompletuje,
- „rozbalená“ data předá přes SAP do vyšší vrstvy.

Není řečeno, že se nutně mají zúčastnit všechny vrstvy, některé nejsou pro konkrétní druh komunikace potřebné.

Při odesílání se tedy provádí *enkapsulace* (zabalení, encapsulation) dat a při přijetí *dekapsulace* (rozbalení, decapsulation). To, co je pro nadřízenou vrstvu její vlastní PDU, to je pro jí podřízenou vrstvu pouze sekvence dat, ze kterých vytvoří vlastní PDU.



Obrázek 1.6: Zapouzdření PDU protokolu HTTP

Na obrázku 1.6 je naznačen proces enkapsulace. Postup při odesílání:

- data vyprodukovaná webovým prohlížečem jsou na vrstvě L7 zabalena do protokolové datové jednotky protokolu HTTP (do záhlaví se uloží například adresa serveru, informace o „žádajícím“ webovém prohlížeči, znakové sadě, časové údaje a další),
- následně se na transportní vrstvě dojedná spojení (nebo se využije už dojednané), pokud je to nutné, provede se segmentace (rozdělení na menší bloky) a následně se připojí TCP záhlaví (to obsahuje například číslo portu určující, že na nadřízené vrstvě komunikuje protokol HTTP, informace pro zajištění zkompletování celku při rozdělení na víc segmentů, kontrolní součet atd.), čímž je vytvořen TCP segment,
- na vrstvě L3 je tento segment předán entitě protokolu IP a je přidáno záhlaví protokolu IP (obsahuje například IP adresu zdroje a cíle, informaci o tom, který protokol sestavil to, co je „uvnitř“, atd.), výsledkem je IP paket,
- přes SAP mezi síťovou a linkovou vrstvou se IP paket dostane k entitě protokolu IEEE 802.3 (Ethernet), který před něj připojí ethernetové záhlaví (to obsahuje například synchronizační sekvenci bitů, aby byl rozeznatelný začátek bitů paketu, fyzické adresy apod.) a za něj ethernetové zápatí (s kontrolním součtem),
- další na řadě je fyzická vrstva, která již zajistí převod sekvence bitů na signál podle určeného protokolu a připojeného přenosového média.

Na straně adresáta proběhne opačný proces – rozbalování.

Ve skutečnosti by se tohoto přenosu účastnily i další protokoly, například protokol DNS překládající adresy nebo při zabezpečené komunikaci protokol SSL.

Všimněte si, že některé vrstvy modelu ISO/OSI se tohoto procesu neúčastní, jsou tedy výjimky z pravidla komunikují pouze bezprostředně sousedící vrstvy – protokoly aplikační vrstvy mohou používat přístupové body SAP z vrstev L6, L5 i L4.

Jak se vlastně komunikuje přes takový SAP, co vše se na tomto přístupovém bodu děje? Ke každému SAPu jsou definována komunikační *primitiva*, což jsou jednoduché funkce, například *Request* (žádost o poskytnutí služby k nižší vrstvě, navazuje komunikaci přes SAP směrem dolů na straně odesílatele), *Indication* (upozornění na potřebu komunikace od nižší vrstvy k vyšší vrstvě, na straně příjemce), *Response* (odpověď na Indication), *Confirm* (odpověď na Request).

Tato primitiva mohou mít, podobně jako běžné funkce, parametry – buď se jedná o SDU předávanou přes SAP z vyšší vrstvy, nebo se může jednat o dodatkové provozní informace, které nemají být součástí výsledné PDU.

Poznámka:
Zvídavého čtenáře určitě napadlo, že musí existovat způsob, jak se například síťová vrstva (L3) „dozví“, na kterou adresu má být vlastně dotyčný paket poslán. Přes záhlaví PDU to být nemůže, protože dovnitř záhlaví vyšších vrstev se protokol IP nedostane (je to pro něj prostě součást dat, která je třeba poslat), navíc v některých záhlavích tato informace ani není. Ano, dozví se to z parametrů primitiv. Další informace najdete na
http://www.erg.abdn.ac.uk/users/gorry/course/intro-pages/service-prim.html.

Socket je kombinace adresy zařízení (infromace vrstvy L3 nebo něco, co se dá na ni přeložit) a čísla portu (vrstva L4). Pokud tento pár zapisujeme „ručně“ (třeba do adresního řádku webového prohlížeče), umístíme mezi oba údaje dvojtečku. Ukázky (číslo portu 8080):

- `77.75.74.176:8080` (IPv4 adresa)
- `[2a02:598:3333:1::1]:8080` (IPv6 adresa; všimněte si, že dvojtečky nesmí kolidovat)
- `www.neco.cz:8080` (jmenná adresa)

Sockety (sockets) najdeme na vrstvách L5–L7, víceméně fungují jako rozhraní mezi aplikačními protokoly a transportní vrstvou. Pro aplikace jsou dostupné jako Socket API (tj. rozhraní pro programování aplikací) ve formě dynamicky linkovaných knihoven obsahujících funkce pro práci se sockety (pro vytvoření socketu, akceptování na druhé straně spojení, naslouchání, čtení, zápis do socketu). Speciálním typem socketu je *stream socket*, který zaručuje dodání více bloků posílaných dat ve správném pořadí, a tedy na transportní vrstvě spolupracuje pouze s protokoly zajišťujícími spojovaný přenos (třeba TCP).

Se sockety se setkáme jak v UNIXových systémech (včetně Linuxu a MacOS X), tak i ve Windows (WinSock API).

1.4.4 Protokolové zásobníky

Sada protokolů (Protocol Suite) je definice (určení) skupiny protokolů, které spolupracují. *Protokolový zásobník* (Protocol Stack) je implementace některé sady protokolů. Tyto dva pojmy se často používají jako synonyma.

Nemusí nutně jít o specifikaci protokolů pro naprosto všechny vrstvy modelu ISO/OSI, mohou být specifikovány pouze některé, přičemž se předpokládá spolupráce s jiným (doplňujícím) protokolovým zásobníkem.

✎ *TCP/IP Protocol Stack* (taky se nazývá *Internet Protocol Suite*) je sada navzájem spolupracujících protokolů, kterou potřebujeme na koncových zařízeních připojených k rozsáhlé síti (typicky Internetu). Kromě protokolů obsažených přímo v názvu (TCP, IP) zahrnuje ještě další, nejvíc na aplikační vrstvě. Přímo jsou určeny protokoly na vrstvách L3–L7, pro nižší vrstvy je pouze specifikováno rozhraní.

Tento protokolový zásobník je formalizován jako síťový model TCP/IP, kterému se budeme podrobněji věnovat v následujícím textu (včetně protokolů).

✎ *IPX/SPX* je konkurenčním protokolovým zásobníkem k TCP/IP od společnosti Novell vytvořeným pro operační systém Novell Netware – IPX pracuje na L3 místo protokolu IP, SPX na vrstvě L4 místo TCP. Byl projektován spíše pro menší sítě, zatímco TCP/IP je určen i pro rozlehlé sítě. V současné době se už téměř nepoužívá.

✎ *Protokolové sady pro lokální sítě* jsou například Ethernet (IEEE 802.3), Wi-fi (IEEE 802.11) Token Ring (IEEE 802.5, už se nepoužívá) a další. Obvykle implementují pouze vrstvy L1 a L2, nad ně se nasouvá obvykle protokolový zásobník TCP/IP (ten pro změnu přímo nespecifikuje protokoly na L1 a L2).

✎ *Protokolové sady pro rozlehlé sítě* jsou například ATM, Frame Relay, MPLS a další. Taky se obvykle napojují na TCP/IP, ale každá „trochu jinak“. Například ATM se podsouvá pod síťovou vrstvu (L3), ale mezi ni a svou implementaci vrstvy L2 vsouvá speciální přizpůsobovací vrstvu. Frame Relay implementuje vrstvu L2, na L1 předpokládá některé vhodné fyzické rozhraní, většinou dle standardů EIA/TIA (spoléhá na ISO/OSI).

Oproti tomu MPLS se vsouvá mezi L2 a L3, tedy MPLS paket v sobě zabaluje paket z vrstvy L3 (většinou IP paket) a je zabalen do rámce vrstvy L2 (například do etnernetového rámce). Taky může běžet nad ATM nebo Frame Relay, a tedy lze využít technologie ze starších zařízení. Taky dokáže běžet nad PPP a dalšími protokoly pro přístupové sítě.

✎ *Protokolové sady pro mobilní sítě* jsou například sady pro LTE, GPRS, CDMA, UMTS a další. Implementují obvykle vrstvy L1 a L2 a předpokládají některé konkrétní protokoly i na vyšších vrstvách, ale ve skutečnosti záleží, o jaký typ zařízení jde (koncové zařízení bude potřebovat jinou sadu protokolů než základnová stanice nebo jiná specializovaná zařízení v mobilní síti).

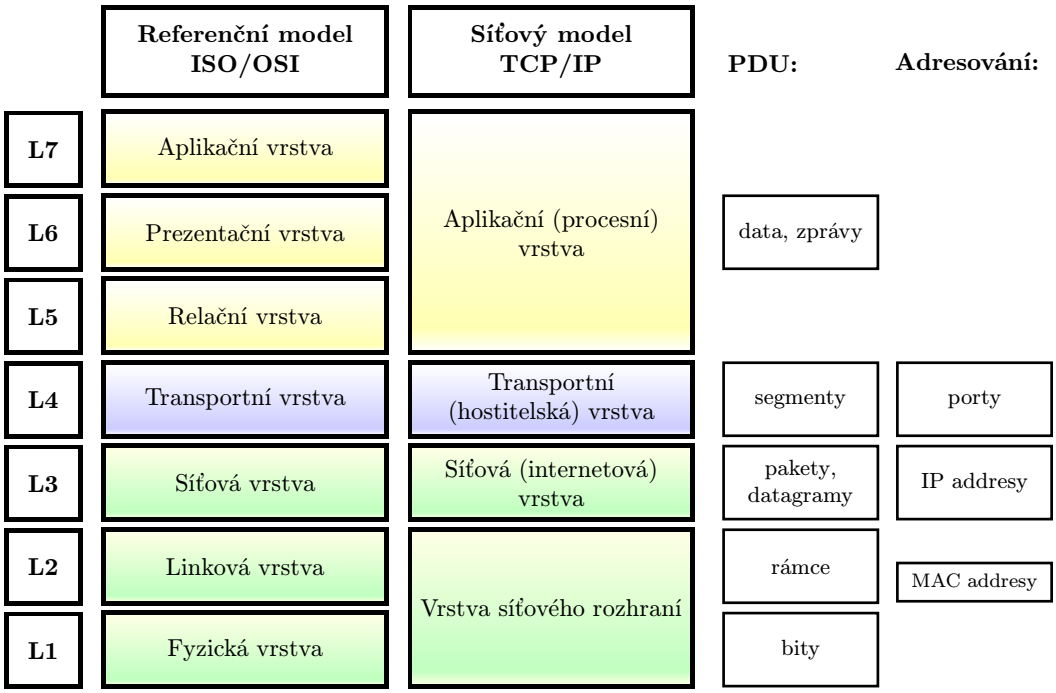
1.4.5 Model TCP/IP

Referenční model ISO/OSI je velmi komplexní, tudíž složitý, a příliš teoretický. Postupně bylo vytvořeno několik zjednodušených variant, z nichž je nejznámější právě síťový model TCP/IP, který je také nazýván DoD model (USA Department of Defense Model, tedy model Ministerstva obrany USA). Jeho součásti jsou také standardizovány organizací IETF a dostupné v RFC dokumentech.

Síťový model TCP/IP je vlastně formální popis síťového zásobníku TCP/IP, jeho napojení na spolupracující zásobníky a obecně možnost zapojení jiných protokolů. Skládá se ze čtyř vrstev, jejichž vztah k vrstvám RM OSI je naznačen na obrázku 1.7.

Vztahy vyjádřené na obrázku platí i co se týče funkčnosti – aplikační (procesní) vrstva TCP/IP plní tutéž roli jako vrstvy L5–L7 v ISO/OSI. TCP/IP je navržen tak, aby

- byl co nejvíc decentralizovaný (žádná centrální správa),
- byl co nejodolnější vůči různým (i kritickým) podmínkám provozu a co nejodolnější vůči přenosovým chybám,



Obrázek 1.7: Srovnání modelů RM ISO/OSI a TCP/IP

- nechával co nejvíc práce na koncových zařízeních (jádro sítě musí být rychlé a pružné), a aby byl spravovatelný distribuovaně (každá část si spravuje „to svoje“,
- dokázal propojit i sítě s hodně odlišnou síťovou architekturou a technologiemi (aby byl co nejuniverzálnější při zachování předchozích vlastností).

Poznámka:

Síťový model TCP/IP se také nazývá DoD (model ministerstva obrany), z toho je patrné, že na jeho návrhu spolupracovaly ozbrojené složky USA. Proč se tímto způsobem angažovaly nejen v návrhu síťových standardů, ale vlastně k fungování rozlehlých počítačových sítí obecně?

Jednoduše proto, že už v té době bylo jasné, že rozlehlé počítačové sítě jsou a budou pro fungování státu velmi důležité. Síť ARPANET byla navržena *maximálně decentralizovaně*, tedy tak, aby poškození jedné části sítě co nejméně narušilo fungování zbytku sítě, a tuto vlastnost přejal i Internet. Také síťový model TCP/IP je navržen s podobnými úmysly, jak je vidět na vlastnostech uvedených v odrážkách nad touto poznámkou.

Postupně projdeme všechny vrstvy a na rozdíl od referenčního modelu se soustředíme především na protokoly pracující na těchto vrstvách.

Vrstva síťového rozhraní. Tato vrstva v sobě sdružuje funkčnost vrstev L1 a L2 referenčního modelu. Přímo v TCP/IP pro ni nejsou stanoveny žádné protokoly, je jen určeno, jak mají komunikovat se síťovou vrstvou. Komunikuje s hardwarem (síťovým rozhraním), případně její část může být hardwarově implementovaná. Obvykle se do této vrstvy napojují protokolové sady lokálních a rozlehlých sítí, například

- IEEE 802.3 (Ethernet), IEEE 802.11 (Wi-fi),
- protokolové sady WAN sítí nebo jejich části, xDSL, mobilních sítí.

Na nižší části této vrstvy (obdoba L1) najdeme jednoduchá zařízení pracující pouze se signálem typu hub (rozbočovač) nebo repeater (opakovač). Ve vyšší části této vrstvy (obdoba L2) pak o něco složitější zařízení pracující s rámci a spojující (či oddělující) jednotlivé segmenty sítě, tedy switch (přepínač) a bridge (most).

Pokud si budeme všímat jen vyšší části této vrstvy, pak ze probíhá proces *přepínání rámců*.

Síťová vrstva. Také se nazývá „internetová vrstva“, na této vrstvě probíhá *internetworking* (propojování sítí), včetně *směrování* mezi sítěmi. Pojem „internet“ (s malým počátečním písmenem) značí obecně síť sítí, tedy síť propojující nikoliv jen jednotlivé uzly, ale menší sítě. Typické zařízení této vrstvy je router (směrovač) nebo L3 switch.

Co se protokolů týče, bude nás zajímat především protokol IP (Internet Protocol), a to jeho verze IPv4 a IPv6, které jsou dnes v praxi používány, a dále směrovací protokoly (OSPF, EIGRP, RIP, BGP a další).

Z dalších protokolů to je například ICMP sloužící k jednoduchému dorozumívání mezi zařízeními implementujícími síťovou vrstvu (tento protokol je používán například tehdy, když se příkazem `ping` dotazujeme určitého uzlu v síti, zda je dostupný).

Protokolům síťové vrstvy je věnována samostatná kapitola.

(Transportní vrstva). Také ji nazýváme „hostitelská vrstva“, protože je implementována pouze na hostitelích v síti. *Hostitel* je název pro koncové zařízení (počítač, server, tablet, chytrá televize, atd.), které „hostí“ data, aplikace, služby; každý hostitel má svůj název (hostname).

Poznámka:
„Hostitel“ se anglicky řekne „host“, kdežto anglickým ekvivalentem českého „host“ je „guest“. Tedy anglické „hostname“ se překládá jako „název hostitele“ (název koncového zařízení).

Nejznámější protokoly transportní vrstvy jsou TCP a UDP:

- Protokol TCP se používá tehdy, když je třeba navázat spojení a ošetřovat výskyt přenosových chyb (tedy poskytuje spolehlivou spojovanou službu – naváže spojení, udržuje ho, při výskytu chyby opakuje odeslání a na konci přenosu toto spojení ukončí).
- Protokol UDP naproti tomu poskytuje nespolehlivou službu bez navázání spojení – spolehlivost je pouze na úrovni best-effort, žádné spojení se nenavazuje, pouze sestaví segment a odešle ho.

Aplikační vrstva. Také „procesní vrstva“. Sdružuje v sobě vše, co je v RM ISO/OSI na nejvyšších třech vrstvách (L5–L7). Na této vrstvě najdeme velké množství protokolů, s většinou z nich komunikují aplikace potřebující přistupovat na síť. Příklady aplikačních protokolů:

- Protokol HTTP je kromě jiných využíván webovými prohlížeči.
- Protokoly SMTP, IMAP a POP3 jsou využívány poštovními klienty nebo jakýmikoliv aplikacemi schopnými pracovat s e-maily. SMTP umožňuje odeslat e-mail, další dva protokoly se používají při práci s doručenou poštou.
- Protokol DNS pomáhá s překladem adres (když do adresního řádku webového prohlížeče zadáme jmennou adresu, například `www.seznam.cz`, DNS ji přeloží na číselnou IP adresu, které již „rozumí“ aktivní síťové prvky na síti).
- Protokol DHCP nám přidělí dynamickou IP adresu, když se přihlašujeme do sítě.

Aplikačním protokolům je také věnována samostatná kapitola.

Poznámka:

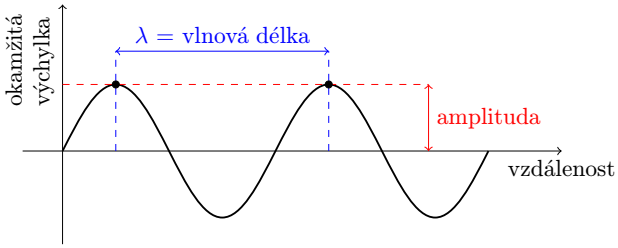
Pokud jste na rozpacích, který z výše uvedených modelů (RM ISO/OSI nebo TCP/IP) vlastně používat, pak buďte ujištěni, že oba. Každý z modelů má své výhody a nevýhody. Většinou se používá terminologie z ISO/OSI (například označení vrstev, entity apod.), ale celkové rozvržení činností souvisejících se sítí a implementace postupů jsou obvykle podle TCP/IP.

1.5 Charakteristiky přenosu a přenosových cest

1.5.1 Přenosový signál

Pro přenos dat se používá *přenosové médium*. To může být metalický kabel (většinou měděný), optický kabel, vzduch (rádiový spoj) nebo něco jiného. Ať už použijeme jakékoliv přenosové médium, data se přes něj přenášejí jako *sled signálů*. Podrobnosti o fyzikální podstatě přenosu dat nejsou součástí sylabu tohoto předmětu, nám bude stačit seznámit se s některými pojmy, které budeme ve zbytku semestru používat.

Signál se přenosovým médiem přenáší ve formě *vlny* (kmitání částic – elektronů, fotonů apod.). Signál může být generován nebo je přenášená informace *modulována* na *nosný signál* (tj. původní signál je pozměněn). Na obrázku 1.8 jsou naznačeny některé fyzikální charakteristiky signálu.



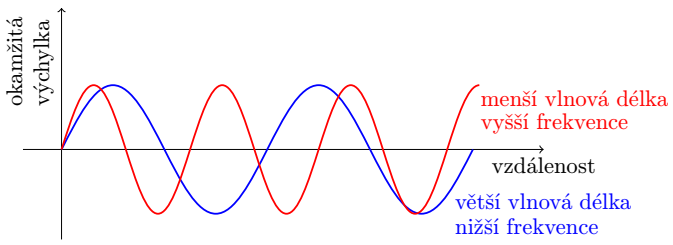
Obrázek 1.8: Fyzikální charakteristiky signálu

Amplituda je maximální výchylka signálu.

Vlnová délka je vzdálenost, kterou signál urazí během periody signálu, tedy vzdálenost mezi dvěma nejbližšími body ve stejné fázi. Vlnová délka nás bude zajímat zejména u optických kabelů.

Frekvence je obrácená hodnota vlnové délky vynásobená rychlostí signálu. Čím větší je vlnová délka, tím menší je frekvence, a naopak.

Poznámka:



Obrázek 1.9: Vztah mezi vlnovou délkou a frekvencí

Připomeňme si, že jednotkou frekvence je 1 Hz, a je definována jako počet cyklických dějů probíhajících za dobu 1 sekundy, takže odpovídá 1 s^{-1} podle soustavy SI. Na obrázku 1.9 je naznačen vztah mezi vlnovou délkou a frekvencí. Vzorec je

$$f = v \cdot \frac{1}{\lambda} \qquad \text{resp.} \qquad \lambda = v \cdot \frac{1}{f}$$

kde f je frekvence, v je rychlost signálu a λ je vlnová délka.

Příklad

Uveďme si příklad o zvukovém signálu. Rychlost zvuku záleží na nadmořské výšce (tj. hustotě vzduchu) a teplotě, předpokládejme, že v našem případě to bude 330 m/s. Zvuk o frekvenci 1000 Hz šířící se touto rychlostí bude mít vlnovou délku $\lambda = 330 \cdot \frac{1}{1000} = 0,33$ m $= 33$ cm.

V optickém vlákně určeném pro přenos na velké vzdálenosti (jednotky až desítky kilometrů) je laserem emitován optický signál o vlnové délce 1310 nm. Jedná se o světlo, takže hodnota v bude velmi velká (téměř 300 000 m/s), kdežto vlnová délka takového signálu je naopak velmi malá (1310 nm $= 1310 \cdot 10^{-9}$ m). Ze vzorce plyne, že frekvence bude velmi vysoká (přes 200 THz).

Šířka pásma (bandwidth) je rozdíl mezi nejnižší a nejvyšší frekvencí přenosového signálu neboli šířka intervalu frekvencí, které lze přenosovou cestou přenést. Například pro měděný etnernetový kabel kategorie 5e (o tom se budeme učit později) je šířka pásma 100 MHz, u kategorie 6 to je 200 MHz. Obecně můžeme říct, že čím větší šířka pásma, tím větší rychlosti přenosu se dá dosáhnout (ale záleží i na dalších parametrech).

1.5.2 Přenos v základním a přeloženém pásmu

Přenos se provádí buď v základním pásmu nebo v přeloženém pásmu.

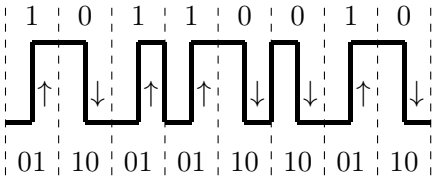
Přenos v základním pásmu (baseband). Přenos probíhá tak, že sekvenci jedniček a nul, která má být přenesena, přímo zakódujeme do frekvenčního spektra (emitujeme signál) a takto vzniklý signál je přenesen komunikačním kanálem. Obvykle nám stačí nízké frekvence. Baseband se používá pro metalické lokální sítě (Ethernet na měděném kabelu) a pro optické sítě.

Nevýhodou je omezený dosah (menší vzdálenost pro přenos) a problematická synchronizace (pokud bychom sekvenci nul a jedniček kódovali tak jak je, pak by dlouhé sekvence nul nebo dlouhé sekvence jedniček byly špatně dekódovatelné, nebylo by možné stanovit jejich délku). Odesílající a přijímající strana potřebují mít správně seřízené časovače, aby bylo možné synchronizovat intervaly mezi úseky představujícími jednotlivé bity, ale u dlouhých sekvencí bitů se stejnou hodnotou to nestačí.

Tento problém se obchází jednoduše tak, že posloupnost bitů před kódováním na signál pozměníme podle určitého klíče tak, aby se v posloupnosti nevyskytovaly dlouhé sekvence stejných číslic, a samozřejmě aby bylo možné data po ukončení přenosu vrátit do původního stavu. Každý bit se jednoduše nahradí určitou sekvencí bitů tak, aby se dostatečně často „střídaly“ hodnoty 0 a 1. Dále si představíme několik obvyklých kódování pro přenos v základním pásmu.

Kódování Manchester je velmi jednoduché. Spočívá v zakódování bitů do směru kmitu signálu – 0 je kódována jako přechod shora dolů, 1 je kódována jako přechod zdola nahoru (viz obrázek 1.10).

0:	High → Low	10
1:	Low → High	01



Obrázek 1.10: Kódování Manchester pro oktet $(10110010)_2$

Sekvence bitů 10111000 je při kódování Manchester zakódována na 01 10 01 01 01 10 10 10.

Jak vidíme, stejné symboly vedle sebe získáme pouze na těch místech, kde se v původní sekvenci měnily hodnoty bitů, a to nejvýše dva stejné symboly. To je výhodou kódování Manchester, nevýhodou je navýšení délky posílaných dat (délka se oproti původním zdvojnásobí). U vyšších rychlostí jsou další techniky jako data scrambling nebo samoopravitelný kód (Gigabit Ethernet).

Kódování 4B/5B: z každé čtveřice bitů se vytvoří pět bitů tak, aby v celé sekvenci bylo co nejvíce jedniček (nejméně dvě na pětici), na začátku pětice nejvýše jedna nula, na konci nejvýše dvě nuly. Kódy jsou v tabulce 1.3.

4B	5B	4B	5B	4B	5B	4B	5B
0000	11110	0100	01010	1000	10010	1100	11010
0001	01001	0101	01011	1001	10011	1101	11011
0010	10100	0110	01110	1010	10110	1110	11100
0011	10101	0111	01111	1011	10111	1111	11101

Tabulka 1.3: Tabulka kódů pro 4B/5B

Sekvenci 10111000, kterou jsme v kódování Manchester zakódovali do 16 znaků, zde zpracujeme na 1011110010 o délce 10 znaků.

Kódování MLT-3 používá tři úrovně napětí (předchozí kódování používala jen dvě úrovně), a to -, základna, +. Změna napětí proběhne pouze na signál 1, a to na „sinusovce“, například pro sekvenci jedniček by bylo 0, 1, 0, -1, 0, 1, 0, -1, atd. Toto kódování se obvykle kombinuje s některým jiným, například signál pro MLT-3 může být předzpracován kódováním 4B/5B.

Oproti Manchestru má signál jen čtvrtinovou frekvenci, proto méně vyzařuje, generuje mnohem méně rušení do okolí.

Přenos v přeloženém pásmu (broadband). Tento přenos probíhá tak, že sekvenci jedniček a nul sice zakódujeme také do frekvenčního spektra jako u basebandu, ale vzniklý signál přeložíme do takového pásma, které je pro tento konkrétní přenos určeno (nebo kódujeme data přímo do příslušného frekvenčního pásma). Tomuto překladu říkáme *modulace*.

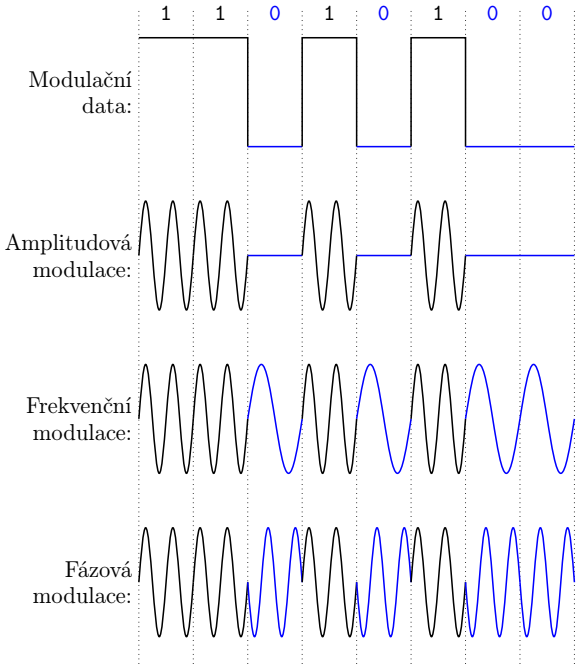
Modulace tedy probíhá následovně:

- Určíme *nosnou*, což je vhodný signál na té frekvenci, na které mají být data přenesena, volí se signál s harmonickou frekvencí, obvykle sinusoida.
- Pozměníme parametry (amplitudu, frekvenci nebo fázi) tohoto signálu podle toho, jaká data mají být přenášena – modulujeme data na signál.
- Podle potřeby vše sloučíme a odešleme.

Rovnice popisující průběh modulace je $s(t) = A \sin(\omega \cdot t + \varphi)$, kde t je čas (to je proměnná), A je amplituda signálu, ω je úhlový kmitočet a φ je fázový posun.

Na obrázku 1.11 na straně 32 jsou ukázky tří základních typů modulace – amplitudové (mění se amplituda), frekvenční (mění se frekvence signálu) a fázové (mění se fáze), vodorovná osa je proměnná t, svislá výsledek $s(t)$. Ve skutečnosti se často mění více než jen jeden z těchto tří parametrů, navíc různé modulace mohou tentýž parametr měnit s různou intenzitou. Podíváme se na několik nejznámějších modulací.

QAM (Kvadraturní amplitudová modulace) používá kombinaci fázového (PSK) a amplitudového (ASK) posunu. Dokáže modulovat jak analogový, tak i digitální signál do příslušného frek-



Obrázek 1.11: Příklad amplitudové, frekvenční a fázové modulace digitálních dat

venčního rozsahu ve výsledném analogovém signálu. Existuje více variant: 16-QAM, 64-QAM, 256-QAM. V každé variantě je u určitého počtu nosných použita amplitudová modulace, u zbývajících fázová, kombinací získáme určitý počet stavů reprezentujících modulační data či signál. Například u nejjednodušší varianty 16-QAM se používají dvě nosné a existuje 36 kombinací, ale využíváno je pouze 16 (těch, které jsou od sebe nejsnáze rozlišitelné).

Modulace QAM v kombinaci s multiplexováním OFDM (viz dále) se dnes používá především v bezdrátových a mobilních sítích, například Wi-fi podle standardu IEEE 802.11n používá až 64-QAM, kdežto IEEE 802.11ac používá až 256-QAM (při špatném signálu „spadne“ na nižší variantu). V mobilních sítích čtvrté generace (LTE Advanced) se používá 256-QAM.

Z dalších modulací můžeme jmenovat například CAP (Carrierless Amplitude/Phase modulation), která se využívá v ADSL, taktéž s multiplexem.

Baseband	**Broadband**
posíláme digitální signál	výsledkem je analogový signál
signál přímo emitujeme	modulační signál modulujeme na nosnou
na kratší vzdálenosti	na delší vzdálenosti
signál využívá celou šířku pásma	signál lze omezit na stanovenou šířku pásma

Tabulka 1.4: Rozdíl mezi přenosem v základním a přeloženém pásmu

Další informace:

- http://www.cs.vsb.cz/grygarek/PS/lect/PREZENTACE/fyzPrincipy.pdf
- http://www.earchiv.cz/l226/slide.php3?l=4

1.5.3 Multiplex

Vezměme jeden přenosový kanál s určitými fyzikálními charakteristikami (frekvence apod.). Za normálních okolností bychom tímto přenosovým kanálem mohli přenášet jen jeden signál.

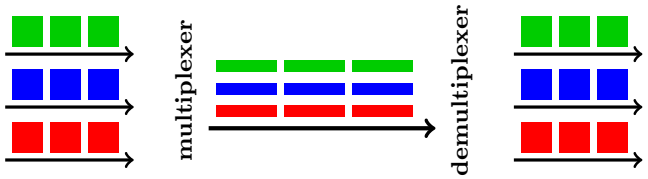
Multiplexing je technika, která umožňuje rozdělit přenosový kanál s dostatečnou šířkou pásma na více logických subkanálů, zdánlivě samostatných, a v každém přenášet jiné bloky dat. V reálu to znamená, že se více signálů sloučí do jediného signálu. Multiplex vlastně znamená „multiple access“, tedy vícenásobný přístup (současný přístup více uživatelů k přenosovému kanálu).

Komponenta, která provádí multiplexing, se nazývá multiplexer (multiplexor, MUX). Opačnou operací k multiplexingu je demultiplexing (DEMUX, DMX): Na odesílající straně přenosového kanálu se provádí multiplexing (rozdělení do subkanálů), na přijímající straně demultiplexing (odebrání ze subkanálů).

Existuje několik běžných typů multiplexování:

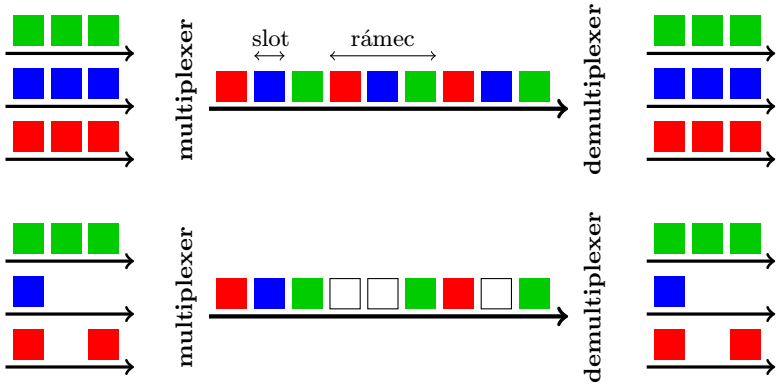
Frekvenční multiplex (FDM, Frequency Division Multiplex): každému přenášenému signálu je přidělena část šířky pásma (mezi přidělenými pásmy musí být odstup, aby nedocházelo k rušení), a tento signál je namapován do této přidělené části. Každý subkanál má tedy přidělen vlastní interval frekvencí.

FDM je použitelný pouze na analogový signál, a další jeho nevýhodou je, že je vhodný spíše pro „stabilní počet“ uživatelů (resp. existuje strop pro množství uživatelů).



Obrázek 1.12: FDM – Frekvenční multiplex

Frekvenční multiplex známe například z rozhlasu, kdy různé stanice mají přiděleny různé frekvence, dále například v satelitních přenosech a kabelových sítích. Také se používal na analogové telefonní síti – každý „telefonista“ měl vyhrazen subkanál o šířce 3.1 kHz.



Obrázek 1.13: TDM – Časový multiplex (nahoře plné vytížení, dole částečné vytížení)

Časový multiplex (TDM, Time Division Multiplex): přenosový kanál je střídavě přidělován různým konkrétním přenosům. Princip je naznačen na obrázku 1.13. Přenosový kanál je rozškálován

na tzv. *rámce* (pozor, to nejsou tytéž rámce jako na vrstvě L2, jen shoda názvů), v každém rámci je pro každého odesílajícího vyhrazen jeden *slot* (zásuvka), do kterého lze umístit paket (nebo ho nechat prázdný).

Problém „čistého“ TDM je, že je taky vhodný spíše pro relativně konstantní počet uživatelů, navíc víceméně podobně komunikujících (co se týče počtu odesílaných paketů). Pokud některý uživatel odesílá výrazně méně paketů, jsou jeho sloty nevyužity, jak vidíme na obrázku 1.13 dole. Další nevýhodou časového multiplexu je nutnost neustálé synchronizace rámců – oběma komunikujícím stranám musí být jasné, kdy začíná a končí rámec a kdy začíná/končí který slot a komu patří. Nicméně na rozdíl od FDM je vhodný i pro digitální signál.

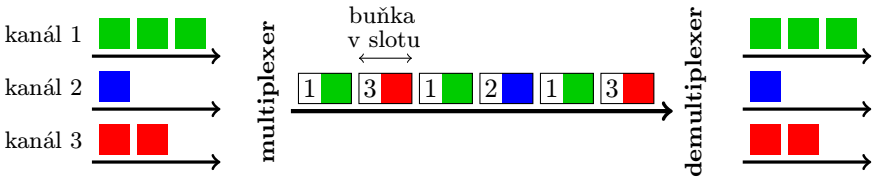
TDM se používá například v mobilních sítích při použití základního přesnosu GSM.

Kódový multiplex (CDM, Code Division Multiplex, také CDMA, Code Division Multiple Access, rozprostřené spektrum) je digitální metoda, jejíž vznik byl motivován především potřebou bezpečnosti (co nejvíc ztížit odposlech). Každý dílčí přenos je zakódován (kódování probíhá v koncových zařízeních, neprovádí je multiplexer), všechny paralelní (zakódované) přenosy jsou pak multiplexerem sloučeny a přenášeny sdíleným kanálem až ke koncovým zařízením. Cílové koncové zařízení pak s pomocí speciálního kódu dekóduje jen to, co mu ve skutečnosti patří, bez tohoto kódu se k obsahu subkanálu nelze dostat.

Problém CDM je potřeba složité synchronizace a horní limit pro počet subkanálů (příliš mnoho subkanálů by se navzájem rušilo a nebylo by možné je dekódovat). Také je nutné zajistit bezpečnou domluvu o kódu pro dekódování. Existuje víc různých variant CDM, většinou se používají v mobilních sítích třetí generace.

FDM, TDM a CDM jsou základní typy multiplexu, ale existují i velmi používané odvozené typy:

Statistický multiplex je podobný časovému multiplexu v tom, že přenosový kanál je členěn na sloty. Ovšem u statistického nejsou sloty napevno přiděleny konkrétním subkanálům, ale jsou přidělovány podle potřeby – vytíženější subkanál dostane víc slotů než méně vytížený. Aby bylo jasné, ke kterému subkanálu který slot patří, musí být k přenášeným blokům dat přidána informace o subkanálu (záhlaví).



Obrázek 1.14: Statistický multiplex

Výhodou statistického multiplexu je o něco nižší potřeba synchronizace (není nutné zabývat se rámci slotů), synchronizují se jen samotné sloty a celková komunikace zůstává asynchronní. Další výhodou je lepší vytěžování přenosového kanálu. Nevýhodou je nutnost opatřovat data záhlavími a s tím spojená režie.

Statistický multiplex se používá v některých WAN sítích, například v ATM (do slotů se skládají datové jednotky zvané buňky).

Vlnový multiplex (vlnové dělení, WDM, Wave Division Multiplex): je to obdoba frekvenčního multiplexu, ale u optického signálu, kde místo frekvencí používáme dělení podle vlnových

délek neboli barev světla (což je, jak víme, vlastně podobné). Každý signál dostane přidělen určitý rozsah vlnových délek, které jsou použity při jeho přenosu. Podle příkladu v předchozí sekci je zřejmé, že u světla se s vlnovými délkami pracuje lépe než s frekvencemi, jsou to menší čísla.

WDM se používá například při přenosu optickým kabelem, tedy především v optických WAN sítích a dále například v technologii FTTx (optické vlákno se vede co nejblíž zákazníkovi). Na signál generovaný laserem nebo LED diodou je po multiplexování modulován signál.

Ortogonální multiplex (OFDM, Orthogonal Frequency-Division Multiplex) mapuje subkanály na různé frekvence podobně jako FDM, ale mnohem efektivněji. Zatímco FDM používá jedinou nosnou pro všechny subkanály, OFDM používá pro každý subkanál jinou nosnou, přičemž jednotlivé nosné jsou navzájem ortogonální, proto se mohou překrývat a přesto je lze na straně přijímače oddělit. Data přenášená přes subkanál jsou modulována na přidělenou nosnou některou vhodnou modulací, většinou se používá některá varianta modulace QAM.

OFDM (a taktéž jeho varianty) se dnes široce používá pro modulaci digitálních dat do analogového signálu v počítačových sítích (Wi-fi, WiMAX, LTE, xDSL apod.), ale také například pro přenos digitální televize.

Existují různé varianty OFDM přizpůsobené konkrétnímu způsobu využití v některé technologii. Například u mobilních sítí se často setkáváme s OFDMA (OFDM Access), kde jsou subkanály jednoho kanálu rozprostřeny po celém spektru a mohou být přidělovány různým klientům.

Další informace:

- http://fyzika.jreichl.com/main.article/view/156-harmonicke-kmitani
- http://www.cs.vsb.cz/grygarek/PS/lect/PREZENTACE/SdileniMedia.pdf
- http://www.wikiskripta.eu/index.php/Elektromagnetick%C3%A9_spektrum
- http://measure.feld.cvut.cz/cs/system/files/files/cs/vyuka/predmety/x38ssl/ofdm.pdf
- https://khanovaskola.cz/video/8/27/1423-amplituda-perioda-frekvence-a-vlnova-delka-periodickeho-vlneni

Kapitola 2

Lokální sítě – Ethernet

Pod pojmem lokální síť (LAN) budeme dále rozumět souhrn navzájem propojených zařízení (hostitelských/koncových zařízení, aktivních síťových prvků apod.), která patří do téže sítě (tj. jako aktivní síťové prvky jsou použity nejvýše switche/přepínače), obvykle v rámci jedné budovy či bytu, s typickou rozlohou jednotek až stovek metrů, výjimečně více.

Rychlý náhled: V současné době jsou nejpoužívanějšími LAN technologiemi Ethernet (coby LAN na měděných nebo optických kabelech) a Wi-fi (bezdrátové). V této kapitole se zaměříme na Ethernet, Wi-fi budeme probírat v samostatné kapitole o bezdrátových a mobilních sítích.

Budeme se zabývat Ethernetem na vrstvách L2 (adresy, formát rámce apod.) a L1 (kabely apod.). Na konci kapitoly probereme technologie související s Ethernetem, jako je napájení přes ethernetový kabel.

Klíčová slova: Ethernet, IEEE 802.3, DTE, DCE, Hub, Switch, přístupová metoda, CSMA/CD, Backoff algoritmus, MAC adresa, fyzická adresa, EUI-48, OUI, LLC, EtherType, SNAP, rámec Ethernet II, kolizní doména, všesměrová doména, UTP, STP, kategorie kabelu, koaxiální kabel, twinax, optický kabel, krimpování, half duplex, full duplex, kolizní okno, burst mode, autonegociace, PoE, strukturovaná kabeláž, patch panel, floor distributor, building backbone, building distributor, campus backbone

Cíle studia: Po prostudování této kapitoly budete umět pracovat s kabely typu UTP, budete rozumět obecně fungování sítě na vrstvách L1 a L2 a konkrétně tomu, jak se přenášejí data s použitím sítě Ethernet.

2.1 Co je to Ethernet

Na původní specifikaci Ethernetu spolupracovaly společnosti Xerox, Digital a Intel, tato specifikace byla zveřejněna roku 1976 a označuje se jako DIX Ethernet (podle počátečních písmen spolupracujících společností).

Později byl Ethernet standardizován jako IEEE 802.3, ale v tomto standardu se název „Ethernet“ vůbec nevyskytuje a s původním DIX Ethernetem je nekompatibilní. Postupně se objevovaly různé

varianty – pro různá přenosová média (metalické kabely různých kategorií, optické kabely) a také se navyšovala rychlost.

Definice (Ethernet, IEEE 802.3)
Síť Ethernet, resp. IEEE 802.3, je standard popisující souhrn technologií pro lokální počítačovou síť používající kabely (metalické nebo optické), kde hostitelská zařízení sdílejí stejnou šířku komunikačního pásma a vzájemně o ni soupeří.

Standard IEEE 802.3 popisuje implementaci pro fyzickou a linkovou vrstvu (tj. celou vrstvu síťového rozhraní podle TCP/IP) včetně metody komunikace a řešení kolizí.

V současné době je v malých LAN sítích nejběžnější buď Fast Ethernet (rychlosti cca 100 Mb/s) nebo Gigabit Ethernet (cca 1 Gb/s). Ve velkých LAN sítích (spíše na páteřích) a vlastně tak trochu i MAN sítích se setkáme s rychlejšími implementacemi (10 Gb/s a více).

2.2 Komunikace v Ethernetu

2.2.1 Typy zařízení v síti

V předchozí kapitole jsme používali pojmy hostitelské zařízení (angl. host) a aktivní síťový prvek. Kromě toho se dále (nejen u Ethernetu) budeme setkávat i s pojmy DTE a DCE.

Definice (DTE, DCE)
DTE (Data Terminal Equipment, terminálové zařízení) je takové zařízení v síti, které je buď příjemcem nebo odesílatelem dat (tedy je nepřeposílá). V síti Ethernet a jiných lokálních sítích se obvykle jedná o koncová zařízení – počítače, servery apod.

DCE (Data Circuit-terminating Equipment, zařízení ukončující okruh) je takové zařízení v síti, které přeposílá provoz (není vlastním zdrojem ani cílem dat), tedy to, co přijme na některém portu, dále odešle na jeden nebo více jiných portů. V síti Ethernet to jsou aktivní síťová zařízení, většinou se jedná o switche (přepínače).

Všimněte si, že v definici je zdůrazněno, že „V síti Ethernet...“. Je tomu tak proto, že v jiných typech sítí se pod pojmy DTE a DCE chápou jiné konkrétní typy zařízení. Například ve WAN sítích (což jsou velké sítě obvykle propojující víc menších sítí) jsou DTE vlastně hraničními zařízeními WAN sítě, tedy taková zařízení, která „rozumí“ příslušnému WAN protokolu, ale jejich úkolem je zprostředkovávat komunikaci s připojenou menší sítí (z pohledu WAN sítě plní podobnou roli jako počítač v lokální síti, jsou „na konci cesty“).

2.2.2 Propojení uzlů v síti

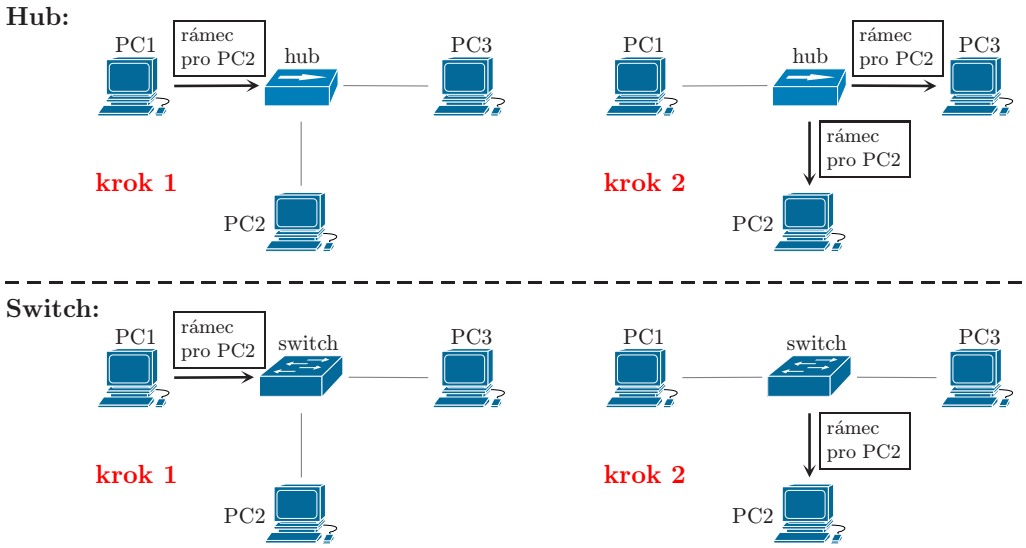
Ještě v 90. letech se jako přenosové médium používal koaxiální kabel (přenosovými médii se budeme zabývat dále v kapitole) a typická fyzická i logická topologie byla *sběrnice*, případně bylo možné více sběrnic propojit pomocí mostů (bridge), případně přepínačů (switch). To znamená:

- Všechna zařízení v síti jsou rovnocenná, žádné nemá prioritu.
- Zařízení připojená na tutéž sběrnici sdílejí přenosové médium, jsou na tomtéž segmentu sítě.

- Jestliže jsou dvě sběrnice propojeny mostem, pak z jedné sběrnice do druhé jde přes most pouze to, co je do druhé sběrnice adresováno, jinak toto vysílání přes most neprojde. Výjimkou je vícesměrné vysílání.
- Správně by mělo v jednom okamžiku vysílat jen jedno zařízení. Pokud toto není splněno, dojde k vzájemnému rušení (kolizi) a oba vysílané signály jsou poškozeny.

Postupně ale došlo k velké změně – jako kabeláž se začala používat kroucená dvojlinka (jako telefonní rozvody) a optický kabel, fyzickou topologií se stává hvězda nebo strom. Logickou topologií zůstává sběrnice. Jako aktivní síťové prvky se původně používaly huby a pomalé routery, ale postupně se prakticky výhradně přešlo ke switchům (přepínačům). Zatímco přes hub prochází cokoliv, co k němu dorazí (tedy je pouze součástí segmentu sítě, neodděluje segmenty), most a switch odesílají signál pouze tam, kam je adresován, a tedy oddělují segmenty.

Na obrázku 2.1 je naznačen rozdíl mezi případem, kdy jsou zařízení propojena hubem, a případem s využitím switche. Jak vidíme, hub opravdu odesílá na všechny porty (kromě toho, ze kterého rámec přišel), nevyužívá informaci o adrese příjemce (vlastně se k ní ani nedostane). Oproti tomu switch zkontroluje adresu příjemce, zjistí, ke kterému portu je příjemce připojen a rámec odešle pouze na ten jeden port.



Obrázek 2.1: Srovnání komunikace zprostředkované hubem a switchem

2.2.3 Přístupová metoda

Přístupová (kolizní) metoda určuje, jakým způsobem se rozhoduje, zda zařízení může či nemůže začít vysílat. Pro Ethernet je specifikována přístupová metoda CSMA/CD.

Definice (Přístupová metoda CSMA/CD)

V Ethernetu se používá technologie vícenásobného přístupu k přenosovému médiu s nasloucháním nosné a detekcí kolizních stavů – CSMA/CD:

- CS (Carrier Sense) – uzly v síti neustále naslouchají na nosné, zda nevysílá jiný uzel,
- MA (Multiple Access) – vícenásobný přístup, tedy kterýkoliv připojený uzel může začít vysílat, pokud nasloucháním zjistí, že nikdo jiný nevysílá,

- CD (Collision Detection) – pokud uzel v síti před vysíláním nestihne včas detekovat vysílání jiného uzlu (například tehdy, když oba začnou vysílat v přibližně stejné době), musí být schopen vzniklou kolizi signálů detekovat a patřičně na ni reagovat.

Co se stane, když nastane kolize:

- Vysílající uzel buď chce vysílat a přitom zjistil, že vysílá jiný uzel, nebo detekoval kolizi (zjistil, že kromě něho vysílá i někdo jiný).
- Pokud už vysílal, přestane vysílat (ne okamžitě, musí umožnit i druhému vysílajícímu uzlu, aby kolizi detekoval). Vyšle *Jam signál*, což je signál oznamující kolizi na médiu.
- Podle speciálního algoritmu (Backoff algoritmus) určí dobu čekání a po uplynutí této doby se pokusí znovu vysílat.
- Jestliže i další pokus selže (buď ještě před vysíláním zjistí, že linka není volná, nebo opět zjistí kolizi), vrací se k předchozímu bodu (doba čekání bez vysílání a nový pokus).

Backoff algoritmus určuje, jak dlouho má uzel čekat s vysíláním, když zjistí, že vysílá jiný uzel. Účelem je zajistit, aby v případě více zájemců o vysílání každý z nich počkal jinou dobu (určenou náhodně generovaným číslem), čímž by se snížila pravděpodobnost další kolize.

Postup (Backoff algoritmus)

Existuje víc různých (různě složitých) variant tohoto algoritmu, základní (Exponential Backoff) je následující:

- Při kolizi vyšle „Jam“ signál, čímž informuje o kolizi a zrušení právě odesílaného rámce.
- Čeká po dobu $0 \ldots 51{,}2$ μs (náhodně vygenerované číslo z tohoto intervalu), pak se znovu pokusí o přenos.
- Jestliže přenos znovu selže, čeká po dobu $2 \times 0 \ldots 51{,}2$ μs (opět se generuje náhodné číslo) a pak se znovu pokusí o přenos.
- Pokud dojde k dalším selháním přenosu, čeká vždy po dobu $K \times 0 \ldots 51{,}2$ μs, kde K je náhodně vygenerované číslo z intervalu $0 \ldots 2^c - 1$, kde c je dosavadní počet kolizí (neúspěšných pokusů o odeslání).

Po 10 pokusech se již c nezvyšuje, pak je vždy K z intervalu $0 \ldots 2^{10} - 1$. Po 16 pokusech postup končí, rámec je považován za nedoručitelný.

Jak vidíme, po prvních dvou selháních přenosu se doba čekání určuje vygenerováním jednoho náhodného čísla, ale od třetího pokusu dále vlastně násobíme dvě náhodná čísla, přičemž pro první z nich se horní mez každým krokem exponenciálně zvyšuje. Tím se snižuje pravděpodobnost, že dva uzly opakují pokus o přenos ve stejnou dobu.

Poznámka:

Uvědomme si, že zatímco fyzická topologie se projevuje na vrstvě L1 (fyzické), logická topologie na vrstvě L2 (linkové). Původní Ethernet používající koaxiální kabel byl sběrnicový na fyzické i logické úrovni (na fyzické díky napojení zařízení na sdílené přenosové médium, na logické díky

přístupové metodě CSMA/CD), novější je sběrnicový jen na logické úrovni, a to jen tehdy, když se používá CSMA/CD.

2.3 Ethernet na linkové vrstvě

Pro připomenutí – na vrstvě L2 (linkové) se jako PDU používají rámce. Při odesílání se na této vrstvě vytváří rámec (k SDU z nadřízené vrstvy se přidá na začátek záhlaví a na konec zápatí) a předá se vrstvě L1, naopak při přijímání se posloupnost bitů přijatá z L1 rozdělí na záhlaví, data a zápatí, analyzují se záhlaví se zápatím a data jsou předána vrstvě L3.

2.3.1 Adresy na vrstvě L2

Na linkové vrstvě se používají MAC adresy (fyzické, hardwarové adresy). V každém rámci je MAC adresa cílového zařízení (adresáta) a MAC adresa zdrojového zařízení (odesílatele). Z toho vyplývá, že odesílatel musí znát svou vlastní adresu a taky adresu cíle (obě obvykle zná).

Jinými slovy – zařízení pracující na vrstvě L2 by mělo mít přehled o *fyzické topologii sítě*, tedy o adresách zařízení, s nimiž může komunikovat, a o tom, přes které síťové rozhraní (kterou cestou) je dotyčné zařízení dosažitelné. Tyto informace si každé zařízení vede ve své tabulce MAC adres (může se nazývat MAC tabulka, CAM tabulka apod., podle toho, jak ji nazve výrobce).

Hardwarová (MAC, fyzická, EUI-48) adresa je vlastně nízkoúrovňovou adresou síťového rozhraní. Tuto adresu má každé síťové rozhraní, a to i tehdy, když zrovna není připojeno k síti. Síťové rozhraní má svou MAC adresu již přidělenu svým výrobcem, a tato adresa by správně měla být celosvětově jednoznačná. Jenže mnohá pravidla mají svou výjimku, toto pravidlo také.

Takže ve skutečnosti máme dva druhy MAC adres:

- adresa přidělená výrobcem (BIA – Burned-In-Address, „vypálená“), která je uložena „napevno“ v ROM nebo flash paměti síťového rozhraní,
- lokální (dočasná) MAC adresa, kterou jsme si nastavili sami.

V současných sítích se používají 48bitové MAC adresy (tj. 6 oktetů) a zapisují se většinou hexadecimálními číslicemi – MAC adresu zapíšeme pomocí 12 hexadecimálních číslic. Jednoznačnost je zajištěna takto:

- První polovinu adresy dostane výrobce přidělenou od sdružení IEEE (velcí výrobci mají přiděleno několik, menším stačí jedna), tedy první polovina je charakteristická pro výrobce a žádní dva výrobci nemají stejnou. Toto číslo se označuje OUI (Organizationally Unique Identifier).
- Druhou polovinu určuje již samotný výrobce, který si hlídá, aby byla tato čísla unikátní v rámci jemu přidělené první poloviny.

V tabulce 2.1 je výše popsaná struktura (dvě části) přehledněji naznačena.

Fyzickou adresu obvykle zapisujeme tak, že dvojice hexadecimálních číslic oddělíme dvojtečkou nebo pomlčkou, nebo po dvou oktetech tečkou, případně bez oddělovačů. Například:

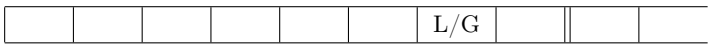
`50:E5:49:A2:80:61` `50-E5-49-A2-80-61` `50E5.49A2.8061` `50E549A28061`

24 bitů	+	**24 bitů**	=	**48 bitů**
OUI = identifikace výrobce	+	identifikace konkrétního výrobku	=	celá adresa
přiděluje IEEE	+	určuje výrobce	=	globálně jednoznačná

Tabulka 2.1: Struktura MAC adresy síťového rozhraní

Změna MAC adresy není až tak běžnou věcí, ale někdy se hodí – například tehdy, když pro účely využití určité technologie či aplikace potřebujeme nutně MAC adresu v určitém konkrétním tvaru a není čas či možnost změnit konfiguraci, nebo ve virtualizovaném prostředí (třebaže i tam mohou být používány unikátní adresy). Hackeři používají pozměněné MAC adresy při pokusu o průnik do sítě chráněné blacklistem nebo whitelistem (seznamy zakázaných/povolených adres).

✎ Každá lokálně platná adresa (tj. ne BIA od výrobce) by správně měla mít nastavený *L/G bit*. Tento bit je v prvním oktetu adresy druhý zprava (v běžných síťových implementacích, například v Ethernetu), jak je vidět na obrázku 2.2.



L/G bit: lokální (1) nebo globální (0) adresa

Obrázek 2.2: Umístění L/G (local/global) bitu v MAC adrese

✎ Pokud jsou všechny hexadecimální číslice nastaveny na F, znamená to, že tato adresa je nejen lokálně platná (není globálně jednoznačná), ale navíc je *všeobecná* (broadcastová), tedy neoznačuje jedno konkrétní zařízení. Vypadá takto: `FF-FF-FF-FF-FF-FF`. Všeobecné MAC adresy se používají například tehdy, když je odesílán rámec určený pro všechna zařízení v lokální síti.

✎ Pokud adresa má označovat skupinu zařízení (ale ne nutně všechna), nazývá se *skupinovou* (multicast) adresou. V Ethernetu a některých dalších sítích používajících MAC adresy poznáme skupinovou adresu podle bitu v prvním oktetu nejvíc vpravo (nejméně významný bit prvního oktetu). U skupinové adresy je nastaven na 1, což znamená, že první oktet bude liché číslo.

I/G bit: unicast (individuální, 0) nebo multicast či broadcast (group, 1)

Obrázek 2.3: Umístění I/G (individual/group) bitu v MAC adrese

2.3.2 Podvrstvy L2

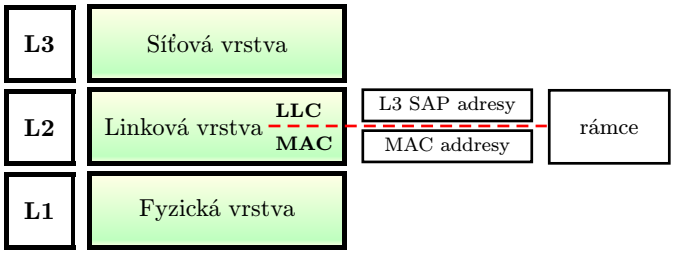
✎ Na vrstvě L2 (linkové) pracují tzv. *spojové protokoly* (protokoly vrstvy L2). Takový protokol buď sám určuje fungování celé vrstvy L2, nebo se používají dva protokoly, které totéž zvládnou společně. Vrstva L2 se totiž ve skutečnosti dělí na dvě podvrstvy:

- LLC (Logical Link Control) – horní podvrstva, zajišťuje spolupráci s vrstvou L3 (síťovou), přidává do rámce informaci o SAP pro spolupráci s L3 a další řídící informace, zde se také provádí multiplexování.
- MAC (Media Access Control) – spodní podvrstva, zajišťuje spolupráci s vrstvou L1 (fyzickou), přidává do rámce fyzickou adresu příjemce a odesílatele, přizpůsobuje strukturu rámce

pro stanovenou přenosovou rychlost a na začátek synchronizační posloupnost bitů (aby bylo zřejmé, kde konkrétně začíná rámec), zde je implementován mechanismus CSMA/CD.

Z funkce těchto podvrstev vyplývá, že na aktivních síťových prvcích, které implementují nejvýše vrstvu L2 (switche, mosty), vlastně ani není nutné mít celou podvrstvu LLC.

Na obrázku vpravo je naznačen vztah podvrstev linkové vrstvy k okolním vrstvám. Jak bylo výše řečeno, v protokolové sadě může buď obě podvrstvy „naplnit“ jeden protokol, nebo jsou použity dva protokoly – jeden pro podvrstvu LLC a druhý pro podvrstvu MAC.



Ethernet může být na vrstvě L2 implementován následovně (co se týče protokolů na L2):

1. Pro podvrstvu LLC je použit protokol IEEE 802.2 (přímo se nazývá LLC), data z nadřízené vrstvy se zapouzdří do LLC rámce. Ten je následně zapouzdřen do MAC rámce podle protokolu IEEE 802.3, čímž vznikne kompletní rámec vrstvy L2.
2. Podobně jako první bod, jen místo protokolu LLC se použije jeho rozšíření SNAP.
3. Data z nadřízené vrstvy (L3) se zapouzdří podle protokolu Ethernet II, který „zvládá“ obě podvrstvy – LLC i MAC.

První dvě možnosti mohou být využívány, ale momentálně se s nimi u Ethernetu téměř nesetkáváme. Protokol IEEE 802.2 se pro implementaci podvrstvy LLC používá spíše v bezdrátových sítích a také ve starších sítích typu Token Ring nebo FDDI. SNAP je vylepšením původního protokolu LLC, umožňuje napojovat do komunikace širší množinu protokolů. Nejčastěji se setkáváme právě s třetí možností, takové rámce se označují *Ethernet II*.

2.3.3 EtherType

Než se zaměříme na strukturu rámce, řekneme si něco o identifikačních kódech určujících, se kterým protokolem nadřízené vrstvy (tedy síťové) vlastně komunikujeme, tedy co konkrétně má být do rámce zapouzdřeno. Pozor, nejde o SAP, nejsou to adresy nadřízených entit.

EtherType je identifikátor určující typ dat, která jsou zapouzdřována do rámce, většinou určuje protokol, který na síťové vrstvě vytvořil odesílaný blok dat. Zapisuje se čtyřmi hexadecimálními číslicemi, takže maximální hodnota je $(\mathrm{FFFF})_{16}$ (zapisujeme 0×FFFF), to je v dekadické soustavě 65 535. Ve skutečnosti existuje i spodní limit – pro EtherType se používají čísla o minimální hodnotě 0×0600, to je 1536. To znamená, že EtherType je vždy z rozsahu 0×0600 . . . 0×FFFF.

Poznámka:

Jak později zjistíme, v záhlaví rámce vytvářeného na vrstvě L2 jsou dva speciální oktety (odtud maximum 0×FFFF), do kterých se obvykle EtherType ukládá. Jenže totéž pole může být použito i pro jiný typ údaje – délku rámce. Ovšem v technice musí být vždy vše naprosto jednoznačné, proto musí být možné rozlišit, zda v daném poli je EtherType nebo délka rámce.

Délka SDU zapouzdřeného do ethernetového rámce je (téměř) vždy menší než 0×05DC (dekadicky 1500), přičteme rezervu (zaokrouhlíme hexadecimálně nahoru) a získáme spodní mez pro

hodnotu EtherType 0×0600.

Pokud je nutné přenést velmi velký rámec (nad stanovený limit), pak se jedná o *jumbo rámec* (jumbo frame) a pro ten je v daném poli speciální hodnota.

Hodnot EtherType je velmi mnoho, podíváme se jen na několik:

- 0×8870: jumbo frames
- 0×8100: VLAN rámce podle 802.1Q
- 0×0800: IPv4
- 0×86DD: IPv6
- 0×0806: ARP
- 0×0835: RARP
- 0×8137, 0×8138: IPX
- 0×8847: MPLS unicast
- 0×8914: FCoE Initialization Protocol
- 0×814C: SNMP

Z těchto protokolů zatím známe především IP verze 4 a IP verze 6.

Poznámka:

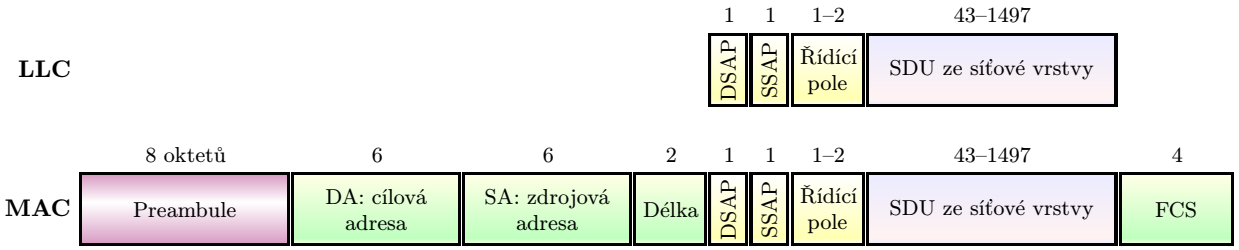
Z názvu „EtherType“ by se mohlo zdát, že tento identifikátor je používán jen v Ethernetu. Sice jako první byl v Ethernetu použit, ale ve skutečnosti se s ním setkáváme i v jiných sítích.

2.3.4 Formát ethernetového rámce

Nyní se podíváme, jak vlastně vypadá rámec posílaný přes síť Ethernet. Vlastně je víc druhů ethernetových rámců:

1. *LLC rámec* podle protokolu IEEE 802.2, tvoří se na podvrstvě LLC a pak je třeba na podvrstvě MAC přidat zbývající informace (to zajistí IEEE 802.3), neboli zapouzdřit do MAC rámce.
2. *SNAP rámec* (podobný LLC, jen má pozměněnou strukturu záhlaví), který se taktéž zapouzdřuje do MAC rámce.
3. *Rámec Ethernet II*, který obsahuje kompletní záhlaví a zápatí vrstvy L2 (LLC a MAC rámec v jednom). Pro identifikaci protokolu nadřízené vrstvy používá EtherType.

Postupně si projdeme všechny tři možnosti. Formát rámce (a později formát různých dalších PDU) budeme reprezentovat formou tabulky, kde v záhlaví je naznačena délka jednotlivých polí, celý PDU si můžeme představit tak, že řádky tabulky naskládáme postupně za sebe.



Obrázek 2.4: Rámec podle IEEE 802.3/802.2: LLC rámec zapouzdřený v MAC rámci podle 802.3

LLC rámec. Struktura LLC rámce (podle protokolu IEEE 802.2) je jednoduchá – k SDU přijaté z nadřízené vrstvy se přidá záhlaví se třemi poli.

Na obrázku 2.4 je naznačen formát rámce v případě, že byly zkombinovány protokol IEEE 802.2 na podvrstvě LLC (vznikl LLC rámec, jeho záhlaví je žluté) a protokol IEEE 802.3 na podvrstvě MAC (LLC rámec jsme zapouzdřili do MAC rámce podle 802.3, záhlaví a zápatí MAC rámce je zelené a fialové). Jednotlivá pole v LLC rámci (žlutém) mají tento význam:

- *SSAP, DSAP* (po 1 oktetu) – přístupové body (SAP) na cílovém (DSAP) a zdrojovém (SSAP) zařízení, které zabírají 7 bitů z každého oktetu; zbývající (nejméně významný, vpravo) bit v každém oktetu označuje:
  - u cílového údaje I/G bit (individual/group) určující, zda jde o skupinovou adresu SAP,
  - u zdrojového údaje C/R bit (command/response) určující typ paketu (příkaz nebo odpověď),
- *řídicí pole* (většinou 1 oktet) – určuje typ rámce z hlediska služby, pořadové číslo apod.,
- data z vyšší vrstvy (SDU), která jsou v tomto rámci zapouzdřena.

Vraťme se k polím DSAP a SSAP. Hodnoty pro tato pole jsou standardizovány, jmenujme si opět některé z nich:

- 0×06: DoD IP
- 0×42: IEEE 802.1 Bridge STP
- 0×98: Arpanet ARP
- 0×E0: Novell Netware (IPX)

Některé další hodnoty jsou používány například pro management LLC (tyto rámce jsou „služební“, s nadřízenou vrstvou nemají nic společného), další jsou vyhrazeny některým již nepoužívaným protokolům.

Příklad

Pokud například LLC rámec obsahuje tyto údaje:

0×42	0×42	0×03	data

Pak to znamená, že na zdrojovém i cílovém zařízení vrstva L2 (její podvrstva LLC) komunikuje s protokolem STP (Spanning Tree Protocol). Číslo 0×42 je binárně 1000010, tedy I/G bit a C/R bit mají hodnotu 0 (individuální SAP, příkaz). Uvnitř (jako „data“) je PDU protokolu STP.

MAC rámec podle IEEE 802.3 přidává tato pole (na obrázku 2.4 zeleně a fialově):

- *Preambule* (8 oktetů) je identifikace začátku rámce, podle preambule se na druhém konci linky pozná, že „začíná rámec“, je to synchronizační informace. Prvních 7 oktetů preambule obsahuje střídající se jedničky a nuly, osmý oktet taky, až na poslední bit – ten je místo na nulu nastaven na jedničku. Celá preambule tedy vypadá takto:

  ```
  10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101011
  ```

  Všimněte si rozdílu v osmém oktetu, resp. na jeho konci. V literatuře se setkáme taky s jinou charakteristikou – preambule na 7 oktetech se střídajícími se jedničkami a nulami, a pak jeden oktet, který toto pravidlo ve svém nejméně významném bitu porušuje. Tento oktet se nazývá SOF nebo SD (Start-of-frame Delimiter).
- *Destination Address* (DA, adresa cíle) a *Source Address* (SA, adresa odesílatele) jsou MAC adresy komunikujících uzlů v síti. Zdrojová adresa musí být vždy unicast, cílová může být unicast, multicast nebo broadcast.

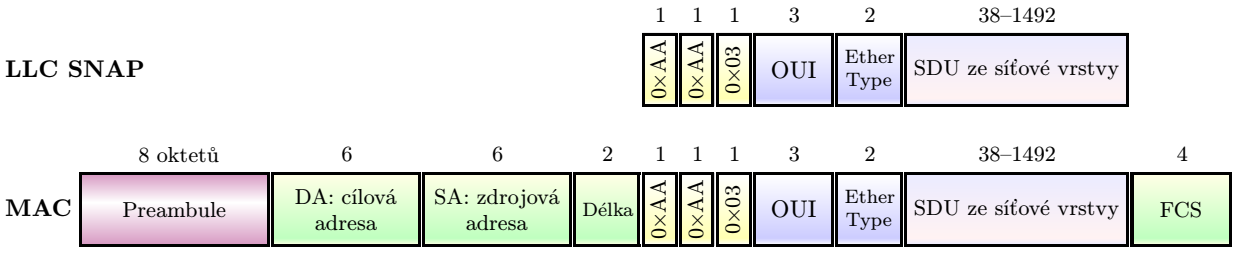
- *Délka* vnořeného LLC rámce (takže délka dat z vyšší vrstvy plus LLC záhlaví).
- *FCS* (Frame Check Sequence) je kontrolní součet, který se vypočítává ze všech předchozích polí (samozřejmě z pole FCS ne, to se teprve tvoří).

Poznámka:
Kontrolní součet FCS (používá se algoritmus typu CRC – Cyclic Redundancy Check) je důležitý: počítá ho jak odesílatel, tak i příjemce. Když cílové zařízení přijme rámec, vypočte si FCS přes stejná pole, jak to provedl odesílatel, a pokud zjistí, že jeho vypočtená hodnota je jiná než ta, kterou najde v zápatí, považuje rámec za poškozený a zahodí ho.

Je zajímavé, že v tomto poli jsou bity uloženy v opačném pořadí než v jiných polích, a navíc pozice tohoto pole se dá zjistit algoritmicky (takže technicky vlastně ani není třeba mít v některém poli délku rámce nebo délku payloadu).

Do LLC rámce podle IEEE 802.2 se obvykle zapouzdržují datové jednotky „provozních“ protokolů na L2, jako je například výše zmíněný STP, nebo třeba LLDP (Link Layer Discovery Protocol). LLDP je otevřený standard (tj. mohou ho implementovat různí výrobci, a taky tak činí), přes který se zařízení na vrstvě L2 domlouvají a navzájem si sdělují své parametry.

SNAP rámec. SNAP (SubNetwork Access Protocol) vznikl rozšířením protokolu IEEE 802.2 LLC, pozměněním a prodloužením záhlaví podvrstvy LLC. SNAP rámec vypadá takto:



Obrázek 2.5: Rámec IEEE 802.3/SNAP: SNAP rámec zapouzdřený v MAC rámci podle 802.3

Na obrázku 2.5 jsou pole přidaná rozšířením SNAP vybarvena modře. Jak vidíme, SNAP dosazuje do záhlaví protokolu LLC určité konstantní hodnoty – jako DSAP a SSAP se použije kód 0×AA (nebo je přípustný kód 0×AB) a do řídícího pole se uloží 0×03. Informaci, kterou by nám jinak tato pole dávala (tedy konkrétní protokoly na nadřízené vrstvě), najdeme právě v přidaných modrých polích.

Rozšíření SNAP přidává k LLC tato pole (na obrázku 2.5 modře):

- *OUI* (Organizationally Unique ID, organizace) je většinou nastaveno na 0. Pokud ne, pak se jedná o identifikátor organizace, v rámci jejíchž zařízení je tento rámec posílán. Například Cisco má OUI = 0×00 00 0C.
- *EtherType* (nebo obecněji Typ) určuje protokol nadřízené vrstvy, pokud je OUI = 0. Pokud není, pak je toto pole čistě v režii organizace, jejíž OUI je v předchozím poli, například pro přenos paketů podle proprietárních síťových protokolů.

O dvě stránky výše je seznam několika běžných hodnot pro pole SSAP a DSAP v LLC rámci. K těmto hodnotám si můžeme přidat ještě 0×AA jako identifikátor rozšíření SNAP.

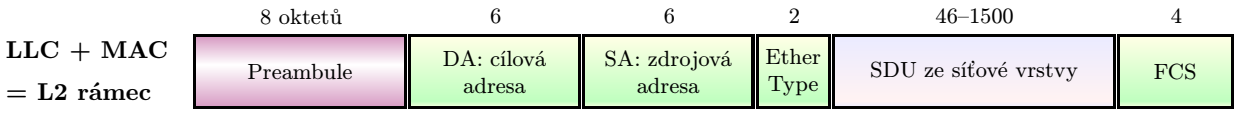
Poznámka:

Jak poznáme, jestli se jedná o rámec IEEE 802.3/LLC nebo IEEE 802.3/SNAP?

Několik polí (záhlaví MAC rámce podle 802.3) je stejných, odlišnost začíná na začátku vnořeného LLC/SNAP rámce. Pokud v „žluté části“, resp. třech oktetech za MAC záhlavím, najdeme hodnotu 0×AAAA03, pak je jasné, že jde o SNAP rámec. Jinak jde o LLC rámec.

Do rámce typu SNAP se dnes zapouzdřují spíše datové jednotky proprietárních protokolů (běžících na L2). Například Cisco má svůj protokol CDP (Cisco Discovery Protocol), který má stejný účel jako výše zmíněný LLDP (zařízení od Cisca umí oba tyto protokoly). Výhodou SNAP záhlaví je, že dokáže bez problémů transportovat i proprietární protokoly včetně informace o typu protokolu.

Rámec Ethernet II. V současné době se v Ethernetu pro data (tj. typicky pro IP pakety) používá téměř výhradně tento typ rámce, a není divu – je nejjednodušší a nejefektivnější.



Obrázek 2.6: Rámec Ethernet II

Na obrázku 2.6 vidíme, že struktura rámce je vlastně velmi podobná tomu, co v předchozích případech na MAC podvrstvě dodával protokol IEEE 802.3. Pole pro preambuli, obě MAC adresy a kontrolní součet mají stejný význam.

Rozdíl je jen v poli, ve kterém byla v předchozích případech délka rámce – v rámci Ethernet II tam najdeme hodnotu EtherType. Proč? Evidovat délku rámce vlastně ani nepotřebujeme. Konec rámce jednoduše poznáme tak, že na přenosovém médiu nebude žádný signál (přenáší se v základním pásmu, kdy je signál na médiu generován, nikoliv modulován existující).

Poznámka:

Podle čeho poznáme, že se jedná o rámec Ethernet II a ne žádný z předchozích?

V předchozí sekci je psáno o hodnotách a významu pole EtherType. Kromě jiného se tam v jedné poznámce dočteme, že EtherType a délka rámce jsou ukládány do téhož pole, přičemž číslo větší než 0×0600 znamená EtherType, menší znamená délku rámce. Takže přepínač během načítání prvních $8+6+6=20$ oktetů ještě typy rámců nedokáže rozlišit, ale hned podle 21. a 22. oktetu už dokáže rozlišit, zda se jedná o Ethernet II.

Další informace:

- https://www.ionos.com/digitalguide/server/know-how/ethernet-frame/
- http://ciscopub.blogspot.cz/2015_01_01_archive.html

Poznámka:

Pokud se někde uvádí skutečná, minimální či maximální délka rámce, preambuli do ní obvykle nepočítáme. Když se vrátíme k nákresům struktury jednotlivých typů rámců a sečteme čísla uve-

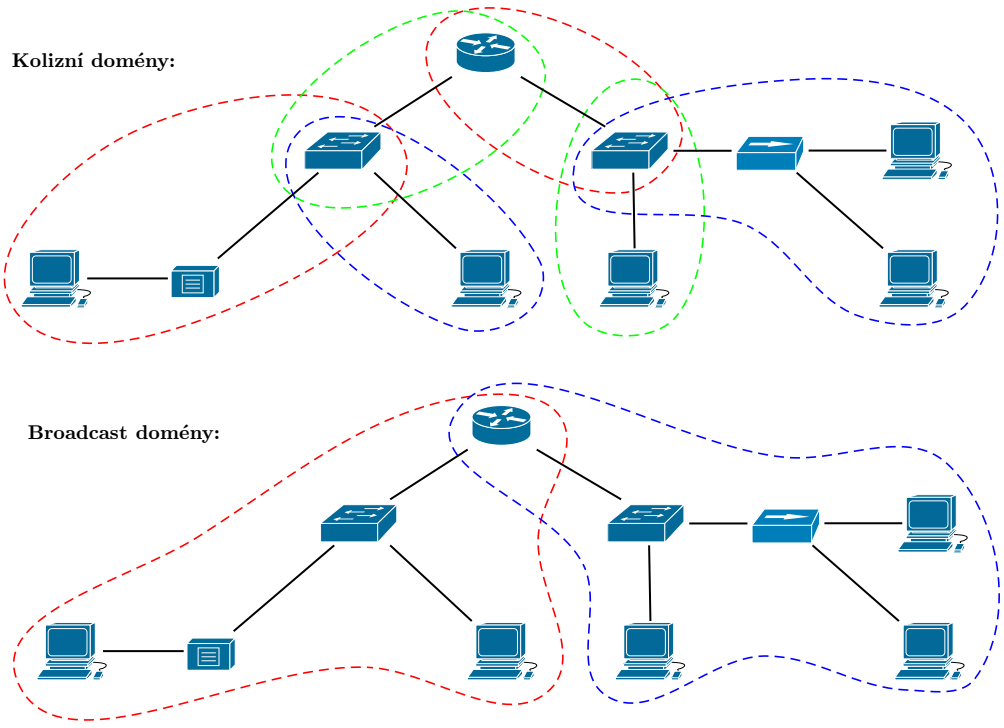
dená nad jednotlivými poli (délky těchto polí) kromě preambule, zjistíme, že ve všech případech je minimální délka rámce $46 + 18 = 64$ oktetů, maximální délka je $1500 + 18 = 1518$ oktetů.

2.3.5 Kolizní a všesměrová doména

Definice (Kolizní doména a všesměrová doména)

Kolizní doména je souhrn zařízení v síti, mezi nimiž může dojít k vzájemným kolizím v komunikaci. Switch (a taky router) dokáže filtrovat provoz a unicast rámce pouští jen na ten port, ke kterému je připojeno cílové zařízení, proto odděluje kolizní domény (na komunikačním kanálu není žádná cizí komunikace – pro každé zařízení se jedná buď o rámec, který vyslalo, nebo rámec, jehož je cílem). Hub neodděluje kolizní domény (naopak je spojuje), protože rámec přeposílá na všechny porty. Kolizní doména obvykle odpovídá segmentu sítě.

Všesměrová (broadcastová) doména je souhrn zařízení v síti, která mohou navzájem přijmout všesměrové (broadcast) rámce – pokud kterékoliv z těchto zařízení vyšle všesměrový rámec, všichni ostatní tento rámec dostanou. Protože switch a hub přeposílají všesměrové rámce na všechny porty (kromě toho, ze kterého rámec přišel), neoddělují všesměrové domény. Naopak router všesměrové rámce nepropouští (zahazuje), tedy odděluje všesměrové domény. Všesměrová doména odpovídá pojmu síť.



Obrázek 2.7: Kolizní a všesměrové domény vzhledem k různým síťovým prvkům

Na obrázku 2.7 je naznačen význam pojmů kolizní a všesměrová doména vzhledem k použití různých aktivních síťových prvků. Jak vidíme, stačí místo hubu použít switch, čímž se na logické

úrovni všechny spoje stávají typu point-to-point, a problém s kolizemi je téměř odstraněn, protože každá kolizní doména se skládá pouze ze dvou zařízení (z toho jedno je switch). CSMA/CD se používá, ale problém s kolizemi značně minimalizujeme.

2.4 Ethernet na fyzické vrstvě

Na fyzické vrstvě v Ethernetu jsou zajištěny tyto operace:

- spolupráce s vrstvou L2, přebírání SDU z této vrstvy (tedy rámce),
- kódování, multiplexing,
- vysílání a přijímání signálu, synchronizace se zařízením „na druhé straně“,
- auto-negociace (vyjednávání) – síťové rozhraní si potřebuje dojednat se síťovým rozhraním druhého zařízení veškeré přenosové parametry (rychlost, poloviční nebo plný duplex apod.), aby si navzájem rozuměly.

2.4.1 Kabely

Ethernet je, co se týče kabelů, poměrně variabilní. Z metalických kabelů jde obvykle o nestíněnou kroucenou dvojlinku (UTP), ale existují i standardy pro stíněnou kroucenou dvojlinku (STP) a starší pro koaxiální kabel. Na větší vzdálenosti (na páteřní sítě) se mohou použít optické kabely.

✎ **UTP (Unshielded Twisted Pair).** Nestíněná kroucená dvojlinka je metalický kabel odvozený z telefonního kabelu. Správně má mít čtyři páry vodičů (tj. celkem osm vodičů), ale setkáváme se i s „ošizenými“ kabely obsahujícími pouze dva páry vodičů. Tyto „light“ kabely jsou použitelné, ale nedosáhneme s nimi na vyšší rychlosti (pouze do 100 Mb/s). Kabely mají pouze základní stínění (každý vodič má barevnou plastovou izolaci a celý kabel je v plastovém obalu).

Vodiče jsou v párech, dvojice vodičů v páru je do sebe zkroucená (od toho název, vlastně bychom správně měli říkat „svazek kroucených dvojlinek“). Důvodem zkroucení je snížení příjmu elektromagnetického rušení z okolí a taky naopak snížení vyzařování do okolí (páry ve svazku by se neměly rušit) – potlačení „anténového efektu“. Čím vyšší rychlost, tím vyšší vyzařování, proto je pro vyšší rychlosti nutný kvalitnější kabel.

✎ Pro přenos signálu je vždy nutný celý pár (nestačí jeden vodič), protože signál je charakterizován jako *rozdíl potenciálů* dvou vodičů v páru. Standardy definují, které páry jsou k jakému účelu používány (pro plný duplex je obvykle používán minimálně jeden pár pro každý směr).

✎ **STP (Shielded Twisted Pair), FTP a spol.** Stíněná kroucená dvojlinka má kromě plastové ochrany přidáno stínění, a to buď stínění párů (každý pár je navíc obalen hliníkovou fólií) nebo celého kabelu (všechny páry dohromady jsou obaleny hliníkovou fólií nebo kovovým opletením) nebo obojí. Na stíněné kabely se koncovka nasazuje trochu jinak, vzhledem k nutnosti uzemění kabelů. Pokud kabel neuzemníme, kovové stínění (zejména ve formě kovového opletení) funguje jako anténa a způsobuje rušení.

✂ Ve zkratkách názvů kabelů se používá „F“ pro obalení jednotlivých párů nebo všech párů hliníkovou fólií (foil), „S“ pro opatření svazku dvojlinek kovovým opletením. Například U/FTP znamená, že každý pár má vlastní fólii, F/UTP znamená, že všechny páry dohromady jsou zabalené do fólie, F/FTP obojí (fólie kolem dvojlinek i svazku), S/UTP je kabel s kovovým opletením svazku

Kategorie	Použití
Cat.1, Cat.2	V současné době se nepoužívá.
Cat.3	Většinou telefonní rozvody. V počítačových sítích pro starý 10Mb Ethernet.
Cat.4	Pro sítě Token Ring. V současné době se prakticky nepoužívá.
Cat.5	Pro Fast Ethernet (100 Mb/s). Některé kvalitnější kabely se dají použít i pro Gigabit Ethernet, ale nedoporučuje se.
Cat.5e	Pro Fast Ethernet a Gigabit Ethernet. V současné době nejpoužívanější kabeláž pro lokální sítě.
Cat.6	Pro Gigabit Ethernet a vyšší.
Cat.6_A	Podobně jako Cat.6, ale má větší šířku pásma (500 MHz) a i v rychlosti 10 Gb/s přenáší až na vzdálenost 100 m. Tyto kabely často seženeme ve stíněné verzi.
Cat.7	Pro 10G Ethernet a vyšší, na rozdíl od Cat.6_A je plně stíněný a nabízí ještě vyšší šířku pásma (600 MHz).
Cat7_A	Další vylepšení, ještě větší šířka pásma (1000 MHz).
Cat.8, 8.1, 8.2	Pro Ethernet o rychlosti 40 Gb/s, šířka pásma je až 2000 MHz.

Tabulka 2.2: Kategorie kabelů typu kroucená dvojlinka

dvojlinek, atd. pro další kombinace. Někteří výrobci se tohoto označení moc nedrží, pod S-STP se často skrývá stínění hliníkovou fólií pro jednotlivé páry i celý svazek.

✎ Existuje víc *kategorií* TP (Twisted Pair) kabelů, které se liší počtem párů (u nižších kategorií), mírou zakroucení párů, šířkou přenosového pásma, stíněním celého kabelu (izolací), obvyklým přídavným stíněním, použitelnými konektory (u nejvyšších kategorií), atd. Týká se to jak stíněných, tak i nestíněných kabelů.

Obecně platí, že čím vyšší kategorie, tím vyšší kvalita pro větší vzdálenosti. V tabulce 2.2 na straně 49 je přehled kategorií, přičemž většina řádků se právě týká UTP (nestíněné kroucené dvojlinky). Momentálně nejpoužívanější kategorie je Cat.5e, v novostavbách se prosazuje vyšší kategorie (doporučuje se kategorie Cat.6_A). Vyšší kategorie již vyžadují jiný typ konektoru (navíc je dražší jak kabel, tak i konektory), proto se zatím moc neprosazují.

Kroucená dvojlinka se používá pro přenos v základním pásmu (baseband). Jako konektor se do kategorie 6_A používá RJ-45 s osmi piny, který se správně označuje 8p8c. Pořadí vodičů v konektoru se až do rychlosti 100 Mb/s řídí standardem TIA/EIA 568-B nebo TIA/EIA 568-A, ale pokud kabel chceme používat pro vyšší rychlosti, je třeba použít standard ANSI/TIA 568-C.

✎ **Koaxiální kabel.** Koaxiál je silný kabel se dvěma vodiči – první vodič je středový pevný drát, druhý vodič je tvořen kovovým opletením (takže zase tady máme dva vodiče). Mezi těmito dvěma vodiči je nevodivá izolace (dielektrikum), obvykle z plastu. Je souosý (což se anglicky řekne coaxial), tedy oba vodiče mají jednu společnou osu a signál je určen rozdílem elektrických potenciálů těchto vodičů.

Typickou fyzickou topologií pro koaxiál je sběrnice – na jeden kabel se připojuje celá řada zařízení, přičemž pro každé zařízení musí být vytvořena „odbočka“ z hlavního kabelu. Oba konce hlavního kabelu je třeba opatřit ukončovacími členy (terminátory), které pohlcují signál. Kdybychom to neudělali, signál by se na konci kabelu odrážel zpět a vznikalo by rušení. Šířka pásma

je větší než u kroucené dvojlinky, tedy je možné použít přenos v překládaném pásmu (broadband), ale většinou se v počítačových sítích přenášelo v základním pásmu.

Výhodou je lepší stínění. Podstatnou nevýhodou je však celkově horší manipulace s kabelem (koaxiál je silný a málo pružný, navíc krimplování konektorů je náročnější). Další typickou vlastností (někdy výhodou, jindy nevýhodou) je, že je to typické *sdílené médium* – může bý připojeno více než jen dvě zařízení a signál se vždy šíří oběma směry (kroucená dvojlinka je naproti tomu vyhrazené médium, kde lze sdílení implementovat jen pomocí rozbočovačů).

V současné době se koaxiál už pro Ethernet nepoužívá, najdeme ho především v televizních rozvodech (propojuje televizory s anténou či satelitem) nebo u bezdrátových řešení k připojení externí antény. Zatímco v rozvodech pro počítačové sítě včetně antén používáme koaxiál s impedancí (měrným odporem) 50 Ω, v televizních rozvodech se používají koaxiály s impedancí 75 Ω. Pro daný účel je vždy nutné použít kabel s tou impedancí, která tam má být, jinak nebude možné signál správně přenést.

✎ **Optický kabel.** V optickém kabelu vede jedno nebo více optických vláken. Optickým vláknem (fiber-optic) vedeme optický signál, tedy světelné impulsy. Na odesílající straně je generátor světelného impulsu, na přijímající straně je fotodetektor, který tento impuls přijme (rozpozná) a buď pošle dál optickým systémem nebo konvertuje na elektrický signál. Zatímco při přenosu elektromagnetických vln se informace reprezentuje například úrovněmi napětí, u optického systému máme vlnové délky světla.

Optický signál může být generován buď laserem (dražší, výkonnější a přesnější) nebo LED diodami (levnější, méně výkonné, méně přesné). Přenášíme v základním pásmu (baseband). Protože je rychlost světla výrazně vyšší než rychlost elektromagnetických vln, je možné tímto kabelem přenášet data na mnohem větší vzdálenosti.

✎ Rozlišujeme *jednovidové* (single-mode) a *mnohavidové* (multi-mode, taky vícevidové) optické vlákno. Mnohavidové kabely jsou levnější (nejen samotné kabely, ale především generátory a detektory světelných impulsů), ale na druhou stranu je jejich maximální délka i řádově menší než u jednovidových kabelů. Typický dosah mnohavidového vlákna je ve stovkách metrů až jednotkách kilometrů, u jednovidového desítky kilometrů.

✎ Srovnání UTP a optických vláken:

- přenáší se optický signál vs. elektromagnetický signál,
- UTP jako kabel je levnější,
- generování a detekce signálu je u UTP značně jednodušší a levnější než u optického vlákna,
- signál v optických vláknech má jen minimální odpor, kdežto u UTP je odpor mnohem vyšší,
- důsledkem předchozího je velký rozdíl v dosahu signálu (vzdálenost, na kterou je garantován relativně nerušený přenos) – u kroucené dvojlinky jen cca 100 m,
- u optického vlákna nehrozí elektromagnetické rušení.

2.4.2 Křížení a krimpování

Zařízení na svém síťovém rozhraní vlastně provádí dvě základní operace:

- Tx (Transmit) – odesílá,
- Rx (Receive) – přijímá (tj. naslouchá a když zjistí, že druhá strana vysílá, toto vysílání přijme).

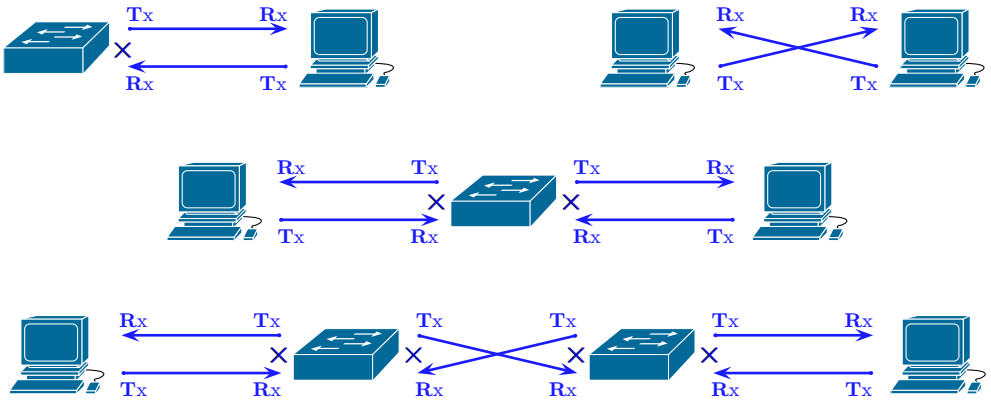
V reálu to znamená, že například u kroucené dvojlinky jsou páry vodičů vyhrazené pro operaci Tx a páry vodičů vyhrazené pro operaci Rx, případně je toto přiřazení určováno dynamicky (ale vždy platí, že v jednom okamžiku je toto přiřazení jednoznačné) – daný pár vodičů nemůže v jednom okamžiku provádět totéž. Jinými slovy – síťové rozhraní se skládá ze dvou částí – Tx a Rx, a je důležité, na kterou z těchto částí je který pár v kabelu napojen.

U dvojice přímo komunikujících zařízení (tj. propojených tímtéž kabelem) musí platit, že na konkrétním páru jedno zařízení přijímá a druhé vysílá – kdyby na tomtéž páru obě zařízení odesílala, došlo by ke kolizi, a kdyby na tomtéž páru obě zařízení naslouchala, k žádnému přenosu by naopak vůbec nedošlo. Vysílající zařízení odešle signál do části Tx svého síťového rozhraní, a cílové zařízení tento signál přijme na části Rx svého síťového rozhraní, a tyto dvě části musí být fyzicky propojeny tímtéž párem vodičů.

Poznámka:

Z toho vyplývá, že někde musí být přenosová cesta *překřížena*. Pokud jsou dvě zařízení propojena kroucenou dvojlinkou, pak (minimálně jeden) pár vodičů vedoucí z Tx jednoho zařízení musí být napojen za Rx druhého zařízení a naopak. Křížení bývá zajištěno takto:

- v kabelu/konektoru (každá koncovka je jinak nasazena),
- v síťovém rozhraní (to provede například switch, když ho propojíme s koncovým zařízením), tedy v síťovém rozhraní je na přehozen význam zakončení Tx a Rx,
- přímo v síťovém zařízení – zařízení přijme rámec ze zakončení Rx v jednom síťovém rozhraní (portu), podle potřeby ho zpracuje a odešle na zakončení Tx v jiném síťovém rozhraní.



Obrázek 2.8: Princip křížení na kroucené dvojlince

Na obrázku 2.8 vidíme několik řešení – v „první řadě“ křížení zajistí síťové rozhraní aktivního síťového prvku nebo je přímo v kabelu – křížení zajištěné síťovým rozhraním je naznačeno křížkem. Níže jsou případy, kdy máme jedno nebo dvě mezilehlá zařízení (která zajišťují křížení).

Rozlišujeme dva typy kabelů typu kroucená dvojlinka:

- *přímý kabel* – žádné křížení neobsahuje, používá se obvykle k propojení DTE a DCE (tedy například počítače ke switchi),

- *křížený kabel* – obsahuje křížení, používá se k propojení zařízení stejného typu (tedy buď DTE a DTE, nebo DCE a DCE); pokud chceme propojit dva počítače přímo bez mezilehlého DCE, měli bychom použít křížený kabel.

Ve skutečnosti se v obou případech používá tentýž kabel; to, zda je přímý nebo křížený, se určuje nasazením koncovek (konektorů). Přímý kabel má na obou stranách koncovky nasazeny stejně (vodiče jsou na obou koncích ve stejném pořadí), kdežto křížený kabel má na jednom konci vodiče jinak seřazené než na druhém konci.

Poznámka:
V současné době máme situaci trochu jednodušší – novější síťová rozhraní většinou dokážou automaticky rozpoznat, jestli mají nebo namají zajistit vnitřní křížení, takže ve většině případů můžeme prostě použít přímý kabel.

Krimpování (konektorování) je postup nasazování konektoru na kabel. V případě kroucené dvojlinky se většinou používají konektory a zásuvky typu *8p8c* (to znamená 8 pozic a 8 kontaktů), kterým se poněkud nepřesně říká RJ-45.

V konektoru (i zásuvce) jsou vodiče umístěny v řadě za sebou a jejich pořadí je určeno standardem TIA/EIA-568-B z roku 2001, nověji TIA/EIA-568-C z roku 2014. Standard ve skutečnosti určuje dvě pořadí – T568A a T568B (pozor, neplést si písmenka s verzí standardu), přičemž u přímého kabelu se na obou koncích použije pořadí podle T568B, kdežto u kříženého na každém konci jedno z těchto pořadí. Pozor, nejde jenom o obrácení pořadí!

Poznámka:
Na cvičeních si všichni studenti vyzkouší krimpování konektoru 8p8c na kabel, a ve skriptech pro cvičení také najdete víc informací včetně postupu.

V případě optických vláken je křížení zajištěno vždy v kabelu (zde jde o pořadí optických vláken).

2.4.3 Parametry

Naprostá většina druhů Ethernetu používá přenos v základním pásmu (baseband). Dalšími důležitými parametry jsou rychlost a typ přenosového média (kabelu), obojí má vliv na použité kódování a průběh multiplexování. Na fyzické vrstvě se v označení objevuje:

- rychlost: 10, 100, 1000 apod. – v Mb/s, u vyšších se píše s písmenem G, např. 10G,
- přenos v základním (base) nebo přeloženém (broad) pásmu,
- typ kabelu – většinou „T“ znamená kroucenou dvojlinku (twisted), „F“ optický.

Takže pojďme od historie, ale přeskočíme nejstarší standardy. Nebudeme se zabývat všemi, u každé rychlosti vybereme jen některé typické zástupce.

Nejdřív *10Mb Ethernet*. Všichni zástupci dovolují rychlost 10 Mb/s, další parametry:

- 10Base-T: baseband, kabel nestíněná kroucená dvojlinka (UTP),
- 10Base-F: baseband, kabel optický,

- 10Broad-36: broadband (přeložené pásmo), kabel koaxiál o impendanci 75 Ω; nerozšířil se, ale stal se základem pro pozdější Ethernet provozovaný v sítích kabelových televizí (dnes se místo toho v kabelových televizích používá pro přenos dat standard DOCSIS).

Pokračujeme *Fast Ethernetem*. Pro všechny zástupce platí rychlost 100 Mb/s a baseband, oproti pomalejšímu předchůdci přibyla možnost auto-negociace (síťová rozhraní se dokážou automaticky dohodnout na parametrech přenosu). Další parametry:

- 100Base-TX: kabel UTP alespoň Cat.5, přičemž využívá pouze dva páry (jeden pro vysílání a druhý pro příjem); vpodstatě pokračovatel 10Base-T, v síti bylo možné kombinovat síťové karty těchto typů,
- 100Base-FX: používá se optické vlákno, ve skutečnosti dvě – jedno pro každý směr. Není kompatibilní s 10Base-F, protože používá světelný signál o jiné vlnové délce.

Gigabit Ethernet umožňuje rychlost 1 Gb/s, baseband. Další parametry:

- 1000Base-T: pro kabel UTP minimálně kategorie 5 (doporučeno minimálně 5e), používají se všechny čtyři páry (takže žádný light se dvěma páry nelze použít), směr pro jednotlivé páry se určuje adaptivně – žádný není vyhrazen pro konkrétní směr,
- 1000Base-TX: jedná se o nepříliš úspěšný pokus o úpravu standardu 1000Base-T tak, že pro každý směr jsou vyhrazeny dva páry a je vyžadován kabel kategorie 6; zatímco 1000Base-T je standard od organizace IEEE (IEEE 802.3ab), 1000Base-TX pochází od TIA (TIA/EIA-854); komerčně neúspěšný – nelze použít na starších rozvodech kategorie 5 nebo 5e,
- 1000Base-X: souhrn standardů pro (většinou) optická vlákna
  - 1000Base-SX („S“ jako short) používá optická vlákna mnohavidová, typický dosah je ve stovkách metrů,
  - 1000Base-LX („L“ jako long) používá optická vlákna buď jednovidová (dosah v jednotkách kilometrů) nebo mnohavidová (stovky metrů),
  - 1000Base-CX – nepoužívá optiku, ale stíněnou kroucenou dvojlinku; typický dosah je do 25 metrů, používá se v datových centrech k připojení serverů ke switchům (například u některých produktů společnosti IBM).

Poznámka:
Velmi často se setkáváme se záměnou názvu – místo 1000Base-T je udáno 1000Base-TX. Obvykle to je proto, že u „dřívější generace“ (tedy Fast Ethernetu) byla přípona TX. U síťových rozhraní podporujících různé rychlosti bývá označení 100/1000Base-TX, třebaže u druhé uvedené rychlosti jde ve skutečnosti o „Téčkový“ standard.

10 Gigabit Ethernet (10GE, 10GigE) umožňuje rychlost 10 Gb/s, přenos baseband. Přístupová metoda CSMA/CD je zcela odbourána, komunikuje se pouze ve full-duplexu.

- 10GBase-R je skupina standardů pro optická vlákna:
  - 10GBase-SR (short range) pro mohavidové vlákno, dosah do 62 m,
  - 10GBase-LR (long range) pro jednovidové vlákno, dosah cca 10 km,
  - 10GBase-LRM (long reach multimode) pro mnohavidové vlákno, dosah do 220 m,
  - 10GBase-ER (extended range) pro jednovidové vlákno, dosah 40 km,

  liší se nejen kabely, ale také vlnovými délkami, na kterých je vysílán signál,

- 10GBase-CX4 používá kabel twin-ax (podobně jako koaxiál, jen místo jednoho středového vodiče jsou dva), dosah je jen 15 metrů, ale na druhou stranu má velice dobré hodnoty odezvy (latence) a cena je vcelku příznivá; používá se v datových centrech pro připojení serverů k DCE,
- 10GBase-T je pokračovatelem 1000Base-T, používá kroucenou dvojlinku minimálně Cat.6; doporučuje se však použít vyšší kategorii, protože na Cat.6 má dosah jen max. 55 m,
- 10GBase-W už míří mimo LAN sítě – na vrstvu L1 přidává novou podvrstvu WIS, která umožňuje komunikovat s WAN sítí (SONET/SDH, viz dále); funguje podobně jako 10GBase-R, jen navíc ethernetový rámec zapouzdří do WIS rámce, ve kterém data procházejí napojenou WAN sítí.

Poznámka:

Zvídavý čtenář se určitě ptá: a kde jsou tedy ty standardy? Tak například 100Base-TX a 100base-FX (tedy kroucená dvojlinka a optika pro Fast Ethernet) jsou standardizovány v IEEE 802.3u, kdežto 1000Base-T (kroucená dvojlinka pro Gigabit Ethernet) je v standardu IEEE 802.3ab a Gigabit Ethernet na optice najdeme v IEEE 802.3z.

Standard IEEE 801.3 a jeho dodatky (písmenka na konci) definují vrstvu L1 a spodní část vrstvy L2 (MAC podvrstvu), přičemž se v praxi využívá spíše to, co platí pro vrstvu L1 (už dřív jsme si vysvětlili, že na L2 se setkáváme spíše s rámci podle Ethernet II).

2.5 a 5Gigabit Ethernet byly jako standardy zveřejněny až v roce 2016, tj. po „rychlejších“ sourozencích. Jedná se o komunikaci po kroucené dvojlince, formálně 2.5GBase-T a 5GBase-T, standard IEEE 802.3bz. Fyzická vrstva byla převzata z 10GBase-T a přizpůsobena „horší“ kabeláži. 2.5G-Base-T může využívat nestíněnou kroucenou dvojlinku kategorie Cat.5e, 5GBase-T používá kategorii Cat.5e (menší vzdálenost než 100 m) nebo Cat.6.

40 Gigabit Ethernet a 100 Gigabit Ethernet (40GigE, 100GigE) předepisuje optické kabely s laserem jako zdrojem světla (na vzdálenost cca 100 m pro mnohavidová vlákna, resp. kilometry pro jednovidová), na vzdálenost několika metrů kabel twinax, a na vzdálenost do 30 m lze pro rychlost 40GigE použít kroucenou dvojlinku Cat.8.

Poznámka:

Jak je možné řádově zvyšovat rychlost, když se pořád používají vpodstatě tytéž kabely (i když generaci od generace lepší)? Zvýšení se dosahovalo kombinací více způsobů, například:

- vytížením všech čtyř párů kroucené dvojlinky místo pouhých dvou,
- použitím kvalitnější kabeláže s lepším stíněním, kroucením s vyšší hustotou a větší šířkou přenosového pásma,
- použitím lepšího kódování dat na signál (v reálu se přenáší menší objem dat).

2.5 Průběh přenosu

V předchozí kapitole jsme se dozvěděli, že obousměrný přenos mezi dvěma zařízeními může probíhat jedním z těchto způsobů:

- v poloviční duplexu (half duplex), kdy vysílat mohou obě zařízení, ale střídavě (v jeden okamžik může vysílat jen jedno zařízení), tedy sdílejí tentýž komunikační kanál,
- v plném duplexu (full duplex), kdy mohou vysílat třeba i obě zařízení najednou, protože pro každý směr je vyhrazen (minimálně) jeden komunikační kanál.

2.5.1 Přenos v poloviční duplexu

Režim poloviční duplexu je možné používat maximálně do rychlosti 1 Gb/s. Protože zařízení na segmentu sdílejí komunikační kanál, používá se kolizní metoda CSMA/CD, která byla popsána výše, včetně Backoff algoritmu pro řešení kolizí.

Definice (Kolizní okno)

Kolizní okno (Collision Window, Slot Time) je doba, po kterou musí zařízení naslouchat na nosné, aby zachytilo vysílání jiného zařízení a dokázalo detekovat nebezpečí kolize.

Co se kolizního okna týče, platí tyto vztahy:

- největší možná vzdálenost mezi zařízeními v téže kolizní doméně (čím větší vzdálenost, tím déle musí zařízení naslouchat, protože signálu to „déle trvá“), čím větší je maximální povolená vzdálenost, tím větší musí být kolizní okno,
- minimální délka rámce (zařízení se musí dozvědět ještě během vysílání rámce, že došlo ke kolizi, aby mohlo vysílání zrušit, počkat dle Backoff algoritmu a vysílat znovu) – čím menší je minimální délka rámce, tím menší musí být kolizní okno a menší kolizní doména,
- pokud je malé kolizní okno, musí být malá i kolizní doména.

Z toho vyplývá, že existuje velmi úzký vztah mezi kolizním oknem, velikostí kolizní domény (max. vzdáleností mezi zařízeními v síti) a minimální délkou rámce.

Poznámka:

Představme si tuto situaci: v kolizní doméně jsou dvě zařízení – Z_a a Z_b. Signálu trvá dobu t, než se dostane od jednoho zařízení k druhému. Zařízení Z_a začne vysílat v době t_a rámec, jehož délka je taková, že vysílání rámce trvá po dobu d_a.

Pokud zařízení Z_b začne vysílat ještě před tím, než k němu dojde signál od zařízení Z_a, dojde ke kolizi. Teď je nutné, aby se zařízení Z_a včas dozvědělo, že ke kolizi došlo, aby mohlo vysílání zrušit a po čekání vysílat znovu. Musí se to tedy dozvědět ještě před tím, než skončí vysílání, tedy nejpozději v okamžiku $t_a + d_a$. Zařízení Z_b zjistí kolizi nejdřív v čase $t_a + t$, a pokud okamžitě zareaguje a pošle jam signál, tento signál se dostane k zařízení Z_a nejdřív v době $t_a + 2 \cdot t$. Takže tu máme nerovnici:

$$\begin{aligned} t_a + d_a &> t_a + 2 \cdot t \\ d_a &> 2 \cdot t \end{aligned}$$

To znamená, že minimální doba, po kterou má trvat vysílání jednoho rámce, musí být větší než doba, kterou potřebuje signál pro cestu tam a zpět mezi dvěma nejvzdálenějšími zařízeními na segmentu (v kolizní doméně). Proto pokud povolíme příliš velkou kolizní doménu (= chceme příliš dlouhé kabely), musíme přikázat vyšší minimální délku rámce.

Takže tady balancujeme mezi dvěma požadavky:

- Chceme sesíťovat co největší prostor, tedy jsou žádoucí dlouhé kabely a velká kolizní doména.
- Nechceme zbytečně zahlcovat linku v případě, že potřebujeme posílat i malé rámce (pokud je třeba poslat množství dat menší než je limit pro minimum, musí se přidat „vycpávka“ – data bez významu), takže je efektivnější mít spíše nízký limit pro minimální délku.

K tomu se přidává ještě třetí faktor – rychlost přenosu. Pokud pouze zvýšíme rychlost přenosu, musíme zvýšit limit pro minimální délku rámce nebo zmenšit doménu, protože zvýšení rychlosti ovlivní dobu přenosu.

Zaměřme se na různé rychlosti – 10 Mb/s, 100 Mb/s, 1000 Mb/s (u rychlejších se už nepoužívá CSMA/CD).

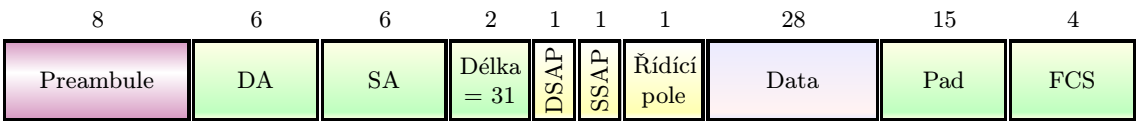
Pro 10 Mb/s je stanovena minimální délka rámce bez preambule na 512 bitů = 64 oktetů (podívejte se na strukturu rámce Ethernet II na straně 46), tedy odvysílání 512 bitů trvá 51,2 μs (to je doba d_a). Doba, po kterou může maximálně trvat cesta mezi dvěma nejvzdálenějšími zařízeními v kolizní doméně, je polovina této hodnoty. Vzdálenost v metrech pak záleží na tom, jaké přenosové médium je zvoleno (metalickými a optickými kabely se signál šíří odlišnou rychlostí).

Pokud je minimální délka MAC rámce bez preambule určena na 64 oktetů, pak by to, co je uvnitř zapouzdřeno, mělo být dlouhé minimálně $64-18 = 46$ oktetů (odečteme délku polí v záhlaví a zápatí MAC rámce).

Teoreticky může nastat situace, že by uvnitř rámce mělo být přeneseno méně než 46 oktetů dat. Standard IEEE 802.3 ve spolupráci s LLC a SNAP to řeší „vycpávkou“ (pad) přidanou za data, přičemž délka vycpávky se dá odvodit z hodnoty v poli *Délka*: pokud je v tomto poli číslo < 46, pak platí:

$$\begin{array}{rcrcrcl} \text{DSAP} & + & \text{SSAP} & + & \text{Řídicí pole} & + & \text{Data} & + & \text{Pad} & = & 46 \\ & & & & & & \textit{Délka} & + & \text{Pad} & = & 46 \\ & & & & & & & & \text{Pad} & = & 46 - \textit{Délka} \end{array}$$

Pro případ naznačený na obrázku 2.9 platí $\text{Pad} = 46 - (28 + 1 + 1 + 1) = 15$. Tento údaj je důležitý, aby bylo zřejmé, na kterém oktetu rámce začíná pole s kontrolním součtem. Pozor, pole *Délka* v sobě započítává i délku LLC záhlaví (na obrázku žlutě).



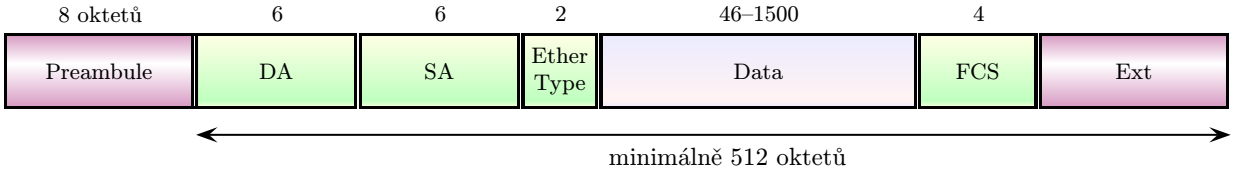
Obrázek 2.9: Velmi krátký LLC rámec v MAC rámci podle IEEE 802.3

Rámce podle Ethernet II toto neřeší, předpokládá se, že zapouzdřená data zachovávají určitou minimální délku. Protože je „uvnitř“ většinou IPv4 paket, jehož záhlaví je minimálně 20 oktetů dlouhé, na to, co je zapouzřeno uvnitř IP paketu, zbývá minimálně 26 oktetů, což není tak málo.

Při přechodu na vyšší rychlost, tedy na 100 Mb/s, bylo rozhodnuto zachovat minimální délku rámce (tj. pro formát rámce platí totéž, co bylo napsáno v předchozím odstavci) i velikost kolizního okna, tedy bylo třeba zmenšit velikost kolizní domény mezi DCE (přibližně 10×).

Na metalickém kabelu to znamenalo zmenšení kolizní domény z 2000 m na 200 m, což znamená, že mezi DTE a DCE je vzdálenost max. 100 m (to zůstalo) a mezi dvěma DCE max. 200 m.

Při přechodu na Gigabit Ethernet (1 Gb/s) toto řešení nebylo možné (20 m jako nejvyšší možná vzdálenost by bylo opravdu málo), tedy se pro změnu zvýšila minimální délka rámce bez započítání preambule na 512 oktetů (4096 bitů). Protože se často přenáší menší množství dat, je u rámců s délkou pod limitem nutné přidat další „vycpávku“ (Carrier Extension, nedatovou příponu), a to za kontrolní součet. Umístění je naznačeno na obrázku 2.10 (pole zcela vpravo).



Obrázek 2.10: Nedatová přípona u rámce pro Gigabit Ethernet

Nedatová přípona se při odesílání přidává až po vypočtení kontrolního součtu na MAC podvrstvě, při přijetí rámce je na MAC podvrstvě cílového systému odstraněna ještě před kontrolou podle kontrolního součtu. Skladba symbolů v nedatové příponě je patentovaná – musí být odlišitelná od pole s kontrolním součtem.

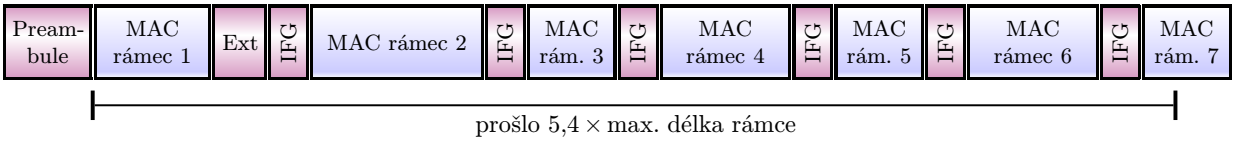
Údaje o patentu jsou dostupné na http://www.google.com/patents/US5940401.

Pokud by se posílalo velmi často větší množství rámců s malým množstvím zapouzdřených dat, linka by samozřejmě byla zbytečně vytěžována víc než je nutné. Takový případ se řeší jinak:

Burst mode (shlukový režim) je právě určen pro posílání sekvencí krátkých úseků dat. Je používán pouze u přenosů v rychlostech od 1 Gb/s výše, tj. od Gigabit Ethernetu. „Krátké“ rámce se posílají ve shluku, který má následující strukturu:

- první rámec ve shluku má výše popsanou strukturu, včetně případné nedatové přípony, pokud je menší než 512 oktetů,
- následuje *IFG* (InterFrame Gap) – speciální sekvence bitů vyplňující prostor mezi rámci, slouží k rozpoznatelnému oddělení rámců ve shluku,
- další rámce v posloupnosti navzájem oddělené IFG sekvencemi; pokud jsou kratší než 512 oktetů, není nutné je doplňovat o nedatovou příponu.

Celková délka shluku by neměla překročit přibližně 5,4násobek maximální možné délky rámce (přesněji 8 192 oktetů), ale pokud je tato hranice překročena až během odesílání posledního rámce ve shluku, může být jeho odesílání ještě dokončeno a shluk je ukončen až poté.



Obrázek 2.11: Přenos v burst (shlukovém) módu

Poznámka:

To znamená, že během přenosu v burst modu je třeba sledovat, zda se neblížíme k hranici 5,4násobku maximální délky rámce. Pokud jí dosáhneme, pak rámec, který právě odesíláme, je posledním rámcem shluku. Je to důležité, protože během odesílání shluku rámců vlastně blokujeme linku a znemožňujeme odesílat ostatním zařízením v segmentu (v téže kolizní doméně), tedy blokování nemůže trvat příliš dlouho.

Výpočty týkající se minimální délky rámce, velikosti kolizní domény apod. u zkoušky nebudou...

2.5.2 Přenos v plném duplexu

V plném duplexu je situace mnohem jednodušší. Na logické úrovni se vpodstatě setkáváme jen se spoji typu point-to-point a veškeré spoje jsou vlastně vyhrazené, zařízení může vysílat kdykoliv. Nepoužívá se přístupová metoda CSMA/CD a parametry spojení nejsou omezeny vlastnostmi kolizní domény a kolizního okna, pouze fyzikálními vlastnostmi přenosové cesty (například útlumem).

Vysílání může prakticky pořád probíhat v burst modu tak, jak bylo popsáno u poloviční duplexu. Vysílání a příjem probíhají po jiné cestě, ale může jít o jeden společný kabel (například v mnoha standardech pro fyzickou vrstvu je pro UTP se čtyřmi páry kroucené dvojlinky vždy jedna dvojice vyhrazena pro příjem a druhá dvojice pro vysílání, nebo se směr určuje adaptivně). U optických kabelů je buď pro každý směr jeden kabel, nebo pravděpodobněji v kabelu máme jedno či více vláken zvlášť pro různé směry.

Jediný problém (který může nastat i u poloviční duplexu, ale pro plný duplex je typický) nastává tehdy, když vysílající zařízení neustále vyvíjí aktivitu, ale přijímající zařízení nestíhá rámce přijmout. Přijetí rámce totiž není až tak triviální, kromě zpracování na fyzické vrstvě je třeba také postupně rámec rozbalit a patřičně ho zpracovat, a procesor (který se na tom taky podílí) může mít i jinou práci než zpracovávat rámce od dotyčného zařízení. Tento problém se řeší kombinací těchto způsobů:

- zařízení má dostatečně velký *buffer* (vyrovnávací paměť), tam si ukládá to, co zatím nemohlo být zpracováno,
- odesílatele je třeba zpomalit, sdělit mu, že má určitou dobu počkat s vysíláním.

První možnost je celkem jasná, ale nemusí být dostačující. Druhá možnost znamená použití tzv. *pause frame*, tedy speciálního rámce, který je poslán „pilnému“ odesílateli; tento rámec obsahuje především informaci o tom, na jak dlouho je třeba přerušit vysílání.

Poznámka:

Plný duplex se dá použít pouze na spojích typu point-to-point (v kolizní doméně jsou tedy pouze právě dvě fyzicky propojená zařízení), což například naprosto vylučuje použití hubu (rozbočovače). Navíc obě propojená zařízení „musí umět“ plný duplex, aby bylo možné ho využít.

Od 10G Ethernetu se již používá pouze plný duplex, poloviční není podporován. Ke kolizím z principu nemůže dojít, tedy nemá smysl využívat jakoukoliv přístupovou metodu (CSMA/CD se v tom případě nepoužívá).

2.6 Technologie související s Ethernetem

2.6.1 Autonegociace

Aby vůbec bylo možné zprovoznit přenos přes ethernetovou síť, je třeba, aby si síťová rozhraní komunikujících zařízení „rozuměla“, tedy aby se shodla na následujících parametrech:

- rychlost přenosu,
- pro danou rychlost konkrétní standard, pokud je víc možností pro použité přenosové médium (například aby bylo jasné, jak konkrétně budou jednotlivé páry UTP využívány),
- poloviční nebo plný duplex.

Zároveň je potřeba pravidelně provádět kontrolu funkčnosti spojení.

Pokud propojíme dvě zařízení kabelem a zprovozníme příslušná síťová rozhraní, pak se tato síťová rozhraní nejdřív dohodnou na výše uvedených parametrech – provede se *autonegociace* (tedy automatická dohoda o parametrech) a následně se pravidelně ozývají v době, kdy se nepřenášejí žádné rámce (aby bylo jasné, že dané zařízení je pořád připojeno a linka funguje). Autonegociace se provádí na fyzické vrstvě, je popsána přímo v standardu IEEE 802.3 a dodatcích.

První pokusy o autonegociaci (ale ještě ne zcela v plném smyslu toho slova) byly u Ethernetu 10Base-T, kdy síťové rozhraní vysílalo v pravidelných intervalech impulsy, které sloužily ke zjišťování, zda je linka funkční. Tento signál se označoval *NLP* (Normal Link Pulse). Pokud od zařízení po určitou dobu nepřišel žádný rámec ani NLP signál, linka byla považována za nefunkční.

Dnes, v době vyšších rychlostí, se odesílá signál *FLP* (Fast Link Pulse), kde se série původních NLP signálů prokládá dalšími informacemi (protože se postupně přidávaly další parametry, které bylo možné oznámit), a zatímco původně šlo o nepovinnou funkčnost, od Gigabit Ethernetu je pro metalické kabely autonegociace povinná.

Celý mechanismus je zpětně kompatibilní, a v reálu se pro linku použijí nejvyšší možné parametry z těch, které splňují obě komunikující strany. Například pokud jedno zařízení zvládá rychlost 100 Mb/s v poloviním duplexu a druhé rychlost 1 Gb/s s plným duplexem, použije se rychlost 100 Mb/s a poloviční duplex.

2.6.2 Napájení přes Ethernet

PoE (Power over Ethernet, IEEE 802.3af) je standard z roku 2003 popisující možnost napájení menších zařízení přes síťový (metalický) kabel. Výhodou je, že k dotyčnému zařízení nemusíme přivádět napájecí kabel. Takto napájené zařízení si samozřejmě musí vystačit s nižším odběrem, typicky jde například o IP telefony, IP kamery a některé Wi-fi access pointy připojené kroucenou dvojlinkou k zařízení, které může fungovat jako zdroj PoE (což je většinou switch).

Poznámka:

Ve skutečnosti se v praxi setkáváme s více různými řešeními – aktivní PoE je podle standardu IEEE 802.3af, a dále existuje několik nestandardizovaných starších implementací.

Switch sloužící jako zdroj PoE musí mít samozřejmě dostatečně dimenzovaný napájecí zdroj (což je ta komponenta, která stejně jako u běžných počítačů provádí transformaci střídavého napětí

z napájecí sítě na stejnosměrné napětí), a i přesto obvykle nedokáže napájet zařízení na všech svých portech (většinou i méně než polovinu), zbývající připojená zařízení musí mít vlastní napájení.

Rozlišujeme *výkonové třídy* 0–4 podle toho, jaký proud a maximální příkon napájené zařízení potřebuje, přičemž třída 0 je pro zařízení nepodporující PoE. Čím vyšší příkon, tím vyšší třída.

Podobně jako při stanovování parametrů připojení se v rámci autonegociace dojednává rychlost a duplex, u PoE se při připojení určuje třída. Nejdřív proběhne detekce, při níž se k připojenému zařízení pouští proud o nízkém napětí, které by v případě, že zařízení „neumí“ PoE, nemělo ublížit (odpovídá třídě 0). Jestliže zařízení zareaguje, pokračuje detekce určením výkonové třídy.

Pro účely napájení se v kabelu používají některé z vodičů. Jestliže se pro data využívají jen dva páry (u Fast Ethernetu), pak jsou zbývající dva páry použity právě pro napájení (tj. páry 4+5, 7+8). Pokud jsou ale pro data využívány všechny čtyři páry (Gigabit Ethernet), pak jsou páry vodičů využívány zároveň pro přenos dat i napájení.

Další informace:
http://vyvoj.hw.cz/produkty/ethernet/princip-cinnosti-power-over-ethernet.html

2.6.3 Strukturovaná kabeláž

Se strukturovanou kabeláží se v tomto předmětu seznamujeme především na cvičeních, zde se soustředíme jen na nejdůležitější pojmy.

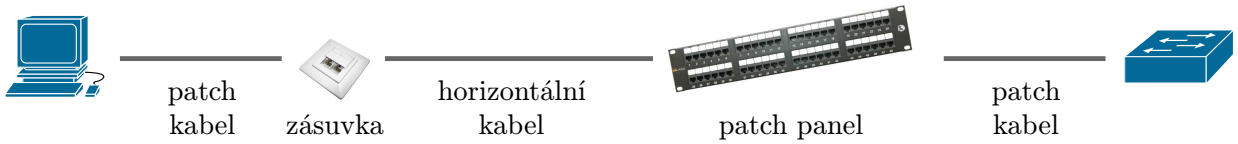
Rozvaděč (rack, rozvodná skříň) je speciální (otevřená nebo uzavřená) skříň poskytující zařízením (switchům, serverům apod.) upevnění, řízené chlazení, elektroinstalaci a konektivitu. Rozměry rozvaděčů jsou standardizované, resp. existuje několik běžných typizovaných rozměrů. Nejběžnější šířka je 19", hloubka 600 nebo 800 mm. Velikost zařízení do rozvaděče tomu samozřejmě odpovídá, přičemž výška se udává v násobcích jednotky 1 U = 1,75" = 4,4 cm.

Patch panel je pasivní síťový prvek, který obvykle umísťujeme mezi horizontální kabel a switch. Můžeme si ho představit jako předsunutou sadu portů pro „schovaný“ switch, přičemž patch panel taky obvykle připevňujeme do racku, typická výška je 1U.

Použití patch panelu nám zpřehledňuje a zjednodušuje přístup k portům switche (tedy pokud je propojení uděláno správně). Patch panel může být v „přístupnější“ výšce než switch, tedy se k němu snáze dostaneme, navíc do jednoho patch panelu mohou být zapojeny kabely z více switchů. Pokud je třeba změnit zapojení pro konkrétní horizontální kabel vedoucí z určité zásuvky, nemusíme provádět změnu na switchi, stačí přehodit kabel na patch panelu.

Horizontální rozvody jsou rozvody sítě vedoucí víceméně vodorovně (horizontálně), obvykle propojují zařízení v rámci jednoho patra budovy. Kabel pro horizontální rozvod je obvykle veden zdí mezi zásuvkou na jedné straně a switchem (přesněji patch panelem) na druhé straně. Co se týče délky jednotlivých částí horizontální trasy, platí:

- Celá fyzická přenosová cesta od hostitele ke switchi musí měřit maximálně 100 m.
- Fyzická délka horizontálního kabelu (od zásuvky k patch panelu) je maximálně 90 m.
- Délka patch kabelu mezi hostitelem a zásuvkou je maximálně 20 m.
- Délka ostatních patch kabelů (vč. mezi patch panelem a switchem) je maximálně 5 m.

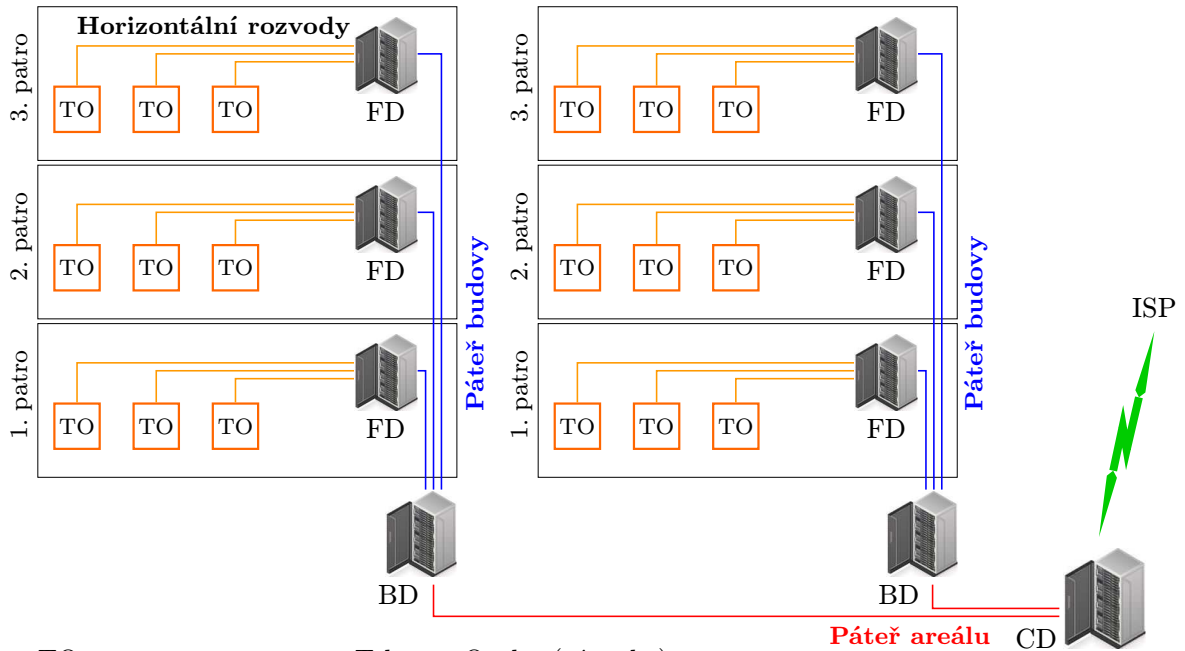


Obrázek 2.12: Horizontální kabeláž

Síťová (telekomunikační) místnost. Na každém patře budovy by měl být dobře zabezpečený prostor (ideálně vyhrazená místnost, pokud je to možné), do kterého jsou svedeny horizontální rozvody od všech zásuvek. Pokud je to samostatná místnost, nazýváme ji *telekomunikační místnost* nebo *síťová místnost* patra. Je v ní především rozvaděč, do kterého ústí horizontální rozvody z celého patra, tedy *floor distributor* (FD).

Páteřní síť budovy (building backbone) propojuje všechna patra budovy, resp. všechny FD. Středem páteře budovy je *building distributor* (BD) – rozvaděč pro celou budovu.

Páteřní síť areálu (campus backbone). Pojem *campus* (areál) označuje skupinu budov, které určitým způsobem patří sobě (například areál patřící jedné firmě. *Campus distributor* (CD) je rozvaděč zastřešující síť v celém areálu, jednotlivé BD jsou s CD propojeny *páteřní sítí areálu*. Právě přes CD jde veškerá komunikace s poskytovatelem internetu (ISP) pro danou firmu.



TO ..... Telecom Outlet (zásuvka)
FD ..... Floor Distributor (rozvaděč pro podlaží)
BD ... Building Distributor (rozvaděč pro budovu)
CD ..... Campus Distributor (rozvaděč pro areál)

Obrázek 2.13: Schéma rozvodů strukturované kabeláže

Kapitola 3

Další témata k lokálním sítím

Rychlý náhled: V této kapitole najdeme témata doplňující problematiku sítí Ethernet. Probereme princip VLAN (virtuálních lokálních sítí), podíváme se na řešení problému redundantních fyzických cest v síti switchů a protokol STP a pro orientaci si uvedeme stručný přehled standardů z rodiny IEEE 802.

Klíčová slova: VLAN, přístupový (access) port, trunk, nativní VLAN, management VLAN, IEEE 802.1Q, STP, IEEE 802.1D, konvergence, port předávání, kořenový port, blokovaný port, RSTP, MSTP.

Cíle studia: Po prostudování této kapitoly budete vědět, co je to VLAN, jak lze rozdělit síť na jiné než fyzické úrovni, jak zajistit komunikaci v rámci VLAN a mezi různými VLAN sítěmi. Také se naučíte, jak se řeší problém redundantních cest v síti switchů.

3.1 VLAN

VLAN (Virtuální LAN, Virtual LAN) je logické seskupení určitých zdrojů (například počítačů) nacházejících se v jedné nebo několika běžných lokálních sítích. Pokud v lokální síti definujeme VLANy, znamená to, že určujeme několik takovýchto seskupení a do každého přiřadíme konkrétní zdroje nacházející se v síti. Obvykle platí, že každý stroj patří právě do jedné VLANy a komunikace mezi různými VLAN je omezená. Jinými slovy – VLAN určuje, kdo s kým má právo volně komunikovat.

Proč něco takového dělat? Například z těchto důvodů:

- Zabezpečení – máme možnost omezit přístup ke konkrétnímu prostředku (třeba serveru s citlivými firemními informacemi) jen na někoho (členy jedné konkrétní VLANy).
- Omezení komunikace mezi VLAN sítěmi platí i pro broadcastové vysílání, takže máme nástroj fungující (většinou) na vrstvě L2, který dokáže zajistit oddělení všesměrových domén – to by jinak než pomocí VLAN nešlo.

Obojí by se dalo provést i jinak – zařazením routerů (směrovačů) do sítě. Jenže routery mají své nevýhody: jsou dražší a mnohem pomalejší než switche, a navíc router vlastně pracuje „mimo Ethernet“ – na vrstvě L3.

Poznámka:
Dále popisovaný algoritmus VLAN sice souvisí s Ethernetem, ale není povinný (ne každý Ethernetový switch ho implementuje) a je standardizován zvlášť.

Používáme tato termíny:

- *datové* VLAN – VLAN pro běžný provoz, do nich přiřazujeme zařízení (vlastně porty),
- *default* – pokud na portu není nastaveno nic jiného, je tam defaultní VLAN (u většiny výrobců je to VLAN číslo 1),
- *native* – pokud se mezi dvěma switchi přenáší něco, co nemá žádnou VLAN nastavenou, spadne to do „kanálu“ nativní VLAN (přes nativní VLAN bývají přenášeny například také STP rámce); jestliže jsme v konfiguraci žádnou nativní nenastavili ručně, VLAN 1 se použije jako nativní,
- *management* – je určená pro management rámce (SSH, Syslog, SNMP,...), výchozí nastavení je VLAN 1, ovšem rozhodně je dobré ji přenastavit.

Konkrétní port na switchi je

- *přístupový* – za ním je přímo jedno zařízení, má nastavenu příslušnost do jediné VLAN,
- *trunkový* – za ním je další switch (tedy potenciálně celá řada různých zařízení patřících do různých VLAN), sousední switch má příslušný port taky nastavený jako trunkový.

Přes přístupový port přenášíme pouze rámce patřící do na něm nastavené VLAN, přes trunkový port lze přenášet rámce patřící do více různých VLAN (nastavuje se seznam „povolených“ VLAN, seznam musí být stejný na obou koncích trunku).

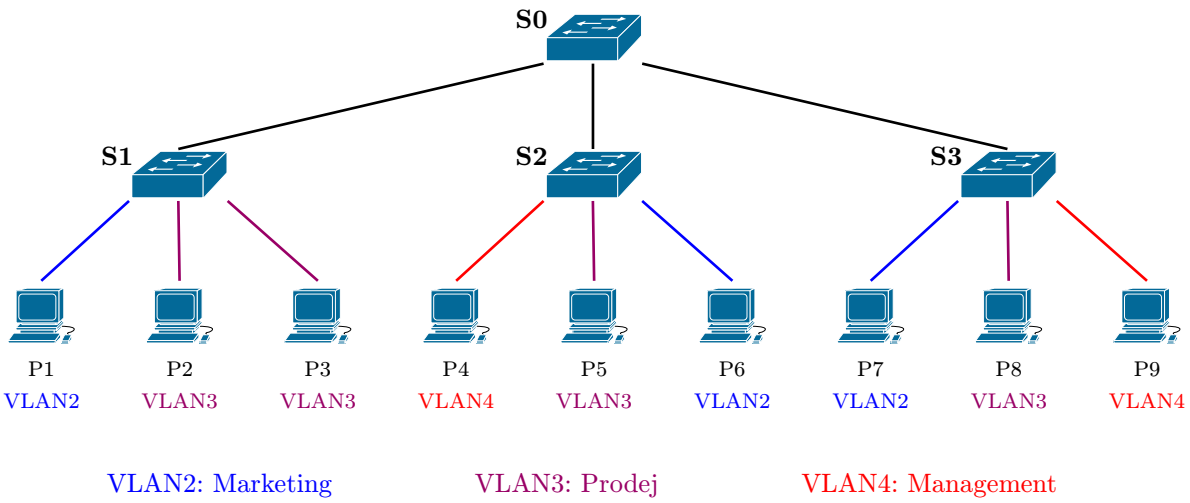
3.1.1 VLAN na jednom switchi

Princip VLAN je jednoduchý – rozdělíme zařízení v síti do skupin, každé skupině přiřadíme identifikátor (číslo VLAN) a zajistíme oddělení komunikace mezi těmito skupinami. Na obrázku 3.1 je naznačeno, jak by to mohlo vypadat (poněkud zjednodušeně).

Na switchi je pro každý port určeno, do které VLAN patří zařízení přes tento port přístupné. Tedy v našem případě je na switchi S1 první port přiřazen do VLAN2 a další dva porty do VLAN3. Port vedoucí směrem nahoru ke switchi S0 bude přenášet rámce z různých VLAN, proto ho necháme ve VLAN1.

Na každém switchi potřebujeme tabulku MAC adres. Protože přepínání má fungovat pouze mezi zařízeními patřícími do stejné VLAN (kromě nativní VLAN1), musí být ke každé MAC adrese a portu ještě přidána informace o číslu VLAN. Přepínání pak bude fungovat takto:

- Switch přijme na daném portu rámec, přičemž tento port patří do konkrétní VLAN.
- V tabulce MAC adres se zaměří pouze na záznamy patřící k téže VLAN, jiné nekontroluje.
- Mezi těmito „profiltrovanými“ záznamy najde ten, který odpovídá adrese cílového zařízení.
- V nalezeném záznamu najde port a přepne (odešle) rámec na tento port.



Obrázek 3.1: Ukázka využití VLAN

Zjednodušeně si to můžeme představit tak, že na switchi je pro každou VLAN samostatná tabulka MAC adres (ve skutečnosti tomu tak není, protože bychom pak neměli kam zařadit ta zařízení, která do běžných „komunikačních“ VLAN nepatří).

Pokud komunikace zůstává v rámci jednoho switche, je přepínání rámců celkem jednoduché. Pokud například počítač P2 posílá rámec počítači P3, přijme switch S1 tento rámec na „fialovém“ portu (VLAN3) a tedy se řídí podle řádků tabulky pro VLAN3, kde je přímo uvedeno, na kterém portu je cíl (počítač P3) dostupný. Takže v reálu to funguje stejně jako bez virtualizace.

Co se týče broadcastových rámců, jsou šířeny pouze v rámci té VLAN, ve které byly odeslány.

Poznámka:

Uvědomte si, že koncová zařízení naprosto netuší nic o nějaké virtualizaci. Když odesílají rámec, vědí jen to, na kterou MAC adresu má dojít, ale číslo VLAN neznají, toto číslo (zatím) není ani součástí odesílaného rámce. VLAN dokáže určit až switch, protože ten si ke každému portu eviduje informaci o VLAN, do které port (a k němu připojené zařízení) patří.

3.1.2 VLAN rámce na cestě přes trunk

Dále potřebujeme mechanismus, který nám umožní správně přepínat rámce, které mají jít od zdroje k cíli přes více než jeden switch. Dokud jsme zůstali v rámci jediného switche, bylo to jednoduché – switch určil VLAN podle portu, ze kterého rámec přišel, a tím byly stanoveny i konkrétní řádky tabulky, na kterých má hledat adresu a port cílového zařízení.

Pokud však je cíl připojen k jinému switchi, pak je třeba rámec nejdřív odeslat na tento switch, který pak již zajistí předání cíli. Jenže jak „tomu druhému“ switchi předat informaci o tom, do které VLAN patřil zdroj, aby následující switch věděl, na které řádky se podívat?

Zaměřme se opět na obrázek 3.1. Předpokládejme, že počítač P1 posílá rámec počítači P6. Nějak to jít musí, protože oba počítače patří do VLAN2.

Switch S1 přijme rámec od počítače P1, podle tabulky zjistí, že ho má předat switchi S0, ale musí *přidat informaci o číslu VLAN*. Switch S0 přijme rámec včetně této dodatečné informace

a obojí předá dál switchi S2. Až switch S2 zjistí, že tento rámec může předat přímo na port s přiřazeným číslem VLAN, a tedy dodatečnou informaci o číslu VLAN sice použije (aby určil tabulku adres), ale na cílový port odešle pouze rámec bez dodatečné informace.

Spoj, kterým posíláme rámce patřící do různých VLAN, se nazývá *trunk*. Vede obvykle mezi dvěma switchi nebo mezi switchem a routerem, výjimečně mezi switchem a serverem. Z toho vyplývá, že switche S1–S3 mají oba druhy portů – přístupové i trunkové.

Na obrázku 3.1 jsou spoje vedoucí k přístupovým portům zobrazeny barevně (podle barev VLAN), kdežto trunkové porty černě. Při komunikaci mezi dvěma počítači patřícími do téže VLAN je vždy prvním a posledním spojem na cestě spoj u přístupového portu, mezilehlé spoje jsou trunkové. Pouze přes trunkové spoje se přenáší dodatečná informace o VLAN.

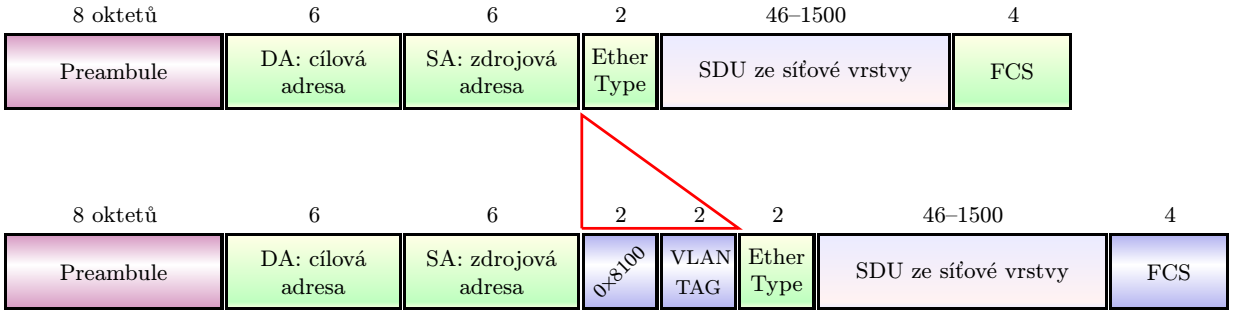
Nyní k té dodatečné informaci o VLAN. Už jsme si řekli, že při komunikaci v rámci jednoho switche (tj. pouze přes přístupové porty) něco takového není zapotřebí (dokonce platí, že pokud přijde přes přístupový port rámec s VLAN informací, bude zahozen, protože pravděpodobně se někdo pokouší „vpašovat“ do VLAN, do které nepatří), ale při komunikaci přes trunkový port je tato informace nutná, protože další switch na cestě nemůže vědět, ze kterého přístupového portu (a tedy VLAN) dotyčný rámec původně přišel.

Existují dva přístupy – první mění záhlaví rámce (přidá tam informaci o VLAN), kdežto podle druhého přístupu rámec neměníme a místo toho jej zabalíme do speciálního rámce s informací o VLAN.

První způsob je podle *standardu IEEE 802.1Q*. Podle tohoto standardu se informace o VLAN vkládá přímo do záhlaví přenášeného rámce, a to tak,

- aby bylo jasné, že to je VLAN rámec, a zároveň
- aby se ze záhlaví neztratily žádné původní položky.

Takže všechna původní pole necháme, ale dvě pole s informací o VLAN přidáme. Na obrázku 3.2 vidíme, kam jsou tato dvě pole vsunuta – hned za adresy.



Obrázek 3.2: VLAN rámec podle IEEE 802.1Q

Obsah nových polí je následující:

- EtherType hodnota pro VLAN (podle toho swich pozná, že doručený rámec obsahuje VLAN informaci). Pro VLAN to je hodnota 0×8100.
- VLAN tag, který v sobě zahrnuje
  - prioritu podle IEEE 802.1p (3 bity), obvykle je tady 0 (to znamená Best Effort – bez priorit, „nic nezaručuje, ale slibuje, že udělá maximum pro doručení“),

– indikace kanonického tvaru adresy (1 bit), obvykle 0,
– číslo VLAN (12 bitů).

Protože se tímto změnilo záhlaví rámce, je třeba znovu vypočíst kontrolní součet, a tedy se mění obsah posledního pole celého rámce (FCS).

Pokud switch přijme takový rámec, pak především určí, co s dotyčným rámcem má vlastně dělat. Takže pokud přijme switch (například switch S2 podle obrázku 3.1) na trunkovém portu rámec, zkontroluje pole EtherType. Pokud tam najde číslo 0×8100, je jasné, že se jedná o VLAN rámec a v následujícím poli bude identifikátor té VLAN, do které patří odesílatel.

Switch má přepínat rámce mezi porty (z jednoho rámec přijme, na druhý ho pošle) podle tabulky MAC adres, a totéž je třeba provést i zde. V reálu bude přepínání na různých switchích na cestě probíhat odlišně.

- Switch přijme rámec (bez VLAN ID) na přístupovém portu, podle portu určí VLAN. Pokud zjistí podle tabulky (vyfiltrováním řádků podle VLAN), že cílem nebude jiný přístupový port, přidá do záhlaví obě pole související s VLAN, vypočte kontrolní součet a odešle hotový VLAN rámec na trunkový port.
- Pokud switch přijme rámec na trunkovém portu a zjistí, že cíl je dostupný na přístupovém portu (určil VLAN podle záhlaví rámce), odstraní ze záhlaví obě VLAN pole, vypočte kontrolní součet a odešle na dotyčný přístupový port.
- Pokud máme switche v stromové hierarchii, pak nejvyšší úrovně obvykle pracují pouze v režimu nativní VLAN, tedy nekontrolují VLAN ID – aby páteřní síť zbytečně nezdržovala provoz. Takže takový switch přijme rámec, přečte adresu cíle a jednoduše přepne na příslušný port.

Poznámka:

Pojmy *přístupový port* a *trunkový port* patří do terminologie Cisco. U jiných výrobců se můžeme setkat s jinou terminologií (ale jinak to funguje úplně stejně, podle IEEE 802.1Q), například na zařízeních HP mluvíme o *untagged member* a *tagged member*.

K přístupovému portu (resp. untagged member) je připojeno zařízení, které „nerozumí“ VLAN (běžný počítač, server apod.), kdežto na trunkovém portu (resp. tagged member) je připojeno zařízení, které „rozumí“ VLAN (typicky switch) a umí pracovat s VLAN rámci.

Druhý přístup implementace VLAN rámců spočívá v tom, že se do původního rámce nezasahuje, ale je zapouzdřen do speciálního rámce podle určitého protokolu – přidá se nové záhlaví obsahující kromě jiného identifikaci VLAN. Na (některých) zařízeních Cisco je k tomuto účelu používán *protokol ISL*.

Kromě ISL existují i další protokoly používané k tomuto účelu (obvykle proprietární od určitého výrobce, navzájem nekompatibilní). Vznikly ještě v době, kdy IEEE 802.1Q neexistoval, a proto tehdy pro jejich používání existoval důvod.

V současné době se pro identifikaci VLAN na trunkových portech téměř výhradně používá IEEE 802.1Q, tedy se jinými protokoly (včetně ISL) ani nebudeme zabývat. Navíc pokud konkrétní zařízení podporuje některý z proprietárních VLAN protokolů, pak obvykle podporuje taky IEEE 802.1Q.

3.1.3 Komunikace mezi různými VLAN sítěmi

Jenže my sice chceme tyto podsítě navzájem oddělit, ale ne úplně. Určitou možnost komunikace bychom chtěli zachovat i mezi různými VLANami. A právě k tomu nám může sloužit router, který je na obrázku připojený ke kořenovému switchi. Obecně – komunikaci mezi různými VLANami může zajistit pouze a jenom zařízení pracující na vrstvě L3, takže jedno z nich:

- router,
- switch s funkcionalitou vrstvy L3.

Dále budeme pro zjednodušení zmiňovat a zobrazovat většinou router.

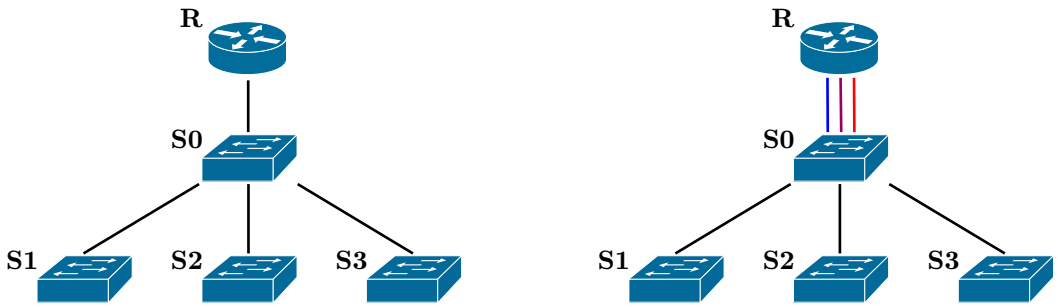
Při komunikaci mezi zařízeními z různých VLAN je rámec posílán přes zařízení vrstvy L3. Jestliže v tomto případě zašle počítač P1 z „modré“ VLAN2 rámec počítači P5 z „fialové“ VLAN3, bude cesta v síti následující:

$$\text{P1} — \text{S1} — \text{S0} — \text{R} — \text{S0} — \text{S2} — \text{P4}$$

přičemž na routeru proběhne určité testování a směrování a také se změní identifikátor VLAN v rámci (ve skutečnosti se na jakémkoliv L3 zařízení vždy odstraní původní záhlaví L2 a vytvoří nové). Proběhne směrování paketu vnořeného v rámci z podsítě určené pro VLAN zdroje do podsítě určené pro VLAN cíle, což lze jen na vrstvě L3.

Ovšem je třeba si uvědomit, že jakákoliv komunikace mezi zařízeními z různých VLAN opravdu půjde přes zařízení vrstvy L3, se všemi důsledky včetně prodloužení cesty rámce v síti a případně i snížení propustnosti spojů v síti (i kdyby zdroj a cíl byly připojeny k témuž switchi).

Různé spoje na L3 zařízení nutně musí být v různých podsítích (samozřejmě s nepřekrývajícím se adresním prostorem), tj. každý port by měl mít IP adresu ze „své“ vlastní podsítě. Ale není nutné, aby to platilo pro fyzické porty a fyzické spoje.



Obrázek 3.3: Zapojení zařízení vrstvy L3 a VLAN

Jsou dvě možnosti, jak napojení VLAN na L3 zajistit. V prvním případě povedeme mezi L2 zařízením a L3 zařízením jediný fyzický kabel, který je na logické úrovni rozdělen na virtuální spoje – pro každou VLAN vlastní virtuální spoj, v druhém případě povedeme mezi těmito zařízeními tolik kabelů, kolik je VLAN (tj. každá VLAN bude mít vlastní fyzický kabel). V prvním případě hovoříme o virtuálních subrozhraních jednoho fyzického síťového rozhraní, a místo abychom nastavovali IP adresu na síťovém rozhraní, budeme ji nastavovat zvlášť na jednotlivých subrozhraních. K oběma možnostem podrobněji:

Pokud router podporuje vytváření subrozhraní: centrální switch a router propojíme *z pohledu switche* trunkovým spojem, *na straně routeru* vytvoříme na daném síťovém rozhraní tolik

virtuálních subrozhraní, kolik VLAN bude moci projít přes trunk od switche. Každá VLAN bude mít své subrozhraní s vlastním označením a IP adresou. Subrozhraní jsou vytvářena postupem popsaným ve standardu IEEE 802.1Q. Situace je naznačena na obrázku 3.3 vlevo.

Konkrétně – na switchi tento port nakonfigurujeme jako trunkový, na routeru na druhé straně kabelu jde o síťové rozhraní s definovanými subrozhraními (například pro port `g0/1` to budou subrozhraní `g0/1.1`, `g0/1.2`,..., `g0/1.10`,... podle potřeby, za tečkou je číslo VLANy). IP adresu přiřazujeme jednotlivým subrozhraním, například subrozhraní `g0/1.10` bude mít IP adresu patřící do podsítě, kterou jsme určili pro VLAN 10.

Pokud zařízení vrstvy L3 nepodporuje vytvoření subrozhraní: Tento postup se týká jak routerů, které „neumí“ IEEE 802.1Q, tak i L3 switchů, které sice daný protokol zvládají, ale obvykle nepodporují vytvoření subrozhraní na jednom fyzickém rozhraní.

Poznámka:
Možná se to zdá jako paradox, ale L3 switche různých výrobců včetně Cisca opravdu obvykle neumožňují vytvořit na jednom rozhraní více subrozhraní, ale když se nad tím zamyslíte – switch (i s funkcionalitou L3) mívá desítky nebo dokonce stovky rozhraní, tak proč bychom si měli komplikovat život nějakými subrozhraními?

Mezi hlavním switchem a zařízením vrstvy L3 je třeba vést tolik fyzických spojů, kolik existuje VLAN. Tyto spoje pak budou na straně switche fungovat jako přístupové porty (tj. na každé je definována jedna konkrétní VLAN) a směrem k zařízení L3 (kde je síťové rozhraní s IP adresou patřící do podsítě pro příslušnou VLAN) půjdou již rámce bez VLAN polí.

Takže zatímco za trunkovým portem bývají obvykle L2 switche, za přístupovým (access) portem bývají buď koncová zařízení, nebo L3 zařízení bez definovaných subrozhraní.

Poznámka:
Na předchozích stránkách bylo uvedeno, že koncová zařízení neobsahují implementaci protokolu IEEE 802.1Q (tedy „nerozumějí VLAN“). Existují však i výjimky. U serverů může být výhodné zařazení do více než jedné VLAN (abychom se vyhnuli neustálému přeposílání provozu přes zařízení vrstvy L3), což se právě dá udělat implementací IEEE 802.1Q přímo na tomto zařízení. Jak se to dá provést:

- Pokud se jedná o server s Linuxem, pak je to jednoduché, protože Linux samotný obsahuje implementaci IEEE 802.1Q.
- U serverů s Windows je situace složitější, protože samotné Windows tento protokol nepodporují. Jediná možnost, jak tam rozjet podporu VLAN, je pořídit speciální síťovou kartu, která IEEE 802.1Q implementuje.

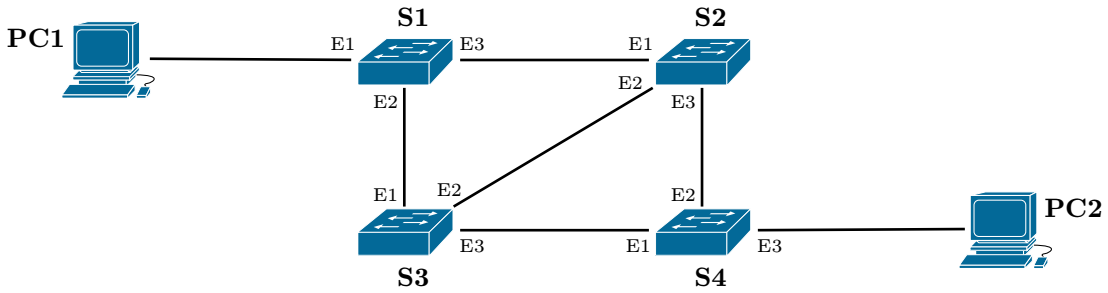
3.2 Switche a smyčky

V následujícím textu se zaměříme na postup, který je standardizován pro zařízení typu bridge, nicméně my budeme hovořit o switchích. Sice je určitý rozdíl mezi mosty a přepínači (přepínače

nejsou standardizované, mají víc portů a přepínání je realizováno hardwarově), ale princip je tentýž (pracují na vrstvě L2, vedou tabulku MAC adres a portů, přepínají provoz, oddělují segmenty).

3.2.1 Protokol STP

Představme si větší síť, ve které máme několik switchů. Protože chceme, aby síť pracovala i v případě, že některá přenosová cesta bude přerušena nebo některý switch přestane být funkční, jsou switche fyzicky propojeny více než je nutné – některé cesty jsou *redundantní* (znásobené).



Obrázek 3.4: Síť se smyčkami

Redundantní cesty jsou praktické pro případ výpadku části sítě, ale mohou přinést určité problémy. Představme si několik možných scénářů:

1. Počítač PC1 odesílá rámec počítači PC2. První část cesty (switch S1) je jednoznačná, ale co potom? Má být rámec odeslán ze switche S1 přes port E2 nebo přes port E3?
2. Počítač PC2 odesílá broadcastový rámec (určený pro všechna zařízení v síti). Switch S4 rámec přijme na portu E3 a odešle na porty E1 a E2. Co provedou ostatní switche?
   - Switch S2 rámec přijme na portu E3 a odešle na porty E1 a E2.
   - Switch S3 rámec přijme na portu E3 a odešle na porty E1 a E2. Tentýž rámec (o čemž neví) přijme i na portu E2 (od switche S2) a odešle na porty E1 a E3.
   - Switch S2 opět přijme rámec na portu E2 (od switche S3) a odešle na porty E1 a E3.
   - Switch S1 tentýž rámec přijme dvakrát z portu E2 a dvakrát z portu E3, odešle ho postupně celkem dvakrát na port E2, dvakrát na port E3 a čtyřikrát na port E1, čímž se rámec dostane k počítači PC1.
   - Mezitím se broadcastový rámec několikrát dostal zpět ke switchi S4, počítači PC2 a následně opět k dalším switchům...
3. Předpokládejme, že se počítač PC1 teprve připojuje do sítě, tedy zatím není v MAC tabulce žádného switche. Počítač PC2 mu odešle rámec (MAC adresa počítače PC2 je cílová). Switch S4 tuto adresu nezná, tedy rámec odešle na porty E1 a E2, atd. – rámec putuje sítí stejným způsobem jako broadcastový. K počítači PC1 se sice dostane, ale ještě dlouho budou jeho kopie bloudit sítí.

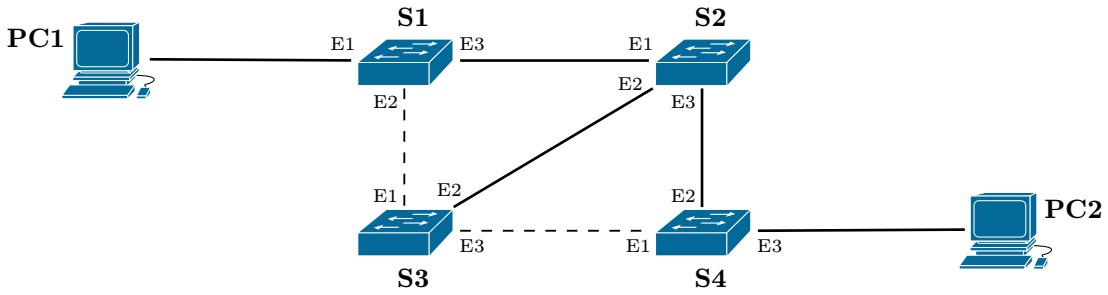
Poznámka:

Scénář popsaný v bodu 2 se nazývá *všesměrová bouře* (broadcast storm). Vyznačuje se tím, že sítí neustále bloudí totožné kopie téhož broadcastového rámce, čímž je síť defacto zahlcena a po určité době nepoužitelná.

Z toho vyplývá, že něco je špatně. Redundance cest je dobrá, ale zároveň bychom měli zabránit nekontrolovanému opakovanému bloudění rámců sítí. Tomu zabráníme, pokud ze sítě *odstraníme smyčky* (co se týče komunikace; ve fyzické topologii mohou smyčky kvůli redundanci zůstat).

Protokol STP (Spanning Tree Protocolol, také protokol kostry v grafu) je standardizován jako IEEE 802.1D jako mechanismus odstranění (logických) smyček v grafu sítě mostů (tedy v našem případě switchů). Tento protokol „vidí“ pouze mosty (resp. switche), žádná jiná zařízení ho nezajímají (takže v naší síti bude počítače PC1 a PC2 ignorovat).

Účelem je překonfigurovat switche v síti takovým způsobem, aby některé porty nebyly pro komunikaci používány (jsou buď deaktivovány nebo používány jen pro servisní účely), čímž ze sítě odstraníme smyčky. Protokol STP vpodstatě řeší „problém hledání kostry v grafu“, jehož účelem je v grafu nechat pouze takové spoje, aby mezi kterýmikoliv dvěma uzly sítě (switchi) existovala právě jedna cesta (ne více, ne méně), ideálně ta nejrychlejší.

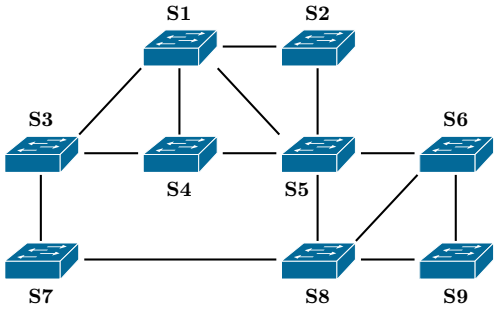


Obrázek 3.5: Síť po odstranění smyček

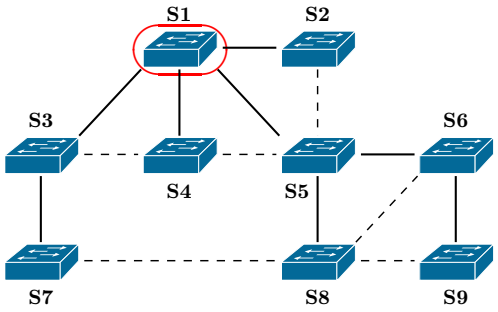
Vlastně ze sítě switchů vytvoříme *strom* („spanning tree“ znamená pokrývající strom). Jeden ze switchů je prohlášen za kořen stromu (*kořenový switch*) a z každého dalšího switche je hledána nejrychlejší cesta ke kořenovému switchi. Všechny ostatní cesty jsou deaktivovány. V našem případě by mohl být kořenovým switchem třeba switch S2 a v síti by zůstaly aktivní pouze spoje S1–S2, S3–S2 a S4–S2. Spoje S1–S3 a S3–S4 by byly deaktivovány.

Příklad

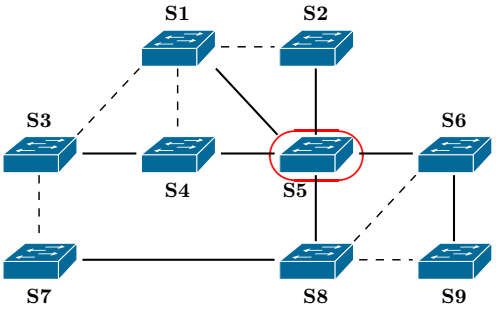
V síti na následujícím obrázku vidíme celkem devět switchů, které jsou propojeny ve fyzické topologii mesh. Máme tady smyčky, které je třeba vyřešit pomocí protokolu STP.

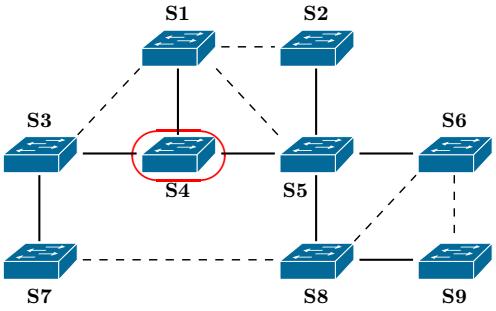


Prvním úkolem je zvolit kořenový switch. Máme na výběr, ovšem v reálné situaci bychom pro tuto volbu zřejmě měli určitá kritéria. Zde jednoduše zvolíme switch S1. Některé porty blokujeme, čímž pro běžnou komunikaci znepřístupníme celkem šest cest.



Jsou i další možnosti – jako kořenový switch můžeme zvolit třeba switch S5 nebo S4:





Také v těchto případech se některé porty deaktivovaly pro běžný provoz, přičemž od každého switche existuje právě jedna (pokud možno nejoptimálnější) cesta ke kořenovému switchi.

3.2.2 Konvergence sítě switchů

Podle čeho se určuje kořenový switch? Každý switch, který „rozumí“ protokolu STP (je na něm tento protokol implementován), má přiřezen speciální identifikátor. Tento identifikátor o délce 8 oktetů se skládá ze dvou částí:

- Priorita (2 oktety)
- MAC adresa (6 oktetů).

1	2	3	4	5	6	7	8
Priorita		MAC adresa					

Standardně je v poli pro prioritu číslo 0×8000, což je desítkově 32 768. Z toho vyplývá, že žádné dva switche nemají tento identifikátor stejný (liší se alespoň v MAC adrese, každý switch má jinou).

Protokol STP vybere jako kořenový strom ten switch, jehož identifikátor je *nejmenší číslo*. Pokud bychom spoléhali jen na tento postup, mohl by být jako kořenový zvolen ten switch, u kterého se nám to moc nehodí (vybíral by podle MAC adres switchů) – řešením je právě změnit prioritu námi vybraného switche na *menší hodnotu*. Díky tomu, že v identifikátoru je nejdřív priorita a pak až MAC adresa, má tato změna vždy žádaný účinek.

Zbývá stanovit, jak určíme, které spoje (resp. porty) mají být deaktivovány a které mají zůstat. Pokud ke kořenovému mostu vede cesta přes víc než jeden port, pak se pro každý port vypočte optimalita této cesty (záleží na šířce pásma/rychlosti, vytížení, hovoříme o ceně cesty, resp. nákladech na port), případně se sečtou části priority všech switchů na cestě ke kořenovému mostu a vybere se cesta s nejnižším součtem. Aktivní bude jen port pro nejoptimálnější cestu.

Takže port buď je nebo není aktivní z pohledu přeposílání rámců s běžným provozem. Tuto vlastnost označujeme jako *stav*, rozlišujeme tyto stavy portu:

- *předávání* (forwarding) – funguje, přeposílá veškerý provoz,
- *blocking* – nepřeposílá běžný provoz, ale přijímá správní rámce protokolu STP (tzv. BPDU rámce), aby se v případě potřeby dokázal aktivovat,
- *naslouchání* (listening) a *zjišťování* (learning) – mezilehlé stavy pro případ, že port má přejít ze stavu blokování do stavu předávání; ve stavu listening switch zpracovává BPDU rámce, ve stavu learning přijímá (ale nepřeposílá, zahazuje) rámce s běžným provozem a začíná si tvořit tabulku MAC adres,
- *disabled* – zcela vypnutý port, nepřeposílá vůbec nic.

Porty se mezi těmito stavy přesouvají, třebaže většinu času stráví právě ve stavech předávání nebo blokování.

Čas od času (především při změnách v síti) probíhá rekonfigurace – *proces konvergence*. Účelem je zajistit, aby byla síť v konvergentním stavu (tj. v našem případě aby v síti nebyly žádné smyčky a aby byly aktivní pouze porty pro nejoptimálnější cestu). Může být nutné určit nový kořenový switch, a také stavy portů se mohou měnit (například pokud má port přejít ze stavu blokování do stavu předávání, stráví určitý čas ve stavech naslouchání a zjišťování, aby si dokázal nastavit správně všechny parametry a doplnit tabulku MAC adres).

Každý port plní určitou roli, která je dána pozicí switche v hierarchii, taky se rozlišuje, jestli port míří směrem nahoru ke kořenovému switchi nebo dolů k podřízeným switchům. Předpokládejme, že máme switche nakresleny tak, že root je nahoře. Podle role se v dané chvíli určuje, v jakém stavu tento port bude. Jaké role tedy existují?

- *root port* – port, který směřuje ke kořenovému switchi, tedy každý switch kromě kořenového má jeden root port mířící „nahoru“, root port je vždy aktivní, tj. ve stavu forwarding,
- *designated port* – port, který je aktivní a v hierarchii míří dolů, tedy k podřízeným switchům (takže za designated portem máme podstrom switchů); kořenový switch má všechny své porty kromě blokovaných v roli designated, naopak switche v hierarchii zcela dole (takové, od kterých je ke kořenovému switchi nejdelší cesta) nemají žádné designated porty,
- *blocked port* – tento port nebyl vybrán, nepoužívá se pro běžný provoz (tedy obvykle je ve stavu blocking).

Aby to vše mohlo fungovat, switche se navzájem domlouvají pomocí speciálních rámců, které označujeme BPDU (Bridge PDU). Slouží k údržbě a případné aktualizaci informací o síti potřebných pro STP, posílají se na multicast adresu označující všechna zařízení implementující STP (tj. adresu `01:80:C2:00:00:00`). BPDU se odesílají pravidelně každé 2 sekundy, a pak při každé změně sítě switchů.

Součástí BPDU je kromě jiného informace o tom, že se jedná o rámec STP a jaké verze, který switch je kořenový (jeho bridge ID/switch ID), dále ID odesílajícího switche, údaje o době platnosti rámce, cena cesty od odesílatele ke kořenovému switchi, atd. Tyto údaje jsme viděli v příkladu na výpisech. BPDU se zapouzdřuje do LLC rámce podle IEEE 802.2 a následně do MAC rámce IEEE 802.3.

Další informace:

https://wiki.wireshark.org/STP

Pokud se do sítě switchů začleňuje nový switch, musí se také zapojit do STP stromu. Nový switch nejdřív sám sebe považuje za kořenový switch, ale své porty zatím nechává ve stavu naslouchání (přijímá BPDU, ale nic neodesílá). Pokud v přijatém BPDU najde ID kořenového switche, které je nižší než jeho ID, přestane se považovat za kořenový switch a postupně se zařadí do struktury (pro každý port si z přijatých BPDU vypočte cenu cesty k ostatním switchům a pro každý switch si vybere jen jeden port, ostatní blokuje).

3.2.3 Protokol STP a jeho varianty

Původní protokol STP byl standardizován jako IEEE 802.1D.

RSTP. Protože u rychlejších standardů přestala původní specifikace stačit, byl vytvořen nový standard RSTP (Rapid STP) jako IEEE 802.1w, nicméně nyní je zahrnut do revize IEEE 802.1D-2004 (takže původní STP a novější RSTP splynuly v jednom standardu).

Zatímco u původního STP trval proces konvergence 50 sekund (po tuto dobu byla celá síť nefunkční, což je strašně dlouho), u novějšího RSTP to trvá kolem 1 sekundy, i méně. Mezi původním STP a RSTP jsou sice i jiné rozdíly, ale tento rozdíl je nejvýraznější.

MSTP. Varianta MSTP (Multiple STP) byla standardizována jako IEEE 802.1s, také se označuje jako MST (Multiple Spanning Tree). Účelem je zefektivnit činnost protokolu STP pro případ, že jsou v síti definovány virtuální sítě (VLAN). Taktéž byla dodatečně přidána do standardu, ale tentokrát IEEE 802.1Q, a to jako revize IEEE 802.1Q-2005 jako rozšíření RSTP pro VLAN sítě.

V principu jde o to, aby pro každou VLAN nebo skupinu VLAN existovala samostatná instance protokolu STP, tedy pro každou VLAN (či několik) by mohla být trochu jiná struktura aktivních linek. Účelem je jak zefektivnění komunikace v rámci jednotlivých VLAN (aby například zbytečně nebyla vedena komunikace přes takové switche, které s dotyčnou VLAN nemají co do činění), ale také možnost vyrovnání zátěže v síti (určitý port bude pro jednu VLAN blokován a pro jinou aktivní, u jiného portu to může být naopak, každá bude využívat jiné cesty).

Další informace:

http://www.samuraj-cz.com/clanek/cisco-ios-10-rapid-spanning-tree-protocol/

3.3 Přehled rodiny standardů IEEE 802

Z předchozího textu vyplynulo, že Ethernet je standardizován jako IEEE 802.3 (ve skutečnosti se pojem „Ethernet“ v dotyčném standardu vůbec nevyskytuje, ale neformálně se tomu tak říká), a že existují standardy IEEE 802.2 (pro podvrstu LLC linkové vrstvy) a standard IEEE 802.1 (s jeho některými částmi jsme se setkali v této kapitole). Ale co dál?

V tabulce 3.1 je přehled momentálně existujících standardů z rodiny IEEE 802. Každý standard je vypracováván konkrétní pracovní skupinou (working group), takže například standard IEEE 802.3 má na starosti *Ethernet Working Group* a standard IEEE 802.11 má „na svědomí“ *Wireless LAN Working Group*.

Standard	Význam
802.1	Technologie mostů a správa sítí na vrstvě L2
802.2	Logické řízení spoje na vrstvě L2 – podvrstva LLC vrstvy L2 (pracovní skupina neaktivní)
802.3	Lokální síť Ethernet, přístupová metoda CSMA/CD (vrstva L1 a podvrstva MAC vrstvy L2)
802.4	Lokální síť Token Bus – sběrnicová LAN (pracovní skupina rozpuštěna)
802.5	Lokální síť Token Ring – kruhová LAN (pracovní skupina neaktivní)
802.6	Metropolitní síť DQDB (pracovní skupina rozpuštěna)
802.7	Širokopásmové (broadband) sítě LAN (pracovní skupina rozpuštěna)
802.8	Sítě LAN/MAN na optických vláknech (pracovní skupina rozpuštěna)
802.9	Integrované služby – izochronní sítě (pracovní skupina rozpuštěna)
802.10	Bezpečnost v sítích LAN/MAN (pracovní skupina rozpuštěna)
802.11	WLAN Working Group: Bezdrátová lokální síť – Wi-fi
802.12	Vysokorychlostní sítě – 100VG-AnyLAN (pracovní skupina rozpuštěna)
802.14	Širokopásmová síť na kabelech pro kabelové televize (pracovní skupina rozpuštěna)
802.15	WPAN Working Group: bezdrátové osobní sítě (například Bluetooth, ZigBee, UWB), včetně koexistence s IEEE 802.11
802.16	Širokopásmové bezdrátové MAN sítě (WiMAX) – neaktivní
802.17	RPR – Resilient Packet Ring, používá se v sítích SONET/SDH, neaktivní
802.18	Regulace rádiových vln
802.19	Koexistence nelicencovaných bezdrátových sítí
802.20	Mobilní širokopásmové bezdrátové LAN/MAN sítě včetně dopravní mobility
802.21	Handover Services Working Group – předávání mezi různými typy mobilních a bezdrátových sítí (GSM, GPRS, Wi-fi, Bluetooth, WiMAX, atd.)
802.22	Bezdrátové regionální sítě využívající nepoužívané frekvence televizního vysílání
802.23	Emergency Services Working Group (neaktivní)
802.24	Smart Grid Technical Advisory Group: vertikální aplikace – Internet of Things (IoT), komunikace Machine-to-Machine, Smart Grid, propojení dopravních prostředků...

Tabulka 3.1: Rodina standardů IEEE 802

Nejdůležitější standardy (tedy z našeho pohledu) jsou zvýrazněny tučným písmem. Některé z nich už známe, s jinými se seznámíme později.

Poznámka:

Mnohé z pracovních skupin IEEE 802 jsou buď neaktivní nebo dokonce rozpuštěné. Neaktivní znamená, že dlouho nebyla vydána žádná aktualizace či dodatek (což není problém, když dodatky nejsou nutné), rozpuštěná znamená, že se ani v budoucnu neplánují žádné aktivity (typicky u technologií, které se v praxi neprosadily).

Například sítě Token Ring, Token Bus a 100VG-AnyLAN byly postupně vytlačeny Ethernetem,

Izochronní sítě nabízející možnost přidělovat prioritu některým typům dat (třeba při přenosu hlasu) taktéž (Ethernet s využitím IEEE 802.1p/Q to umí taky a je rychlejší).

Standard IEEE 802.1 je asi nejrůznorodější. Zatím jsme se setkali s těmito jeho podřízenými standardy a dodatky:

- IEEE 802.1Q – VLAN (Virtuální lokální sítě),
- IEEE 802.1p – stanovení priority (hodnota využívající 3 bity, tedy v rozsahu 0–7) používané například standardem IEEE 802.1Q, ve skutečnosti to není standard, pouze jednoduchý předpis,
- IEEE 802.1D – protokol STP (technologie mostů a síť bez smyček),
- IEEE 802.1w – původně protokol RSTP (opět není chápáno jako samostatný standard, je to rozšíření standardu), později je specifikace přidána do revize IEEE 802.1D-2004,
- IEEE 802.1s – původně MSTP, později je specifikace přidána do revize IEEE 802.1Q-2005.

Z těch, kterým jsme se dosud nevěnovali, je zajímavý například IEEE 802.1X – autentizace zařízení přistupujícího do LAN. Autentizace podle tohoto standardu se často používá ve firemních sítích pro ověřování přístupu uživatele/zařízení do sítě a určování konkrétních oprávnění pro daného uživatele.

Poznámka:

Všimněte si, že zcela chybí řádek IEEE 802.13. Oficiálně je číslo 13 vyhrazeno pro další vývoj fast Ethernetu, ale ve skutečnosti je důvod velmi podobný důvodu toho, proč v mnoha hotelech chybí třinácté patro.

Kapitola 4

Síťová a transportní vrstva

Rychlý náhled: V této kapitole se budeme zabývat tím, co se obvykle děje na síťové a transportní vrstvě. Zatím nás budou zajímat pouze adresy, tvar paketů/segmentů a nejdůležitější procesy, které jsou na těchto vrstvách zajišťovány. Zaměříme se na protokoly IPv4, IPv6, ICMP na síťové vrstvě, a TCP, UDP na transportní vrstvě.

Klíčová slova: IPv4, IPv6, paket, TTL, MTU, fragmentace, ICMP, klient-server komunikace, TCP, UDP, port, spojení

Cíle studia: Po prostudování této kapitoly budete ovládat adresy IPv4 a IPv6, budete vědět, jak vypadá IPv4 a IPv6 paket, dokážete popsat funkci pole TTL a jeho spojitost s životností paketu, dále porozumíte důvodu a procesu fragmentace paketu. Dále porozumíte průběhu TCP spojení, budete vědět, jak vypadá TCP a UDP segment a dokážete popsat rozdíl mezi nimi.

4.1 Síťová vrstva a logické adresy

Síťová vrstva (v terminologii podle RM ISO/OSI, vrstva L3), resp. internetová vrstva (podle síťového modelu TCP/IP), pracuje s logickou topologií sítě. Úkolem vrstvy L3 je zajištění jednotného síťového rozhraní pro nadřízené vrstvy (vyšší vrstvy se již nezabývají samotným procesem komunikace s konkrétními zařízeními), a taky směrování, kterému se budeme věnovat v jiné kapitole.

Zatímco protokoly podřízené vrstvy (L2) komunikují v rámci jedné sítě, protokoly síťové vrstvy zajišťují komunikaci i za hranice jedné sítě, tedy zařízení této vrstvy dokážou propojovat různé sítě (dokonce i takové, které na nižších vrstvách používají navzájem odlišné protokoly). Typickými zařízeními síťové vrstvy jsou routery a switche s funkcionalitou vrstvy L3.

O adresách síťové vrstvy hovoříme jako o *logických (softwarových) adresách*, na rozdíl od fyzických (hardwarových) adres linkové vrstvy. Jejich vztah si můžeme představit podobně jako vztah mezi dvěma údaji v „lidské“ adrese: název města se směrovacím číslem je obdobou logické adresy (IP adresa), kdežto název ulice s číslem domu je obdobou fyzické adresy (MAC adresa). Pošťák rozvážející poštu autem mezi městy má přehled o tom, kterým směrem se nejlépe dostat k jednotlivým městům, ale na druhou stranu by uvnitř konkrétního města byl víceméně ztracen

a nedokázal by doručit poštu „až do domu“, kdežto místnímu pošťákovi jsou jiná města než to jeho vlastní naprosto lhostejná a naopak doručení pošty „až do domu“ uvnitř jeho města mu naprosto nedělá problém.

Takže tím máme rozdělené kompetence – vrstva L2 s fyzickými (hardwarovými, MAC) adresami se zabývá adresováním a doručováním (zde přepínáním) v rámci místní sítě, kdežto vrstva L3 s logickými (softwarovými, IP) adresami se zabývá adresováním a doručováním (zde směrováním) mezi sítěmi.

Na síťové vrstvě je nejdůležitějším protokolem protokol IP. V současné době se setkáváme s protokolem IP ve dvou verzích – 4 a 6. Protože zastoupení v síťovém provozu je velké u obou, budeme se zabývat oběma verzemi.

Z dalších protokolů nás bude zajímat především protokol ICMP určený k posílání krátkých servisních zpráv mezi zařízeními a protokol ARP s jeho následníkem NDP pro evidování vztahu mezi IP a MAC adresou nejbližších sousedů v síti. Dále zde pracují směrovací protokoly.

4.2 Protokol IPv4

Hlavním úkolem protokolu IP (Internet Protocol) je přijímat z nadřízené vrstvy segmenty s daty, zapouzdřovat do IP paketu (tj. přidat IP záhlaví) a předat podřízené vrstvě, a naopak.

4.2.1 Adresy IPv4

IPv4 adresa je dlouhá 32 bitů, tedy 4 oktety. V zápisu se jednotlivé oktety oddělují tečkou a zapisují se buď v desítkové soustavě nebo binárně.

Příklad

IPv4 adresa může vypadat třeba takto:

- `10.6.29.181`
- `169.251.220.5`
- `169.251.255.255`
- `255.255.255.255`

Všechny tyto adresy jsou zapsány v desítkové soustavě. První uvedená adresa by v binárním tvaru byla: `1010.110.11101.10110101`. Poslední uvedená: `11111111.11111111.11111111.11111111`.

IPv4 adresou může být označeno jak konkrétní zařízení, tak i například síť nebo podsíť (to vše jsou unicast adresy), dále existují IP adresy skupinové (multicast) a všesměrové (broadcast) v tom smyslu, jaký známe z předchozích kapitol.

IP adresy jsou *hierarchické* – to znamená, že zohledňují určité členění zařízení s těmito adresami do skupin a podskupin (sítí a podsítí), přičemž „příbuznost“ dvou zařízení v rámci sítě či podsítě (tedy příslušnost do stejné sítě či podsítě) znamená, že tato dvě zařízení budou mít část adresy stejnou. Takže hierarchie se na adrese projevuje následovně:

- *síťová část adresy (prefix)* – všechna zařízení patřící do stejné sítě mají tuto část shodnou,
- *část adresy hostitele* – v zbývající části adresy se tato zařízení budou lišit.

Rozhraní mezi síťovou a hostitelskou částí může být různé, celkových 32 bitů může být rozděleno například na 8+24, 16+16, 18+14, atd. podle pravidel, kterými se budeme zabývat v jedné z dalších

kapitol. Navíc může být tato hierarchie i složitější než jen se dvěma vrstvami, protože síť může být dále členěna na podsítě (přičemž zařízení v téže podsíti mají stejnou adresu sítě a podsítě, lišit se budou jen v části hostitelské).

Příklad

Pokud má zařízení IP adresu `10.6.29.181`, může být stanovena hranice mezi síťovou a hostitelskou částí na rozmezí prvního a druhého oktetu, tj. první oktet (`10`) by byl adresou sítě (tj. adresa všech zařízení v této síti by začínala oktetem `10`) a zbývající tři by jednoznačně určovaly adresu dotyčného zařízení v této síti. Jiné zařízení z téže sítě by mohlo mít třeba adresu `10.6.132.1`.

Pokud v dané adrese nastavíme všechny bity v hostitelské části na 0, získáme *adresu sítě*. Pokud naopak nastavíme všechny bity v hostitelské části na 1, získáme *broadcast adresu pro danou síť*.

Jak bylo napsáno výše, někde uvnitř adresy je hranice mezi síťovou a hostitelskou částí adresy. Ale když ta hranice může být na různých místech, jak tedy to konkrétní místo poznat? Pro určení této hranice se používá jedna z těchto dvou metod:

1. *Maska sítě nebo podsítě* – vypadá jako samotná adresa, ale v binárním zápisu jsou v (pod)síťové části adresy jedničky a v hostitelské části nuly.
2. *Zápis délkou prefixu* – za adresu napíšeme lomítko a číslo určující počet bitů síťové části.

Příklad

Navážeme na předchozí příklad. Vezměme opět adresu `10.6.29.181`, což je binárně po doplnění nul zleva `00001010.00000110.00011101.10110101`. Chceme dát na vědomí, že síťová část adresy je tvořena prvními 14 bity, pak:

1. přidáme masku sítě – v tomto případě bude:
   - binárně: `11111111.11111100.00000000.00000000`
   - dekadicky: `255.252.0.0`
2. napíšeme adresu ve tvaru `10.6.29.181/14`, kdy 14 je délka prefixu (tj. délka síťové části adresy).

IP adresou lze identifikovat nejen konkrétní zařízení v síti, ale i v souhrnu celou síť (protože jde o hierarchický adresový systém). Adresa sítě pak v našem případě bude

- binárně: `00001010.00000100.00000000.00000000`
- dekadicky: `10.4.0.0`

(nechali jsme původní prefix, tj. bity, které jsou v masce nastaveny na 1, ale všechny ostatní bity jsme nastavili na 0). Broadcast adresa pro síť bude

- binárně: `00001010.00000111.11111111.11111111`
- dekadicky: `10.7.255.255`

To znamená, že pokud některé zařízení pošle broadcastový IP paket určený všem zařízením v této síti (tj. síti určené adresou `10.4.0.0/14`), použije jako cílovou právě tuto adresu.

Na vrstvě L3 se provádí směrování mezi sítěmi, tedy musí být možnost propojení různých sítí. K tomuto propojení potřebujeme zařízení, přes které půjde veškerá komunikace mezi propojenými sítěmi – router nebo switch s funkcionalitou L3. V terminologii IP se tomuto zařízení říká *brána*, a z pohledu jakéhokoliv zařízení v síti je brána takový aktivní síťový prvek, na který směrujeme pakety určené pro zařízení v jiné síti – tj. brána je zařízení pro „cestu ven ze sítě".

Abychom vůbec byli schopni komunikovat s někým mimo vlastní síť, potřebujeme znát adresu brány. Takže z pohledu vlastní identity a možnosti komunikace na vrstvě L3 by každé zařízení mělo obdržet tyto informace:

- IP adresa,
- maska (pod)sítě nebo délka prefixu,
- adresa brány.

4.2.2 Speciální adresy podle IPv4

Již dřív jsme si říkali, že podle množství cílových zařízení rozlišujeme komunikaci typu unicast, multicast a broadcast. Abychom přímo podle adresy v paketu poznali, zda je cílem jedno konkrétní zařízení, skupina zařízení nebo všechna zařízení v síti, musí mít tato adresa určitý tvar.

Broadcastová (všesměrová) adresa pro danou síť je taková adresa, kde jsou všechny bity v hostitelské části nastaveny na 1. Cílem jsou všechna zařízení v dané síti. *Univerzální broadcastová adresa* má všechny bity jak v hostitelské, tak i v síťové části nastaveny na 1, tj. je to adresa `255.255.255.255`. Cílem jsou všechna zařízení v propojených sítích.

Multicast (skupinová) adresa se používá jako cílová v těch paketech, které mají být doručeny více než jednomu cílovému zařízení, ale ne nutně všem v síti. Pro tento typ adres je vyhrazen rozsah `224.0.0.0` až `239.255.255.255`
(možná to není vidět, ale jsou to všechny adresy takové, kde jsou první tři bity nastaveny na 1 a čtvrtý na 0). Některé skupinové adresy jsou vyhrazeny pro konkrétní účely:

- `224.0.0.1` označuje skupinu všech zařízení v lokální síti, která „rozumí" protokolu IPv4, tedy je to obdoba broadcast adresy v lokální síti,
- `224.0.0.2` je skupina všech routerů v lokální síti (takže když chceme poslat paket routerům, pošleme ho právě na tuto adresu),
- `224.0.1.1` je skupina pro NTP servery (časové servery) sloužící k synchronizaci času v síti, atd.

Adresy ve tvaru `224.0.0.x` (tedy včetně prvních dvou uvedených) jsou určeny pouze pro použití v rámci lokální sítě. Mohou být použity jako cílové v paketech pouze s nastaveným TTL = 1, to znamená, že se nedostanou přes router do jiné sítě.

Další informace:

Seznam registrovaných multicast IPv4 adres je dostupný na
http://www.iana.org/assignments/multicast-addresses/multicast-addresses.xhtml

Loopback adresa (adresa zpětné smyčky) je adresa začínající oktetem `127`, v naprosté většině případů je to adresa `127.0.0.1`. Loopback adresu má každé zařízení.

Loopback je uvnitř zařízení implementován jako virtuální zařízení, ze síťového pohledu je to vlastně „pohled na sebe sama“ ve směru ze sítě, jakési zrcadlo. Takže když chceme otestovat, zda je naše síťové rozhraní funkční (bez ohledu na to, zda má či nemá přiřazenu IPv4 adresu), otestujeme právě loopback.

Rozlišujeme *veřejné* a *soukromé adresy*. Veřejné adresy jsou celosvětově unikátní a lze je používat bez jakýchkoliv omezení. Soukromé adresy jsou unikátní jen v rámci dané sítě a mimo tuto síť se nesmí dostat (tj. nejdou za router).

Pro soukromé adresy se používají tyto adresní rozsahy:

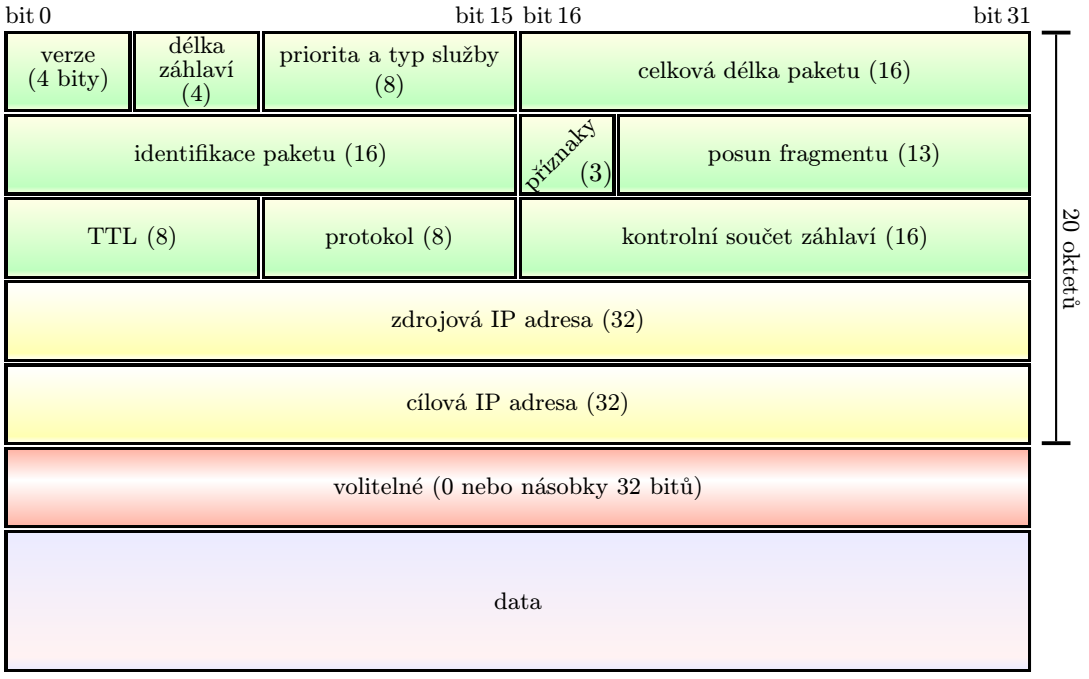
- `10.x.x.x`
- `172.16.x.x` až `172.31.x.x`
- `192.168.0.x` až `192.168.255.x`

Takže například pokud je prvním oktetem adresy číslo `10`, jde o soukromou adresu. O důvodu, proč jsou tyto rozsahy zrovna takové, si řekneme více o dvě kapitoly dál, a taky si tam vysvětlíme, jak se obchází fakt, že taková adresa „nesmí ven ze sítě“ – právě soukromé adresy se totiž využívají velmi hodně a tento mechanismus je pro IPv4 nezbytný.

Nedefinovaná adresa je adresa, kterou zařízení používá, když ve skutečnosti ještě žádnou adresu přidělenu nemá. Číselně to je `0.0.0.0`.

4.2.3 IPv4 pakety

Struktura IPv4 paketu je o něco složitější než struktura rámce nižší vrstvy, tedy si v nákresu nevystačíme s jediným „řádkem“.



Obrázek 4.1: Paket podle IPv4

Formát IPv4 paketu je naznačen na obrázku 4.1. Pokud někomu dělá problém pracovat s víceřádkovou strukturou, může si řádky naskládat do řady za sebe. Velikost polí je udána v bitech. Jednotlivá pole v záhlaví mají tento význam:

- *Verze* (Version, 4 bity) – verze protokolu IP, zde bude číslo 4, binárně 0100.
- *Délka záhlaví* (Header Length, 4 bity) – velikost záhlaví v 32bitových slovech, tedy počet řádků záhlaví (většinou tu najdeme číslo 5, binárně 0101).
- *Priorita a typ služby* (Priority and Type of Service, 8 bitů) – určuje, jak má být s tímto paketem zacházeno na cestě. První tři bity určují prioritu podle IEEE 802.1p, zbývající bity určují, zda má být při směrování vybrána spíše cesta s lepší hodnotou pro menší zpoždění zpracování, propustnost, spolehlivost apod.
- *Celková délka paketu* (Total Length, 16 bitů) – délka celého paketu včetně záhlaví a zapouzdřených dat v oktetech.
- *Identifikace paketu* (Identification, 16 bitů) – číslo přidělené paketu; odesílatel toto číslo přiděluje během odesílání a mělo by být pro každý odeslaný paket odlišné.
- *Příznaky* (Flags, 3 bity) – používají se pouze dva bity (třetí je rezervován), jejich význam si vysvětlíme později v sekci o fragmentaci.
- *Posun fragmentu* (Fragment Offset, 13 bitů) – v případě, že bylo třeba paket fragmentovat (rozdělit na menší části), je do tohoto pole uloženo číslo pro výpočet pozice části dat přenášené v tomto fragmentu vzhledem k původním nerozděleným datům (pozor, není tu přímo adresa, na to by nám 13 bitů nestačilo).
- *TTL* (Time to Live, 8 bitů) – životnost paketu. Na každém aktivním síťovém prvku s funkčností L3 (třeba routeru) se číslo v tomto poli vždy sníží o 1. V případě, že klesne na 0, je paket považován za bloudící a zahozen, přičemž je obvykle odesílatel o zahození informován.
- *Protokol* (Protocol/Type, 8 bitů) – informace o tom, co je v paketu zapouzdřeno jako data.
- *Kontrolní součet záhlaví* (Header Checksum, 16 bitů) – počítá se přes předchozí 2oktetové sekvence.
- *Zdrojová a cílová adresa* (Destination and Source Address, každá 32 bitů) – zdrojová adresa bývá vždy unicast.
- *Volitelné* (Options, 0 nebo násobky 32 bitů) – slouží k servisním účelům, například k ovlivnění směrování paketu sítěmi. Není povinné a obvykle toto pole nepoužíváme. Takže podle (ne)existence tohoto pole má pole Délka záhlaví hodnotu 5 nebo vyšší.
- *Data* (Payload) – přenášená data podle konkrétního protokolu.

Protože je pole pro celkovou délku paketu (čtvrté pole) dlouhé jen 16 bitů, je maximálním možným číslem $2^{16} - 1 = 65\,535$, z čehož vyplývá omezení pro maximální délku celého paketu. Omezení pro délku zapouzdřených dat získáme odečtením délky záhlaví.

V poli *Protokol* najdeme identifikátor protokolu, jehož datová jednotka je v IP paketu zapouzdřena. Svůj identifikátor pro toto pole mají například protokoly TCP (6) a UDP (17) z transportní vrstvy, ale také ty protokoly ze síťové vrstvy, které jsou zapouzdřovány do IP paketu, například ICMP (1) a směrovací protokoly, například OSPF (89).

Další informace:

Seznam všech hodnot pro pole Protokol najdeme na
http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml.

4.2.4 Pole TTL a životnost paketu

Jak bylo napsáno výše, v poli TTL (Time to Live) je číslo, které se na každém aktivním síťovém prvku pracujícím na vrstvě L3 snižuje většinou o 1. Účelem je omezit existenci nedoručitelných paketů, které by jinak do nekonečna bloudily sítí a zahlcovaly spoje (na vrstvě L2 máme k tomu účelu protokol STP, tady je třeba použít jiný mechanismus, protože smyčkám a redundantním spojům na L3 se prostě vyhnout nedá).

U některých protokolů, jejichž PDU se zapouzdřují do IP paketů, je přímo stanoveno, jaké TTL se má použít (například hodnotu 1 dáváme v případě, že paket nemá opustit síť). U jiných existují určité zvyklosti, jakou hodnotu konkrétně použít. Maximální hodnota je samozřejmě 255 (protože máme 8 bitů a $2^8 - 1 = 255$), ale většinou se používá číslo 64 nebo 128. Je to záležitost operačního systému (například Windows používají hodnotu 128).

Další informace:

Typické hodnoty pro pole TTL:

- http://subinsb.com/default-device-ttl-values
- http://www.binbert.com/blog/2009/12/default-time-to-live-ttl-values/

Poznámka:

Pole TTL se také nazývá „hop limit“, maximální počet přeskoků přes sítě. Tento druhý název se dokonce používá jako výchozí u následující verze protokolu – IPv6.

4.2.5 MTU a fragmentace IPv4 paketu

Definice (MTU)

MTU (Maximum Transmission Unit) je hodnota definovaná pro určitý spoj, která stanovuje maximální velikost paketu pro odeslání přes tento spoj. MTU konfigurované na konkrétním zařízení znamená maximální velikost paketu, který toto zařízení dokáže přijmout a zpracovat.

MTU je především ovlivněno nastavením na aktivních síťových prvcích, přes které je paket směrován, ale také záleží na konkrétním protokolu vrstvy L2, do kterého je IP paket zapouzdřen.

V případě IP paketu zapouzdřovaného na vrstvě L2 do ethernetového rámce máme jako maximum číslo 1500 (ale například pro Token Ring je toto číslo zhruba dvojnásobné). Protože naprostá většina (především lokálních) sítí používá na vrstvě L2 Ethernet, je podobná hodnota MTU nastavena i na mnoha aktivních síťových prvcích. Vzhledem k tomu, že data v IP paketu mohou být až 65 535 oktetů dlouhá (a k tomu musíme přičíst i IP záhlaví), je jasné, že tu máme značný nepoměr.

Definice (Jumbo rámec a jumbogram)

Jumbo rámec (jumbo frame) je ethernetový rámec, jehož velikost překračuje stanovenou maximální hodnotu. Zatímco maximum pro payload v běžném rámci je 1500 oktetů, u jumbo rámce to je obvykle 9000 oktetů. Může být odeslán pouze po takové přenosové cestě, kde jsou všechna zařízení na cestě nakonfigurována na přijímání jumbo rámců. U síťových rozhraní pro Gigabit Ethernet to obvykle není až takový problém, u nižších rychlostí už méně pravděpodobně.

Jumbogram je IP paket větší než 65 535 oktetů.

Do jumbo rámce sice naskládáme více než do obyčejného rámce, ale pro uložení opravdu velkého IP paketu to taky nestačí. Z toho důvodu potřebujeme jiný mechanismus: *fragmentaci IP paketu*, tedy možnost IP paket rozdělit na menší části (fragmenty) podle velikosti MTU a zajistit, aby cílové zařízení dokázalo tyto fragmenty zkompletovat do původního paketu.

Každý fragment se samozřejmě taky musí stát paketem, tedy ke každému fragmentu připojíme IP záhlaví. Většinu polí fragment „zdědí“ z původního paketu, ale některá pole budou jiná. Pro fragmentaci a především následné sestavení v cíli potřebujeme určité konkrétní hodnoty v polích druhého řádku paketu podle obrázku 4.1 na straně 80:

- *Identifikace paketu* (16 bitů) – všechny fragmenty mají toto pole stejné (zdědí po původním paketu), v cíli slouží k určení, které fragmenty patří k sobě,
- *Příznaky* (3 bity) – zajímají nás dva jednobitové příznaky:
  - DF (Do not Fragment) – abychom mohli fragmentovat, musí být tento příznak nastaven na 0; pokud odesílatel tento příznak nastaví na 1, zakáže fragmentaci po cestě,
  - MF (More Fragments) – tento příznak nastavíme na 1 ve všech fragmentech kromě posledního,
- *Posun fragmentu* (13 bitů) – u každého fragmentu zjistíme, na kterém oktetu původních dat začíná, toto číslo vydělíme 8 a uložíme sem; z toho vyplývá, že velikost dat ve fragmentu je vždy násobek 8 oktetů (abychom se vždy mohli „strefit“ na použitelnou adresu), případně kromě posledního.

Postup (Fragmentace IPv4 paketu)

Pokud je IP paket větší než kolik dovoluje MTU na cestě, musíme předně zkontrolovat, jestli vůbec můžeme fragmentovat. Jestliže je v záhlaví v poli *Příznaky* bit DF nastaven na 1, ani se s fragmentací neobtěžujeme a rovnou paket zahodíme. V opačném případě postupujeme takto:

1. Vypočteme délku dat pro fragment a počet fragmentů:
   - vezmeme hodnotu MTU na následné cestě a odečteme velikost záhlaví paketu (bývá obvykle 20 oktetů, ale pozor, může být větší),
   - vzniklé číslo vydělíme celočíselně osmi (tedy zaokrouhlíme směrem dolů), toto číslo si označme X,
   - maximální možná velikost dat zapouzdřených do paketu je tedy $X * 8$,
   - pokud je původní délka dat *po odečtení záhlaví* paketu M, pak počet fragmentů je $M/(X * 8) + 1$ s použitím celočíselného dělení, přičtená jednička je poslední fragment se „zbytkem“.

2. Vytvoříme první fragment:
   - většinu záhlaví zkopírujeme z původního paketu (včetně *Identifikace paketu*),
   - příznak MF (More Fragments) nastavíme na 1,
   - do pole *Posun fragmentu* uložíme číslo 0, protože tento fragment je první v pořadí a v posloupnosti oktetů z původního paketu se začíná právě adresou 0,
   - jako data vložíme do prvního fragmentu oktety s adresami $0 \ldots (X*8-1)$ (tedy prvních $X*8$ oktetů).
3. Vytvoříme další fragmenty kromě posledního (n je pořadí paketu, který právě zpracováváme, od $n = 2$):
   - opět přejmeme většinu záhlaví,
   - příznak MF (More Fragments) nastavíme na 1,
   - do pole *Posun fragmentu* uložíme číslo $X*(n-1)$ (obsah tohoto pole po vynásobení osmi dává adresu začátku tohoto fragmentu v původních datech),
   - jako data vložíme oktety s adresami $X*8*(n-1) \ldots X*8*n-1$ (tedy n-tých $X*8$ oktetů).
4. Vytvoříme poslední fragment ($n = M/(X*8)+1$):
   - přejmeme většinu záhlaví,
   - příznak MF (More Fragments) nastavíme na 0, protože další fragmenty už nebudou,
   - do pole *Posun fragmentu* uložíme číslo $X*(n-1)$,
   - jako data vložíme oktety s adresami $X*8*(n-1) \ldots M-1$.

Skládání fragmentů se provádí vždy až v cíli, přičemž se může stát, že některé fragmenty budou po cestě znovu fragmentovány. Cílové zařízení

- shromáždí všechny pakety s tímtéž identifikátorem paketu (první pole druhého řádku),
- seřadí je podle pole *Posun fragmentu*, přičemž poslední v pořadí by měl být fragment s příznakem MF nastaveným na 0,
- zkontroluje, zda některý fragment nechybí (u každého fragmentu porovná jeho délku s polem Posun fragmentu z následujícího fragmentu v pořadí).

Pokud vše souhlasí, seskládá fragmenty dohromady, pokud však najde chybu, všechny fragmenty zahodí a přenos se musí opakovat (v režii nadřízené vrstvy).

Celý postup si procvičíme na cvičeních.

Další informace:

http://www.root.cz/clanky/velke-trable-s-malym-mtu/

Poznámka:

Všimněte si, že tady nikde není zmínka o žádosti o opětovné poslání. Protokol IP je v principu „nespolehlivý“ a ničím takovým se nezabývá (poskytuje službu typu „best effort“). Znovuposlání si dojednají nadřízené vrstvy, pokud je to zapotřebí.

Na síťové vrstvě taktéž není navazováno spojení (alespoň to neprovádí protokol IP). Právě proto se v literatuře často píše o *IP datagramech* (místo IP paketů) – připomeňme si, že datagramová služba je služba bez navázání spojení.

4.3 Protokol IPv6

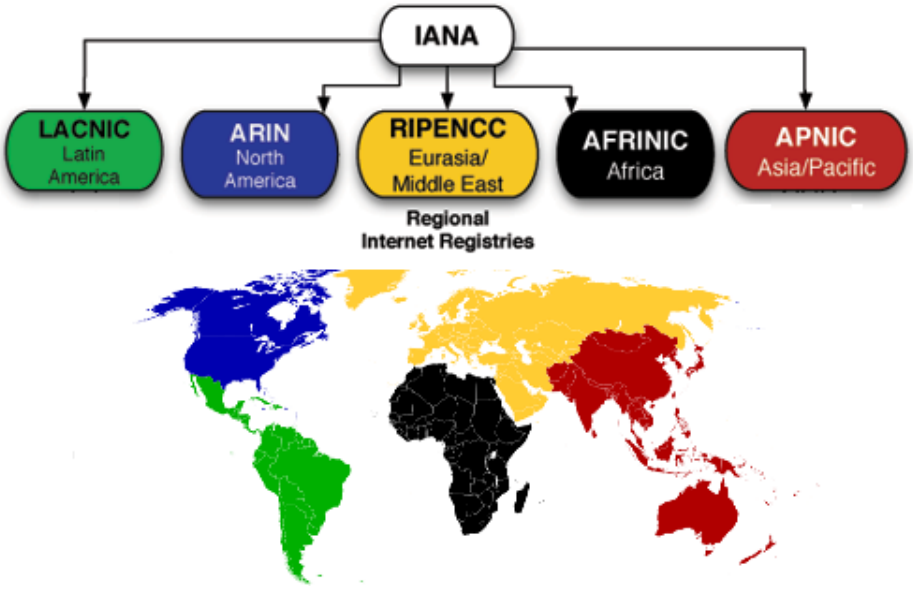
Protokol IP verze 6 (IPv6, také IPng – next generation) má vyřešit zoufalou situaci s akutním nedostatkem adres protokolu IPv4. Zatímco adresy IPv4 zabírají 32 bitů, adresy IPv6 jsou dlouhé 128 bitů, což by bohatě mělo stačit pro jakákoliv zařízení na světě (teoreticky jde o počet 10^{38} adres).

Ve skutečnosti je důvodů přechodu z IPv4 na IPv6 více než jen rozsah adres. Důležitá je například rozsáhlejší podpora zabezpečení – IPv4 tuto možnost vůbec nenabízí, ale v současné době je bezpečnost mnohem důležitější než v minulých desetiletích. Další důvody jsou zvýšená podpora pro mobilní zařízení, možnost zjednodušeného získání IP adresy pro koncová zařízení a další.

Protokoly IPv4 a IPv6 mohou být používány zároveň, dokonce i na tomtéž uzlu v síti.

Hlavním garantem přidělování IPv6 adres je *ICANN* (Internet Corporation for Assigned Network Numbers, http://www.icann.org), kdežto organizace *IANA* (Internet Assigned Numbers Authority, http://www.iana.org/) toto přidělování fyzicky provádí.

Celková struktura přidělování adres je hierarchická. IANA přiděluje bloky adres *regionálním registrátorům RIR* (Regional Internet Registery), což jsou RIPE (Evropa a část Asie), ARIN (Severní Amerika), AfriNIC (Afrika), LACNIC (Latinská Amerika), APNIC (Asie a Pacifik). Další patro hierarchie tvoří *lokální registrátoři* (LIR), kteří své bloky získávají od regionálních registrátorů. Od lokálních registrátorů pak své rozsahy adres získávají zákazníci nebo další subjekty, které mohou své rozsahy dále distribuovat.



Obrázek 4.2: Základ struktury přidělování IP adres[1]

Na obrázku 4.2 jsou naznačena horní dvě patra této struktury. Níže jsou registrátoři pro jednotlivé země LIR, což mohou být třeba ISP (poskytovatelé Internetu), následují jejich zákazníci a drobní poskytovatelé Internetu.

Další informace:
Seznam LIR pro Evropu je na https://www.ripe.net/participate/member-support/info/list-of-members/europe, seznam pro ČR získáte tak, že v mapě klepnete na „Czech“.

4.3.1 Adresy IPv6

Adresa podle IPv6 je 128 bitů dlouhá (tj. 16 oktetů, čtyřnásobek IPv4 adresy) a skládá se ze dvou částí – *prefixu* a *identifikátoru síťového rozhraní* (adresy uzlu v rámci jednoho prefixu), což je vlastně podobné tomu, co známe u IPv4.

Poznámka:
Také se jedná o hierarchické adresování – adresy odrážejí fyzickou strukturu sítě propojené směrovači, a to dokonce i v globálním měřítku. IANA přiděluje základní rozsahy (prvních 12 bitů) jednotlivým regionálním registrátorům RIR, ti své rozsahy dále distribuují lokálním LIR. Další dělení provádějí poskytovatelé Internetu a samozřejmě pro své podsítě jednotlivé organizace.

Protože jsou IPv6 adresy hodně dlouhé, zapisujeme je hexadecimálními číslicemi ve skupinách po čtyřech číslicích (tj. po dvou oktetech) oddělené dvojtečkou. Z toho vyplývá, že adresa bude mít osm částí oddělených dvojtečkami.

Příklad
IPv6 adresa může vypadat takto:

- `a4cb:57b1:60aa:000E:113a:b201:042a:02b1`
- `a4cb:57b1:60aa:E:113a:b201:42a:2b1`
- `2001:0db8:3c4d:0000:0000:a010:0000:0000`
- `2001:db8:3c4d:0:0:a010:0:0`

Všimněte si, že ve dvojicích adres pod sebou je vlastně spodní adresa stejná jako horní, jen jsme v jednotlivých částech odstranili nuly zleva. Totéž jsme samozřejmě mohli udělat i u IPv4.

Dvojtečky oddělují celkem osm částí adresy, přičemž každá část je zapsána maximálně čtyřmi hexadecimálními číslicemi, tedy zabírá dva oktety.

Oproti IPv4 lze zápis zjednodušit *odstraněním posloupnosti nulových skupin oktetů*. V jedné adrese tak můžeme odstranit pouze jednu posloupnost nulových skupin a místo odstranění musí být označeno dvojitou dvojtečkou.

Příklad
Opravdu lze krátit pouze jednu jedinou posloupnost nulových skupin. Například adresu
`2001:0db8:3c4d:0000:0000:a011:0000:0000`

[1] Zdroj: http://whitengreen.com/blog-1124-how-to-trace-and-locate-ip-addresses

lze zkrátit jedním z těchto způsobů:
`2001:0db8:3c4d:0000:0000:a011::` (respektive `2001:db8:3c4d:0:0:a011::`)
`2001:0db8:3c4d::a011:0000:0000` (respektive `2001:db8:3c4d::a011:0:0`)
špatně by bylo `2001:0db8:3c4d::a011::`
protože by taková adresa byla nejednoznačná. Podle počtu částí adresy je zřejmé, že nulové jsou celkem čtyři části (celkem jich musí být 8), ale když v adrese vidíme dvě místa krácení, nemůžeme přesně říci, jak jsou mezi ně tyto čtyři nulové části rozmístěny. Mohla by to být například i chybná možnost `2001:0db8:3c4d:0000:a011:0000:0000:0000`

Kanonický tvar IPv6 adresy je standardizován jako RFC 5952 a předepisuje tyto podmínky:

- hexadecimální číslice se mají zapisovat malými písmeny,
- vynechávání počátečních nul ve skupině je povinné (takže v prvním příkladu na straně 86 by byl správně zkrácený tvar na druhém řádku každého sloupce),
- mechanismus zkrácení počtu skupin pomocí `::` musí mít co největší efekt, což znamená, že pokud máme víc řad nulových skupin, vybírá se ta delší, když je víc stejně dlouhých, vybereme tu víc vlevo (tj. v druhém příkladu na straně 86 je v kanonickém tvaru pouze výsledek `2001:db8:3c4d::a011:0:0`), a musí pohltit všechny dosažitelné nulové skupiny; pokud je nulová skupina v adrese jen jedna, konstrukce `::` se nepoužije (žádná skupina není odstraněna).

Kanonický tvar adresy je jednoznačný.

4.3.2 Speciální adresy podle IPv6

Podle IPv6 rozlišujeme tyto typy adres:

- unicast (konkrétní uzel v síti),
- multicast (skupinové),
- anycast (adresace komukoliv ze zadané skupiny).

Broadcast adresy již nejsou podporovány.

Co se týče konkrétních adres zařízení, existují tyto možnosti:

- *Unique Local Address* (ULA adresy) – slouží k posílání unicast dat v rámci lokální sítě (organizace apod.), je to obdoba soukromé IP adresy v IPv4, nesmí být viditelná mimo lokální síť, ale je zde víceméně zaručena unikátnost (narozdíl od následujícího typu adresy). ULA adresy mají prefix `fd00::/8`, takže hexadecimálně vždy začínají oktetem `fd`.
- *Link Local Address* (lokální na segmentu) – významem odpovídají soukromým adresám podle IPv4, není zaručena unikátnost (v jiné síti může existovat zařízení s naprosto stejnou linkovou lokální adresou), ale je unikátní alespoň v rámci segmentu. Pakety s touto adresou jako cílovou nepřecházejí přes router. Tyto adresy mají vždy prefix `fe80::/10`, takže prvních deset bitů je `1111 1110 10` (pozor, první oktet je jasný – `fe`, ale dál už je to složitější, třetí hexadecimální číslice může být `8`, `9`, `a` nebo `b`).
- *Globální adresy* – jsou vždy unikátní v rámci celého Internetu, jejich prefix je vždy `2000::/3`, tedy v binárním zápisu začínají třemi bity `001` a hexadecimálně je první nibble (čtveřice bitů) na hodnotě `2` nebo `3`.

Unikátnost ULA adresy se zajišťuje odvozením z data (času) generování adresy a z MAC adresy stanice.

Původně se počítalo ještě s tzv. *site local* adresou, která by fungovala přesně jako lokální adresy v IPv4 (včetně překladu adres), její prefix je `feC0::/10`. Nicméně tento typ adres byl velmi brzy ze standardu vyňat a jediný, kdo s ním počítá, je společnost Microsoft v některých svých technologiích.

Loopback (zpětná či lokální smyčka) má tentýž význam jako u IPv4, tedy jde o testovací adresu znamenající „pohled zvenčí na sebe sama“. Adresa loopbacku v IPv6 je `::1/128`, což bychom mohli přepsat jako `0:0:0:0:0:0:0:1`.

Nedefinovaná adresa (tedy informace, že stanice nemá přiřazenu adresu) je `::/128`, což v přepisu znamená `0:0:0:0:0:0:0:0`.

Také v IPv6 se používají *skupinové adresy* (pro multicast), jejich prefix je vždy `ff00::/8` (to znamená, že prvních osm bitů je nastaveno na 1). Některé multicast adresy jsou vyhrazeny, například:

- `ff02::1` – skupinová adresa všech uzlů sítě podporujících IPv6,
- `ff02::2` – skupinová adresa všech dostupných routerů (směrovačů),
- `ff02::1:2` – skupinová adresa představující všechny dostupné DHCP servery (od nich lze získat dynamickou IP adresu).

Další informace:
Seznam všech dobře známých a registrovaných multicast adres je na
http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xhtml.

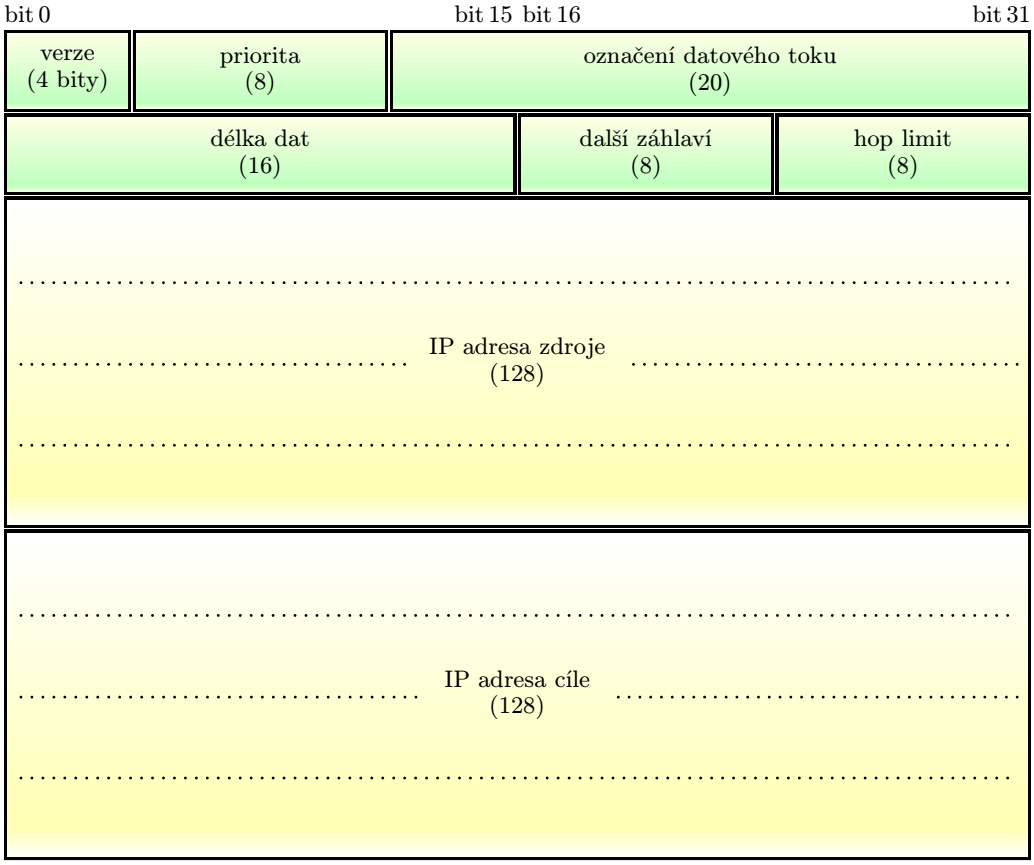
Anycast adresy jsou v principu sice také skupinové (s tím rozdílem, že adresátem není každý člen skupiny, ale jen jeden z nich), ale ve skutečnosti se s nimi pracuje trochu jinak a nemohou označovat koncová zařízení (hosty). Nemají vyhrazený prefix, na první pohled je nijak nerozeznáme od unicast adresy.

Anycast adresy se globálně prakticky nepoužívají (pro routery by to bylo zbytečně složité), spíše se s nimi setkáme v lokálních sítích. Odlišné zacházení než s unicast adresami se zajistí konfigurací v směrovacích tabulkách na síťových zařízeních (routerech apod.).

4.3.3 IPv6 pakety

Struktura IP paketu verze 6 je oproti verzi 4 značně pozměněná. Hlavní odlišnost je struktura záhlaví. Zatímco záhlaví IPv4 paketu je jen jedno a může mít různou délku (podle pole *Volitelné*), v paketu podle IPv6 máme jedno povinné záhlaví o pevné délce (několik nejdůležitějších polí) a dále můžeme přidat volitelná záhlaví, z nichž každé má svůj stanovený účel.

Povinné záhlaví pevné délky je naznačeno na obrázku 4.3. Jak vidíme, první pole je stejné jako u předchozí verze (číslo verze), jen jeho obsah bude samozřejmě jiný – u paketu podle IPv4 tam bude uložena hodnota 4 (binárně 0100), kdežto u paketu podle IPv6 tam bude hodnota 6 (binárně 0110). Další pole jsou už úplně jiná, ovšem přijímající zařízení už podle toho prvního pole rozhodne, o jakou verzi jde a jaká pole má očekávat dále.



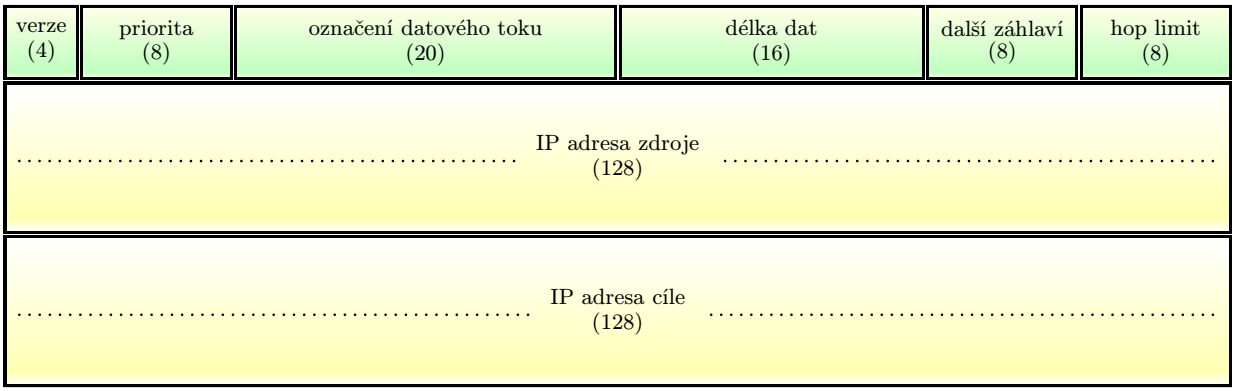
Obrázek 4.3: Paket podle IPv6 – povinné záhlaví

Význam jednotlivých polí:

- *Verze* (Version, 4 bity) – verze protokolu (tedy 6).
- *Priorita* (Priority, 8 bitů) – podobně jako pole *Typ služby* u IPv4, jednotlivé bity umožňují optimalizaci priorit.
- *Označení datového toku* (Flow Label, 20 bitů) – stanovuje způsob „speciálního zacházení“ na směrovačích pro některé typy protokolů, s pakety patřícími do téhož datového toku se má zacházet stejně (žádný datový tok: = 0). Pokud se používá, pak zároveň s předchozím polem (pro všechny pakety téhož datového toku platí stejná priorita).
- *Délka dat* (Payload Length, 16 bitů) – délka zbytku paketu (bez povinného záhlaví), tj. všech volitelných záhlaví a vlastních dat; pokud je zde hodnota 0, jedná se o *jumbogram*.
- *Další záhlaví* (Next Header, 8 bitů) – určuje typ následujícího (volitelného) záhlaví téhož paketu nebo typ zapouzdřených dat.
- *Hop limit* (8 bitů) – obdoba TTL u IPv4, označuje maximální počet směrovačů či jiných zařízení vrstvy L3 na cestě, na každém síťovém prvku se snižuje o 1.
- *Zdrojová a cílová IP adresa* (obě 128 bitů, tj. pro každou adresu čtyři řádky podle obrázku 4.3).

Na obrázku 4.4 je vlastně totéž jako na předchozím, jen je zdvojnásobena velikost řádku z 32 bitů na 64. Zatímco před lety byly výpočetní systémy (počítače, servery apod.) 32bitové, dnes jsou již téměř výhradně 64bitové a tedy se do procesoru (a jinam) načítá vždy najednou 64 bitů. IPv6

je optimalizován právě pro 64bitové zpracování dat, a jak vidíme, téměř všechna pole povinného záhlaví (všechna zeleně podbarvená) se načtou při jediném přístupu, každá z adres pak po dvou přístupech.



Obrázek 4.4: Paket podle IPv6 – povinné záhlaví pro řádek 64 bitů

Volitelná záhlaví následují za povinným záhlavím v předem daném pořadí (tedy pořadí je důležité, RFC 2460), některá (nebo i všechna) mohou být vynechána. Každý typ volitelného záhlaví má své číslo, toto číslo najdeme v poli *další záhlaví* předchozího záhlaví (tj. v povinném záhlaví zjistíme, jakého typu je první volitelné záhlaví, v prvním zjistíme typ druhého volitelného, atd.).

Některá z používaných volitelných záhlaví:

- *Hop-by-hop Options Header* (0, informace pro směrovače na cestě, směrovače čtou z volitelných jen toto záhlaví),
- *Routing Header* (číslo 43, použijeme, pokud chceme „natvrdo“ předepsat cestu, zde mohou být určeny směrovače, přes které má cesta vést, v obráceném pořadí je závazné i pro odpověď),
- *Fragment Header* (44, pokud je paket na zdrojové stanici fragmentován, je použito toto záhlaví s informací o fragmentaci, podobné údaje jako v záhlaví IPv4; stanice musí znát hodnoty MTU na cestě),
- *Encapsulating Security Header* (50, údaje o šifrování),
- *Authentication Header* (číslo 51, obsahuje autentizační informaci),
- TCP segment (6), UDP segment (17), ICMP paket (58), ...

Poznámka:

Všimněte si, že v poslední odrážce předchozího seznamu máme kódy pro TCP segment, UDP segment, ICMP paket atd. Z toho vyplývá, že pole *další záhlaví* plní v IPv6 paketu také roli pole protokol z IPv4 paketu – říká, co je zapouzdřeno uvnitř (je zajímavé, že pro ICMPv6 je jiný identifikátor než pro ICMPv4).

Takže skutečný význam pole *další záhlaví* je: určuje, co konkrétně následuje za tím záhlavím, ve kterém toto pole právě čteme. Může to být buď některé volitelné záhlaví nebo přímo data předaná konkrétním protokolem z transportní nebo síťové vrstvy.

Fragmentace IPv6 paketu může být provedena pouze odesílajícím zařízením, žádné zařízení na cestě ji nesmí provést. Pokud některý mezilehlý síťový prvek (router nebo switch s funkcionalitou L3) zjistí, že daný IPv6 paket je větší než kolik dovoluje MTU na cestě, pak tento paket zahodí, jinou alternativu nemá. Ovšem fragmentovat může odesílatel, a tehdy použije volitelné záhlaví číslo 44, které je pro tento účel určeno a obsahuje podobné informace jako IPv4 paket v druhém řádku podle obrázku 4.1.

4.4 Protokol ICMP a zprávy

4.4.1 Účel protokolu ICMP

Protokol ICMP (Internet Control Message Protocol, tedy protokol pro posílání řídících zpráv) je také protokolem vrstvy L3. Jeho účelem je zajišťovat posílání krátkých zpráv, obvykle reprezentovaných jedním nebo dvěma čísly (každé takové číslo má určitý význam), výjimečně ještě s dodatečnou datovou informací. Typicky se takto posílají informace o chybách, problémech na cestě nebo jiná upozornění.

Každá zpráva posílaná pomocí ICMP má své *číslo*. V následující tabulce jsou některé běžné hodnoty pro ICMPv4 uvedeny:

Číslo	Význam
8	*Echo Request* – žádost o odezvu (posíláme, když příkazem `ping` zjišťujeme, zda je konkrétní zařízení dostupné)
0	*Echo Reply* – odpověď na *Echo Request* (8) ve smyslu „ano, jsem tady“
3	*Destination Unreachable* – zpráva o nedostupnosti cíle (pokud router obdrží IP paket, který nedokáže doručit, pošle ICMP paket se zprávou 3 zdrojovému zařízení)
11	*Time Exceeded* – vypršel čas čekání; buď hodnota TTL v IP paketu klesla na 0, vypršel časový limit při čekání na zbývající fragmenty IP paketu apod. (posílá router zdroji IP paketu)
12	*Parameter problem* – v IP paketu (většinou v jeho záhlaví) je nějaký problém, který nelze oznámit jinou ICMP zprávou
17	*Address Mask Request* – žádost o informaci o masce (pod)sítě
18	*Address Mask Reply* – odpověď na *Address Mask Request* (17)

Tabulka 4.1: Některé zprávy protokolu ICMPv4

Samotná zpráva je v některých případech příliš obecná, tedy ji může upřesnit kód zprávy. Například pro zprávu typu 3 (*Destination Unreachable*) existuje několik kódů, které upřesňují, proč je cílové zařízení nedostupné:

- kód 0 – cílová síť je nedostupná (router tuto síť nezná a nedokáže ji směrovat),
- kód 1 – cílová stanice je nedostupná (v zadané síti nelze najít dotyčnou stanici),
- kód 2 – cílový protokol je nedostupný (v poli *Protokol* je neznámá hodnota),
- kód 3 – cílový port je nedostupný (tím je myšleno číslo portu uvedené v zapouzdřeném segmentu transportní vrstvy),

- kód 4 – paket je větší než MTU na cestě, ale nelze fragmentovat, protože je nastaven příznak DF (Do not Fragment),
- atd. (pro zprávu 3 je celkem 15 různých kódů).

Pro zprávy typu 11 (*Time Exceeded*) se obvykle používají kódy 0 (hodnota pole TTL je 0) nebo 1 (překročen čas čekání na zbývající fragmenty).

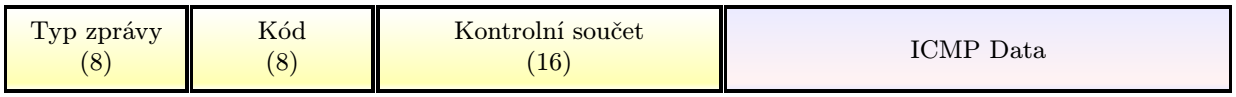
Takže zatím máme dvojici číslo zprávy + kód. Zpráva ale může být upřesněna ještě jinak, například *jako data* může být přiložena část IP paketu, na který je takto reagováno (přiloží se IP záhlaví a prvních 8 oktetů datové části).

Poznámka:
Ve skutečnosti mají datovou část i ty ICMP pakety, které ji vlastně ani nepotřebují – přidávají tam výplň, protože na nižších vrstvách bývá stanovena určitá minimální délka pro zapouzdřená data (payload). S tím se setkáváme například u ICMP zprávy *Echo Request* (8).

ICMP paket má velmi jednodouchou strukturu – žádné adresy, délka paketu apod., záhlaví má jenom tři pole:

- *Typ zprávy* (ICMP Type, 8 bitů) – určení typu zprávy, například číslo 8 pro Echo Request,
- *Kód* (Code, 8 bitů) – upřesňující kód,
- *Kontrolní součet* (Checksum, 16 bitů) – kontrolní součet přes předchozí dvě pole záhlaví.

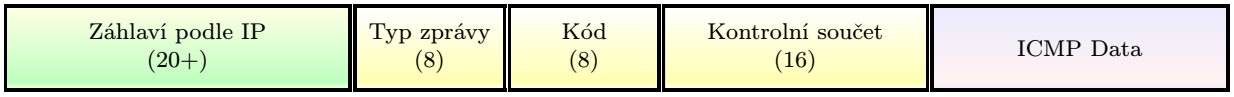
Formát celého ICMP paketu vidíme na obrázku 4.5.



Obrázek 4.5: Formát ICMP paketu

Dále v této kapitole se podíváme na konkrétní případy použití ICMP včetně toho, co je v těchto případech použito jako data.

ICMP paket se sice při odesílání předává na vrstvu L2, ale ještě předtím se zabalí do IP paketu – samotný ICMP paket nemá ve svém záhlaví ani adresy ani jiné důležité položky. Na obrázku 4.6 vidíme formát paketu do zapouzdření do IP.



Obrázek 4.6: Formát ICMP paketu zapouzdřeného v IPv4 paketu

Některé typy zpráv používají TTL = 1, ale většina využívá hodnotu TTL nastavenou v odesílajícím systému (tj. obvykle 128 nebo 64).

Po zapouzdření do IP je paket zpracován na L2 jako každý jiný paket, tj. například je zapouzdřen do paketu typu Ethernet II a poslán po ethernetové síti.

4.4.2 ICMP verze 6

Pokud používáme protokol IPv6, pak je třeba také používat ICMPv6. Oproti ICMPv4 byly některé zprávy přidány a taky se rozšířily oblasti využití tohoto protokolu (což souvisí) – v protokolu ICMPv6 se shrnuly role tří původních protokolů: ICMPv4, ARP a IGMP.

Jenže zdaleka nešlo jen o přidání typů zpráv pro nové způsoby využití protokolu, mnohé původní typy byly přečíslovány. Už u ICMPv4 platilo, že některé zprávy jsou chybové a jiné informační, ICMPv6 toto rozlišení přenesl i do číslování typů zpráv – pole pro typ zprávy je taky 8bitové, ale první (nejlevější, nejvíce významný) bit je nastaven na 0 u chybových zpráv a na 1 u informačních zpráv.

Číslo	Význam
128	*Echo Request* – žádost o odezvu, původně číslo 8
129	*Echo Reply* – odpověď na *Echo Request* (128), původně číslo 0
1	*Destination Unreachable* – zpráva o nedostupnosti cíle, původně číslo 3
3	*Time Exceeded* – vypršel čas čekání, původně číslo 11
4	*Parameter problem* – v IP paketu je nějaký problém, který nelze oznámit jinou ICMP zprávou, původně číslo 12

Tabulka 4.2: Některé zprávy protokolu ICMPv6, které byly i u ICMPv4

V tabulce 4.2 je několik typů zpráv, které najdeme v obou verzích. Všimněte si, že tyto zprávy mají v ICMPv6 jiná čísla než v ICMPv4, a že při novém číslování zachovávají pravidlo, kdy nejvíce významný bit z 8 bitů prvního pole je nastaven na 0 u chybových zpráv (tj. číslo typu je v rozmezí 0–127, poslední tři řádky tabulky) a na 1 u informačních zpráv (rozmezí 128–255, první dva řádky tabulky). Také v kódech pro některé konkrétní typy zpráv jsou změny (ale třeba kódy pro zprávu Time Exceeded jsou stejné, jen číslo typu se změnilo).

Nově přidané typy zpráv souvisejí obvykle buď s mapováním IP adres na MAC adresy (což u starší verze byla práce protokolu ARP) nebo se správou skupin (původně v protokolu IGMP), a taky pár typů zpráv pro běžné nebo mobilní sítě. Nejdůležitější z nich jsou v tabulce 4.3.

Zajímavá je například zpráva *Packet Too Big* (2), která v předchozí verzi nebyla, třebaže by se vpodstatě hodila i tam. Pokud router dostane k přeposlání paket větší než je hodnota MTU na následující cestě, chová se u různých verzí IP odlišně.

- Je to paket protokolu IPv4: buď se ho pokusí fragmentovat, nebo (když to nejde, příznak DF = 1) paket zahodí a jeho původci pošle ICMPv4 zprávu *Destination Unreachable* (3).
- Je to paket protokolu IPv6: v každém případě ho zahodí a jeho původci pošle ICMPv6 zprávu *Packet Too Big* (2).

Další informace:

Seznam všech typů ICMP zpráv a k nim příslušných kódů najdete na

- IPv4: http://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml
- IPv6: http://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml

4.4.3 Testování dosažitelnosti

Protokol ICMP se často používá pro posílání zpráv *ICMP Echo Request* a *ICMP Echo Reply*.

Pokud potřebujeme otestovat dostupnost některého zařízení v síti, obvykle k tomu účelu používáme příkaz `ping` odesílající právě ICMP zprávu *Echo Request* (číslo 8 nebo 128 podle verze). Příkaz `ping` posílá vždy několik paketů s touto zprávou, na každý očekává odpověď a následně podle toho, jak dlouho trvala odpověď na jednotlivé pakety, vypočte statistiky. Tento příkaz najdeme v jakémkoliv operačním systému; do Windows se dostal z Unixu, což je vidět na jeho syntaxi.

Příkaz `ping` posílá ICMP zprávu *Echo Request* v tomto tvaru:

Každý řádek má délku 32 bitů. Pole na prvním řádku jsou součástí ICMP záhlaví, od druhého řádku dále jde o data specifická pro tento typ zprávy. U ICMP *Echo Request* posíláme

- *identifikátor* (16 bitů) – většinou je zde nějaké velmi malé číslo, obvykle 1 nebo 3,
- *pořadové číslo* (Sequence Number, 16 bitů) – pořadové číslo paketu posílaného příkazem `ping`, takže když pošleme čtyři pakety, budou v nich postupně čísla zvyšující se o 1, pokud příkaz spustíme znovu, pokračuje se v posloupnosti (nezačíná se číslovat znovu),
- *volitelná data* (ICMP Data) – datová výplň bez významu, obvykle o délce 32 nebo 48 oktetů.

Volitelná data si zařízení určuje samo. Ve Windows je to abeceda (písmena anglické abecedy v rozsahu `a...wa...i`, všimněte si, že podle Microsoftu je posledním písmenem abecedy písmeno `w`), v Linuxu sekvence speciálních znaků a číslic.

Testované zařízení odpoví ICMP zprávou *Echo Reply*, která víceméně kopíruje pole dotazu (změní obsah pole *Typ zprávy* a samozřejmě kontrolní součet). Tato zpráva je doručena zpět tá-

Číslo	Význam
2	*Packet Too Big* – v případě, že router dostane k přeposlání příliš velký paket (větší než hodnota MTU na portu pro odeslání), paket zahodí a původnímu odesílateli pošle tuto ICMP zprávu
134	*Router Advertisement* (vizitka routeru) – routery v pravidelných intervalech posílají tuto ICMP zprávu s informacemi o sobě a síti, například adrese sítě a délce prefixu
133	*Router Solicitation* (žádost o vizitku routeru) – stanice si touto zprávou může na routeru vynutit vyslání zprávy Router Advertisement (134) mimo interval
135	*Neighbor Solicitation* (žádost o oznámení souseda) – máme IP adresu některého zařízení v sousedství, touto zprávou se ptáme svého okolí na jeho MAC adresu (tj. „Kdo má tuto IP adresu?“, soused, který pozná svou IP adresu, odpoví následující zprávou:
136	*Neighbor Advertisement* (oznámení souseda) – v reakci na předchozí zprávu (pokud poznám svou iP adresu) oznámím svou MAC adresu
130	*Multicast Listener Query* (dotaz na členy skupiny) – posílá router při ověřování, kdo z jeho sítě patří do konkrétní skupiny a tedy odebírá pakety adresované na danou multicast adresu
143	*Multicast Listener Report* (oznámení členství ve skupině) – posílá zařízení, když se přihlašuje do skupiny (informuje router, že chce dostávat příslušné pakety) nebo jako odpověď na předchozí zprávu

Tabulka 4.3: Některé zprávy protokolu ICMPv6, které nejsou v ICMPv4

Typ = 8	Kód = 0	Kontrolní součet
Identifikátor		Pořadové číslo
volitelná data		

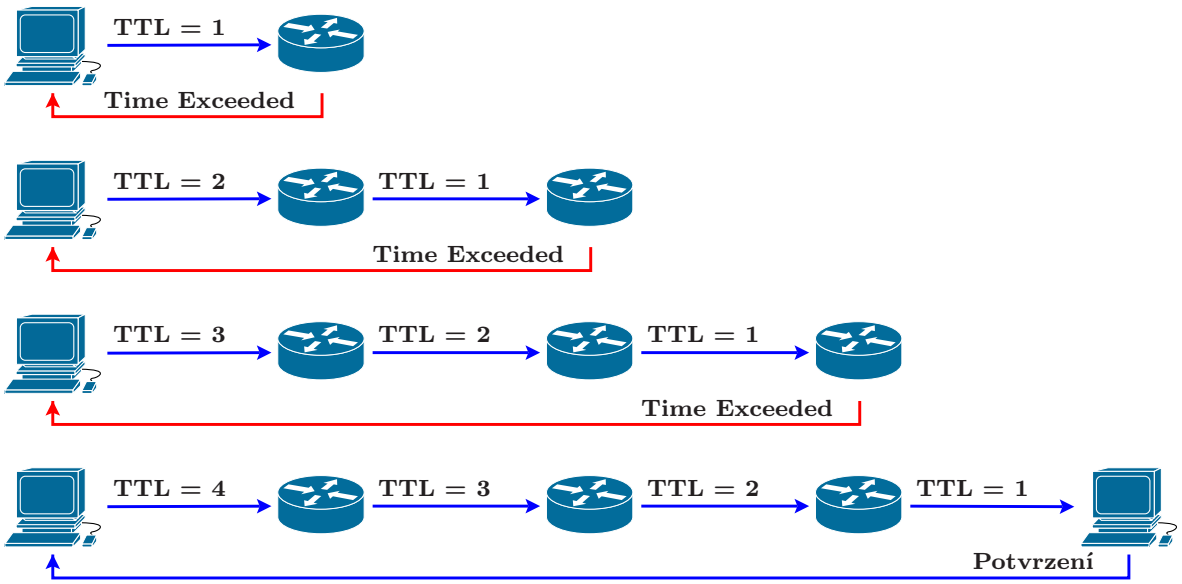
Tabulka 4.4: ICMPv4 paket zprávy *Echo Request*

zajícímu se zařízení, které podle hodnoty pole *pořadové číslo* v jednotlivých paketech odpovědí zkompletuje časy odeslání dotazu a přijetí odpovědi a vypočte příslušnou statistickou informaci.

Poznámka:
Na cvičeních vyzkoušíte příkaz `ping` zároveň se spuštěným Wiresharkem a prohlédnete si celou komunikaci v obou směrech.

Mechanismus `traceroute` (ve Windows `tracert`) také pracuje s ICMP pakety. Na rozdíl od předchozího příkazu zde ani tak nejde o zjištění dostupnosti dotyčného zařízení, ale spíše o zmapování cesty (trasování, proto ten název). Pokud je cíl dostupný, vypíše se seznam všech routerů na cestě včetně doby trvání přenosu od předchozího uzlu sítě, pokud je cíl nedostupný, vypíše se seznam routerů na té části cesty, která je „v pořádku“. Takže pomocí `traceroute` můžeme zjistit:

- kterou cestou (přes které routery) jdou pakety k určitému cíli,
- který úsek této cesty je nejpomalejší (v síti to může znamenat problém s propustností),
- od kterého routeru začíná být cesta problematická.



Obrázek 4.7: Obecně k mechanismu traceroute

Obecně tento mechanismus funguje takto (viz obrázek 4.7):

- z našeho zařízení se odesílají IP pakety s cílovou adresou „zájmového“ zařízení s TTL nastaveným na 1, 2, 3, atd.,
- na každém zařízení vrstvy L3 na cestě se hodnota TTL snižuje o 1,

- na tom zařízení vrstvy L3, kde po odečtení 1 je TTL = 0, už nedojde k přeposlání, náš paket je zahozen,
- zpět nám přijde ICMP zpráva Time Exceeded (v IPv4 číslo 11, v IPv6 číslo 3),
- až se konečně některý paket dostane k cíli, získáme potvrzující odpověď a přestaneme posílat testovací pakety.

Pro totéž TTL se obvykle posílá více než jeden paket, většinou tři pakety pro každé TTL.

Ve Windows se posílají testovací pakety ve formě ICMP zpráv *Echo Request* zabalených do IP paketu s daným zvyšujícím se TTL, potvrzení od cílové stanice je ICMP zpráva *Echo Reply*. Takže podle této zprávy poznáme, že jde o poslední uzel na cestě, tedy cíl.

V Linuxu si můžeme vybrat – ve výchozím nastavení to jsou UDP segmenty zabalené do IP paketu s daným zvyšujícím se TTL, přičemž jako číslo portu v UDP segmentu je použito „nereálné“ číslo. Proto je potvrzením od cílové stanice ICMP zpráva *Destination Unrecheable* s kódem *Port Unreachable* (nedostupný port). Podle tohoto paketu poznáme, že náš dotaz dorazil do cíle. V parametrech příkazu můžeme zvolit jiný druh testovacích paketů – buď ICMP pakety jako ve Windows nebo TCP segmenty.

Poznámka:

Mnohé servery a jiná veřejně dostupná síťová zařízení na příkazy `ping` a `traceroute` (`tracert`) nereagují. Je to proto, že buď samotné zařízení nebo firewall na cestě má v konfiguraci nastaveno nereagovat na zprávy ICMP Echo Request, případně na UDP segmenty s neexistujícím číslem portu. Důvodem je bezpečnost, protože tyto mechanismy bývají často zneužívány hackery k mapování „zájmové“ sítě.

Většinou je dobré nechat reakci na tyto pakety alespoň tehdy, když pocházejí z vnitřní (důvěryhodné) sítě, ovšem to je na administrátorovi.

Ovšem v Linuxu můžeme zvolit TCP segmenty používající známý port 80 (protokol HTTP), ten obvykle přes firewally projde.

Tyto testovací mechanismy fungují na vrstvě L3, takže pokud je problém s odezvou a dostupností na jiné vrstvě (ať už nadřízené nebo na kterémkoliv zařízení na cestě v podřízené vrstvě), nemáme šanci to zjistit.

Další informace:

ftp://ftp.hp.com/pub/hpcp/UDP-ICMP-Traceroutes.pdf

4.5 Komunikace typu klient-server

Základní schéma komunikace mezi dvěma uzly může být buď typu klient-server nebo typu peer-to-peer.

V *komunikaci typu klient-server* je pevně určeno, kdo službu poskytuje (uzel typu server) a kdo službu využívá (uzel typu klient). Komunikaci zahajuje klient svou *žádostí* (dotazem), server

následně klientovi zašle *odpověď*. Obvykle není přípustné, aby v tomto modelu komunikaci zahajoval server.

Komunikace typu peer-to-peer probíhá mezi rovnocennými uzly, které v síti mohou plnit roli klienta i serveru. Komunikaci může zahájit kterýkoliv uzel.

V počítačových sítích se většinou setkáváme právě s komunikací typu klient-server.

4.6 Transportní vrstva

Transportní vrstva je vyrovnávací vrstvou mezi vrstvami orientovanými na přenos (L1–L3) a vrstvami orientovanými na aplikace (L5–L7). Sice se podílí na vyjednávání přenosu dat (čímž je blízká nižším vrstvám), ale žádným způsobem neadresuje zařízení ani sítě. Na druhou stranu se sice podílí na určení konkrétní komunikující aplikace (čímž je blízká vyšším vrstvám), ale je jí naprosto lhostejný formát dat a se všemi druhy dat zachází vpodstatě stejně, aplikace je pro protokoly této vrstvy jednoduše jen číslo, nic víc.

4.6.1 Čísla portů

Na transportní vrstvě se neadresují počítače ani sítě (směrem dolů), adresují se aplikace (směrem nahoru). Každá aplikace musí být jednoznačně určena *číslem portu*, a protože komunikující strany jsou dvě, potřebujeme číslo portu zdroje a číslo portu cíle. Znovu připomínáme, že pojem port na vrstvě L4 není totéž co pojem port na nižších vrstvách.

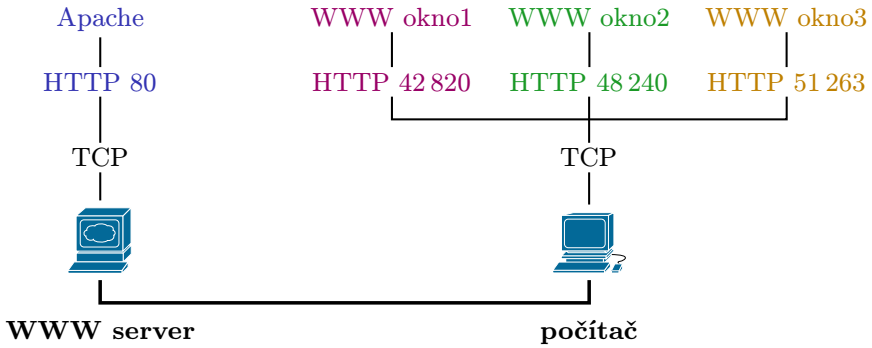
Číslo portu zabírá dva oktety, což nám dává rozsah 0 – 65 535. Z toho jsou

- *porty dobře známé* (well-known) v rozsahu 0–1023, tato čísla se používají pro konkrétní běžně známé služby na serverech (WWW/HTTP, FTP, RPC, SQL, Syslog, Kerberos, atd.),
- *porty registrované* (registered, official) v rozsahu 1024 – 49 151, které si mohou různé společnosti registrovat u organizace IANA (například své registrované porty má Microsoft, IBM, Citrix, Novell, Cisco, Oracle, Eset), sem také spadají porty pro RADIUS, Nessus, nebo BOINC,
- *dynamické* (soukromé) ve zbývajícím rozsahu 49 152–65 535 jsou používány na straně klienta, který komunikuje se serverem.

Poznámka:

Proč toto rozlišení? První dvě skupiny portů jsou určeny pro serverovou stranu komunikace, přičemž na straně serveru obvykle běží jediná instance dotyčné služby (aplikace) – například na web serveru běží jediná aplikace poskytující webové služby (například Apache nebo IIS), tedy nám pro danou aplikaci na zařízení (serveru) bohatě stačí jedno jediné číslo. Ovšem na straně klienta může být více komunikujících aplikací stejného typu – například s webovým serverem může komunikovat víc oken webového prohlížeče (vlastně i víc webových prohlížečů), navíc v každém okně můžeme mít několik záložek se zobrazenými stránkami od různých webových serverů.

Kdybychom i na straně klienta chtěli mít jen jedno jediné číslo, nebylo by možné nijak určit, kterému procesu/oknu/záložce konkrétně je určena zrovna ta webová stránka, která právě došla od některého webového serveru.



Obrázek 4.8: Význam portů na transportní vrstvě

Na obrázku 4.8 je naznačen význam portů pro rozlišení jednotlivých navázaných spojení, kterých může být pro daný protokol v rámci běžného počítače více než jen jedno. Pokud se jedná o komunikaci s webovým serverem (což může být například server Apache), na straně serveru běží služba Apache komunikující na TCP portu 80 (jeden z *dobře známých portů*), kdežto na straně počítače máme více oken či záložek webového prohlížeče, pro něž potřebujeme různá čísla portů, a to z rozsahu *dynamických (soukromých) portů*.

Další informace:
Oficiální seznam dobře známých a registrovaných čísel portů je na
http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

4.6.2 Protokoly transportní vrstvy

Na transportní vrstvě pracují především protokoly TCP a UDP, ale pro určité konkrétní typy úloh zde najdeme i další.

TCP (Transmission Control Protocol) poskytuje potvrzovanou službu se spojením, garantuje řazení, tedy doručení více úseků dat ve správném pořadí. Používá se pouze pro spojení typu point-to-point, komunikuje vždy jen jedno koncové zařízení s jiným koncovým zařízením.

UDP (User Datagram Protocol) poskytuje nepotvrzovanou službu bez spojení, tedy datagramovou službu. Může se používat i pro spojení typu point-to-multipoint, tedy cílem může být i skupina koncových zařízení.

Z dalších protokolů najdeme na transportní vrstvě například *SCTP* (Stream Control Transmission Protocol), který je podobný TCP, ale dovoluje navázat více nezávislých paralelních spojení (streamů), což snižuje pravděpodobnost ztráty PDU po cestě – každý stream může jít jinou cestou, a pokud má zařízení víc IP adres, lze mezi nimi přecházet. Tento protokol je implementován v jádru mnoha unixových systémů (FreeBSD, Linux, atd.), pro Windows existují implementace třetích stran.

Transportní protokol *RTCP* slouží k rezervaci zdrojů převážně v komunikaci vyžadující upřednostňování při přidělování zdrojů (řízení kvality služby), setkáváme se s ním v některých VoIP technologiích.

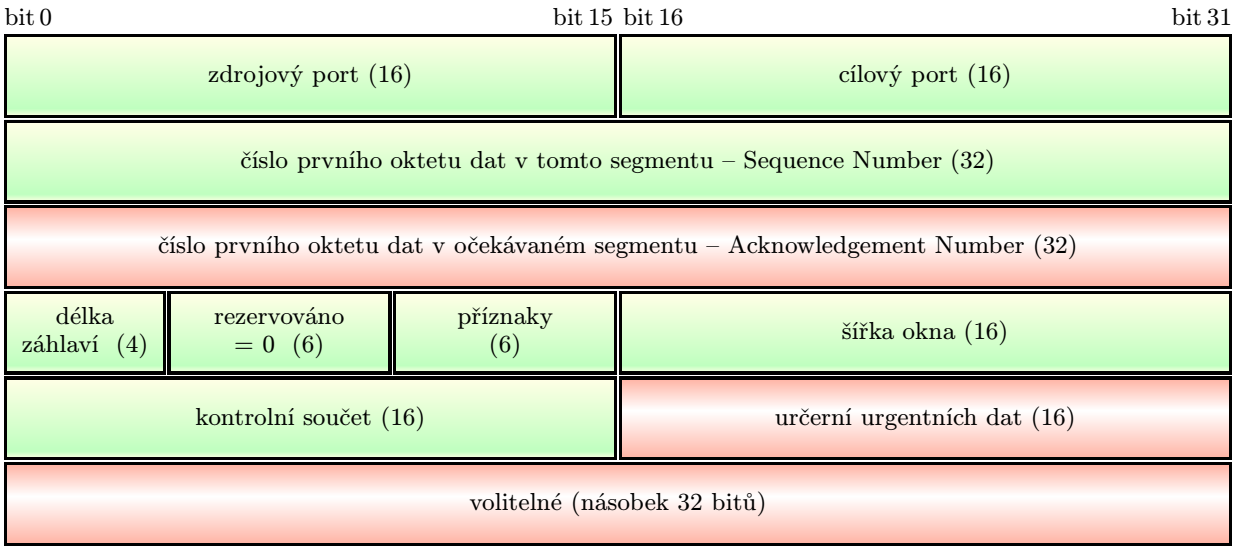
Protokol *DCCP* poskytuje službu se spojením (jako TCP), ale negarantuje doručení ve správném pořadí ani nepotvrzuje (jako UDP), jeho hlavním účelem je zajistit rychlé dodání dat. Za tím účelem poskytuje mechanismus řízení zahlcení (dokáže adaptivně měnit přenosové cesty, aby se vyhnuly přetíženým místům). Používají ho například internetová rádia, některé technologie pro internetovou telefonii, on-line videa.

PDU na transportní vrstvě se většinou nazývají *segmenty*, třebaže v případě protokolů poskytujících datagramovou službu (třeba UDP) se setkáváme také s názvem *datagram*.

4.6.3 Protokol TCP

Úkoly protokolu TCP (Transmission Control Protocol) jsou:

- navázat spojení s druhou stranou (vytvořit virtuální kanál),
- udržovat spojení, potvrzovat přijaté segmenty, řídit datový tok,
- převzít SDU z nadřízené vrstvy (obvykle od některého aplikačního protokolu, třeba HTTP) a takto zpracovat:
  - zkontroluje délku tohoto bloku dat, a pokud je větší než povolená délka segmentu, podle potřeby rozdělí na menší části – *segmentuje*,
  - ke každé z částí vzniklých v předchozím bodu přidá TCP záhlaví, přičemž určí čísla portů pro obě komunikující strany, tak vzniknou TCP segmenty,
  - TCP segmenty postupně předá podřízené vrstvě (obvykle protokolu IPv4 nebo IPv6),
- na konci komunikace ukončit spojení.



Obrázek 4.9: Záhlaví TCP segmentu

Na obrázku 4.9 vidíme záhlaví TCP segmentu. Jednotlivá pole mají tento význam:

- *Zdrojový a cílový port* (Source Port, Destination Port, každý 16 bitů) – čísla portů na zdrojovém a cílovém zařízení.
- *Číslo prvního oktetu v segmentu* (*Sequence Number*, 32 bitů) – posloupnost dat z vyšší vrstvy může být rozdělena do několika segmentů, zde je číslo oktetu z původní posloupnosti dat, kterým začíná úsek zapouzdřený v tomto konkrétním segmentu (oktety jsou číslovány od 0).

- *Číslo prvního oktetu v očekávaném segmentu* (*Acknowledgement Number*, 32 bitů) – číslo oktetu začátku části původních dat, který očekáváme (může náležet buď následujícímu segmentu v pořadí nebo takovému předchozímu segmentu, který nebyl řádně doručen). Toto pole je v každém segmentu kromě prvního segmentu při navazování spojení.
- *Délka záhlaví* (Header Length/Data Offset, 4 bity) – délka celého záhlaví v násobcích 32 bitů (takže „počet řádků“ tabulky podle obrázku 4.9).
- *Příznaky* (Flags, Control Bits/Řídící bity, funkce řízení, 6 bitů) – pole bitů (Wireshark je sdružuje s předchozím polem rezervovaných bitů), z nichž každý má svůj význam, nás budou zajímat zejména tyto příznaky:
  - URG (Urgent) – příznak urgentních dat (v segmentu jsou data, která mají v cíli při zpracování přednost), pole *Určení urgentních dat* má být bráno v úvahu,
  - ACK (Acknowledgement, potvrzení) – tento segment je (kromě jiného) potvrzením předchozího segmentu, pole *Acknowledgement Number* má být bráno v úvahu,
  - SYN (Synchronize bit) – synchronizační bit, používá se při navazování spojení,
  - FIN (Finish bit) – používá se při ukončování spojení.
- *Šířka okna* (Window Size, 16 bitů) – určuje, kolik oktetů maximálně chce odesílatel tohoto segmentu přijmout od svého protějšku bez potvrzování (tj. po odeslání tolika oktetů odešle potvrzující segment).
- *Kontrolní součet* (Checksum, 16 bitů) – počítá se přes celý segment včetně záhlaví plus *pseudozáhlaví*; pseudozáhlaví se vytváří jen pro tento účel (na zdrojovém i cílovém zařízení) a nepřenáší se, obsahuje nejdůležitější údaje z PDU, do které je segment zapouzdřen – většinou protokolu IP (IP adresy zdroje a cíle, protokol – 6 pro TCP, a délku TCP segmentu).
- *Určení urgentních dat* (Urgent Pointer, 16 bitů) – pokud jsou v segmentu také urgentní data a tedy je nastaven bit URG, zde je číslo oktetu, kterým *končí* urgentní data.
- *Volitelné* (Options, násobek 32 bitů) – volitelné možnosti, které si zařízení navzájem potřebují předat (většinou na začátku spojení nebo když je třeba v průběhu spojení pozměnit jeho parametry), například maximální velikost segmentu, časová razítka (kvůli synchronizaci) nebo parametry pro potvrzování.

Protokol TCP předepisuje potvrzování doručení, k čemuž nám slouží pole *Sequence Number* a *Acknowledgement Number*. Tato čísla předepisují určitou návaznost mezi tím, co jeden odesílá, a tím, co druhý přijímá, a stejně to funguje i v opačném směru (takže pozor – *Sequence number* jdoucí v jednom směru nemá nic společného se *Sequence number* v opačném směru, obě strany mohou posílat data).

Šířka okna určuje, po jakém kvantu dat je třeba odeslat potvrzení, a obvykle zabírá velikost několika segmentů – to znamená, že se nepotvrzuje každý segment zvlášť, ale odešle se potvrzující segment až po několika doručených segmentech. Tato hodnota obvykle odpovídá velikosti bufferu (vyrovnávací paměti), ale v případě, že se po cestě ztrácí hodně segmentů, bývá nižší.

Poznámka:

Komunikace obvykle nebývá zcela symetrická (tj. většinou jedna strana posílá „krátké“ žádosti a druhá strana „dlouhé“ odpovědi), takže v jednom směru typicky roste *Sequence number* rychleji než v druhém.

Jak tedy potvrzování probíhá? Použití *Sequence Number* je zřejmé – pokud posílám data, v tomto poli určím, kde konkrétně je posílaná část lokalizovaná v původním streamu dat. Pole *Acknowledge Number* určuje, kterou část dat očekávám v opačném směru.

Jaká tedy má být posloupnost hodnot *Sequence Number* v posílaných (resp. přijímaných) segmentech? Když vezmeme hodnotu *Sequence Number* z jednoho segmentu a přičteme délku dat v tomto segmentu zapouzdřených (tj. délka segmentu mínus velikost záhlaví, obojí v oktetech), získáme hodnotu *Sequence Number* pro následující segment. Délku segmentu zjistíme v IP záhlaví. Wireshark hodnotu pro následující segment taky počítá a zobrazuje ji v hranatých závorkách jako *Next Sequence Number*.

Cílové zařízení přijme tolik segmentů, kolik se vejde do šířky okna, a zkontroluje hodnoty *Sequence Number*, jestli jdou ve správné posloupnosti. Pokud jde všechno tak jak má (žádné segmenty se neztrácejí), odešle druhé straně potvrzující segment, kde je v poli *Acknowledgement Number* číslo vypočtené podle předchozího postupu z posledního získaného segmentu (vezme *Sequence Number* a přičte délku zapouzdřených dat). To odpovídá následujícímu segmentu, který má být přijat.

Jak zjistíme, že se některý segment cestou ztratil? V řadě doručených (zatím nepotvrzených) segmentů provedeme výše naznačený výpočet. Pokud při výpočtu z jednoho segmentu nesedí výsledek na *Sequence Number* následujícího v pořadí (v segmentu je větší číslo než nám vyšlo), pak to znamená, že na tom místě chybí minimálně jeden segment. Proto odešleme potvrzující segment s hodnotou *Acknowledgement Number* takovou, která nám vyšla pro chybějící segment.

Poznámka:

Wireshark ve výchozím nastavení zobrazuje pouze *relativní* Sequence a Acknowledge Number (menší než ve skutečnosti, aby pro daný stream začínalo číslem 0) – účelem je zjednodušit optické srovnávání těchto hodnot. Pokud nám to vadí, můžeme si nastavit zobrazování skutečných hodnot těchto čísel.

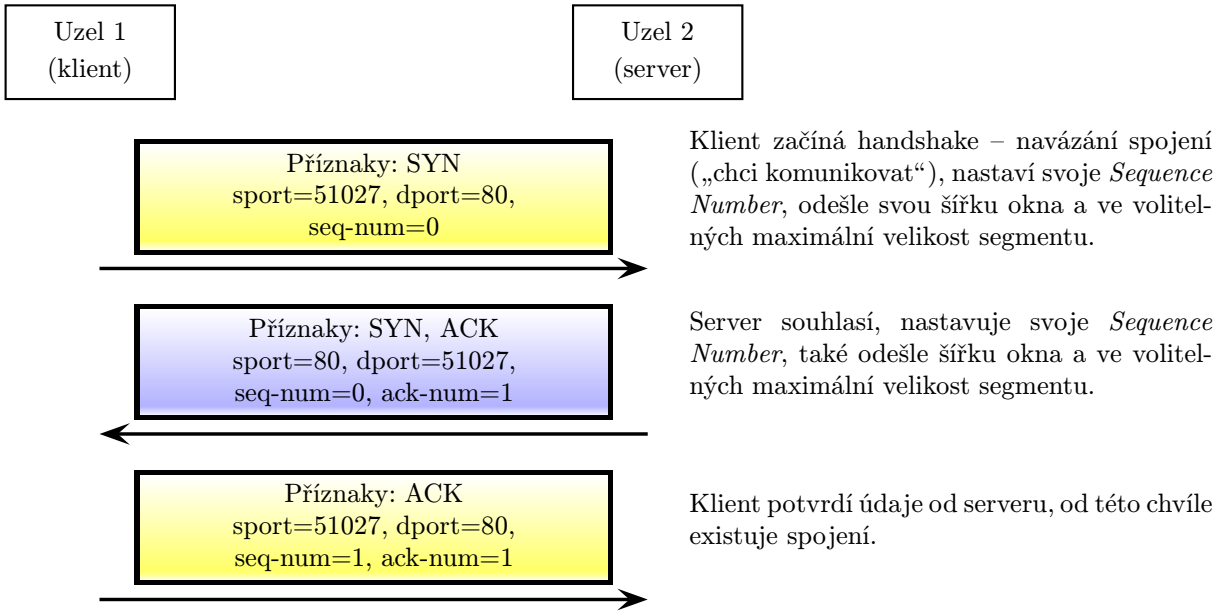
4.6.4 Průběh TCP spojení

TCP spojení musí být nejdřív navázáno, čemuž se říká *Three-way Handshake* (třícestné zahájení). Nazývá se tak proto, že se během zahájení spojení postupně posílají tři segmenty s dojednáváním parametrů spojení (bez dat).

Postup navázání spojení pro komunikaci s webovým serverem je znázorněn na obrázku 4.10:

1. První zařízení (klient) odešle první segment ve významu „Chci navázat spojení“. Tento segment má nastaven příznak SYN a obsahuje prvotní parametry v poli *Volitelné*, například časové razítko kvůli synchronizaci a maximální velikost segmentu, který toto zařízení může přijmout.
2. Druhé zařízení odešle v odpověď svůj první segment (proto má taky nastaven příznak SYN), který je tedy odpovědí, a tedy má nastaven taky příznak ACK. V poli *Volitelné* má podobný typ informací jako byly v předchozím paketu.
3. Druhý segment potvrzoval přijetí toho prvního, tedy je ještě nutné potvrdit přijetí druhého segmentu. Třetí segment má právě tuto funkci. Už není inicializační a neobsahuje synchronizační informace, takže má nastaven pouze příznak ACK. Pole *Volitelné* obvykle není využito.

Čísla portů jsou zřejmá – server používá číslo portu 80, protože se jedná o WWW server, na straně klienta vidíme dynamický (soukromý) port.



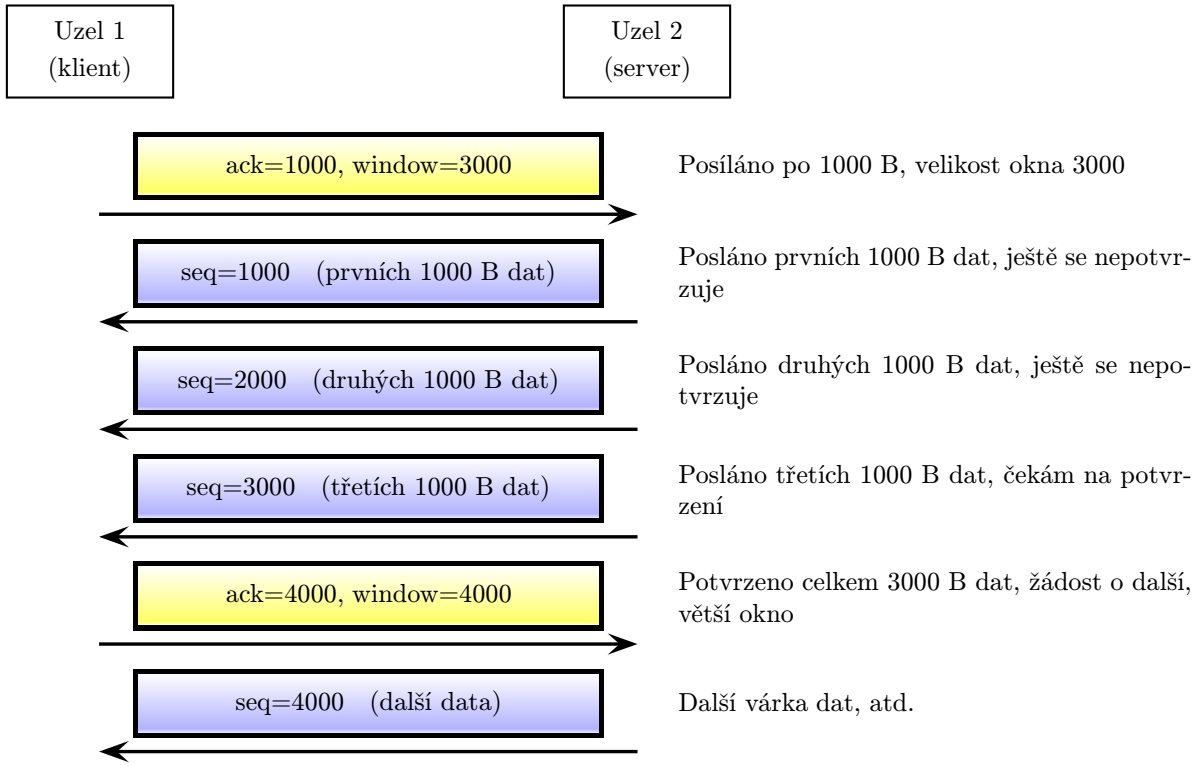
Obrázek 4.10: TCP handshake – navázání spojení s webovým serverem

Poznámka:
V prvních dvou segmentech navolí odesílatel svou výchozí hodnotu pro *Sequence Number* (Wireshark místo toho zobrazí 0 jako relativní číslo), v třetím segmentu je *Sequence Number* o 1 vyšší než v prvním segmentu. *Acknowledgement Number* se používá až od druhého segmentu (v prvním není co potvrzovat) a v druhém a třetím segmentu se použije o 1 vyšší než je *Sequence Number* v předchozím (obdrženém) segmentu. Po navázání spojení (tedy od čtvrtého segmentu dále) se již tato čísla používají tak, jak bylo výše popsáno.

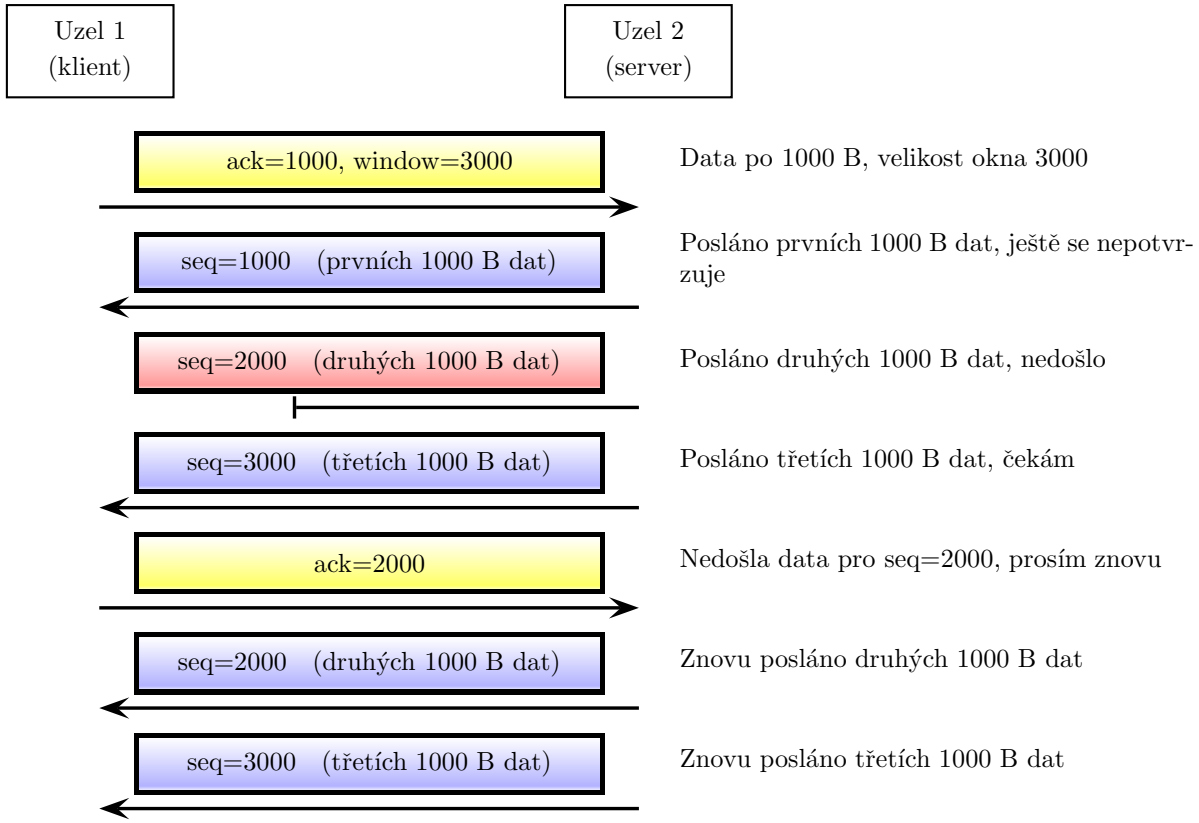
Pokud nedochází ke ztrátám segmentů, posílá se potvrzení (acknowledgement) vždy po tolika segmentech, kolik se jich vejde do šířky okna. Na obrázku 4.11 vidíme průběh takové komunikace pro případ, že velikost segmentu je nastavena na 1000 a velikost okna na 3000. V potvrzujícím segmentu je *Acknowledgement Number* nastaven na 4000, protože žádáme o poslání segmentu s tímto *Sequence Number* (tím dáváme na vědomí, že všechny předchozí byly doručeny).

Na obrázku 4.12 je znázorněna situace, kdy došlo ke ztrátě segmentu. Příjemce dat podle *Sequence Number* v záhlaví segmentů zjistil, že chybí segment pro *Sequence Number* 2000, tedy do potvrzujícího segmentu zadá jako *Acknowledgement Number* právě toto číslo. Pokud za ztraceným segmentem byly další segmenty již doručeny, i tak budou doručovány znovu.

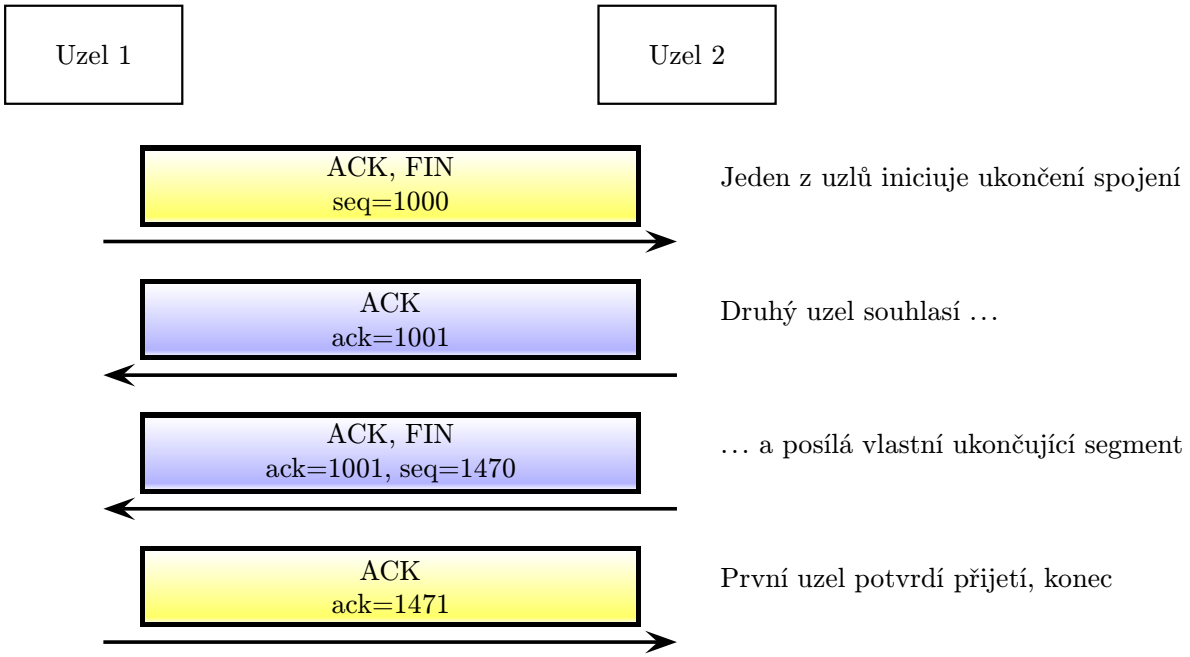
Poznámka:
Důležité je taky to, kdy konkrétně začne příjemce dat zjišťovat, zda má všechny segmenty patřící do probíhajícího okna. Je stanovena maximální doba čekání (timeout) na doručení segmentů, a pokud tato doba uplyne, jsou dosud nedoručené segmenty považovány za ztracené.



Obrázek 4.11: Běžná komunikace v rámci TCP spojení



Obrázek 4.12: Komunikace v rámci TCP spojení, jeden segment nedorazil



Obrázek 4.13: Ukončení TCP spojení

Ukončení spojení může být provedeno kterýmkoliv z komunikujících zařízení, a proběhne pomocí čtyř segmentů, jak je naznačeno na obrázku 4.13. Nejdřív ukončení oznámí jedna strana (je nastaven příznak FIN), pak druhá potvrdí a pošle vlastní ukončující segment (taky s příznakem FIN), následuje potvrzení přijetí. Oba potvrzující segmenty mají nastaven pouze příznak ACK.

4.6.5 Protokol UDP

Jak bylo napsáno výše, protokol UDP (User Datagram Protocol) poskytuje nepotvrzovanou službu bez navázání spojení, tedy jednoduchou datagramovou službu. Jeho úkoly jsou tedy mnohem jednodušší než u protokolu TCP – jednoduše převezme SDU z nadřízené vrstvy, přidá záhlaví a vytvořený segment (datagram) předá podřízené vrstvě. Nenavazuje se spojení, nepotvrzuje se, neřeší se velikost okna, nepočítají se čísla pro chybějící segmenty (vlastně se ani žádná taková čísla nepoužívají).

Proto je záhlaví UDP segmentu mnohem jednodušší, část polí tam prostě nepotřebujeme. Na obrázku 4.14 vidíme záhlaví UDP segmentu – první řádek je stejný jako u TCP, z druhého řádku je stejný kontrolní součet, „navíc“ máme délku celého segmentu (včetně záhlaví) – to u TCP není, a naopak všechna zbylá TCP pole chybí.

Obrázek 4.14: Záhlaví UDP segmentu

Protokol UDP se používá tehdy, když

- je zbytečné navazovat spojení (nebo to z nějakého důvodu není možné či žádoucí),
- posíláme jen málo dat, která není nutné segmentovat do více segmentů,
- chceme, aby byl přenos rychlý a moc nezahlcoval linku,
- chceme poslat data na multicast nebo broadcast adresu (TCP podporuje pouze unicast přenosy).

	TCP	**UDP**
Spojení	navazuje	nenavazuje
Potvrzování	ano	ne
Spolehlivost	ano	ne
Segmentace do více segmentů	ano	ne
Rychlost	pomalý	rychlý
Typ cíle	unicast	unicast, multicast, broadcast

Tabulka 4.5: Srovnání

Kapitola 5

Aplikační protokoly

Rychlý náhled: V této kapitole se zaměříme na vrstvu L7, tedy aplikační. Budeme se zabývat službami poskytovanými na této vrstvě, a protokoly, které určují technologii implementující tyto služby (tedy aplikačními protokoly).

Klíčová slova: DNS, doména, jmenná adresa, zóna, DNS server, tabulka hostitelů, DNS paket, WHOIS, URL, URI, URN, HTTP, HTTPS, Wireshark, SMTP, IMAP, MTA, MDA, MUA, MIME, FTP, SMB, DHCP, Telnet, SSH, SNMP

Cíle studia: Po prostudování této kapitoly získáte přehled v protokolech aplikační vrstvy. Budete vědět, jak funguje DNS překlad, jak klientské zařízení komunikuje s webovým serverm, jak funguje elektronická pošta, jak se přenášejí soubory, jak komunikují počítače v jednoduché síti typu peer-to-peer, jak se přidělují adresy, jak se komunikuje se vzdáleným zařízením.

5.1 DNS aneb překlad adres

5.1.1 Domény a jmenné adresy

Víme, že počítače a sítě jsou adresovány IP adresami, a bez těchto adres (nebo jejich ekvivalentu v konkurenčních technologiích) nemohou na síti existovat. Jenže IP adresy (zvláště pro IPv6) jsou pro člověka těžko zapamatovatelné a lidé by se asi vzbouřili, kdybychom je nutili tyto adresy zapisovat třeba do adresního řádku webového prohlížeče.

Lidé používají *jmenné (doménové) adresy* zařízení, které se lépe pamatují a dobře se zapisují. Například můžeme do adresního řádku webového prohlížeče zapsat adresu `www.google.com` a nemusíme se mořit s adresou `216.58.214.196` nebo `2a00:1450:400d:802::2004`.

Jenže síťová zařízení takové adresy rozhodně používat nebudou. Takže potřebujeme mechanismus, který hlídá přiřazení mezi konkrétní jmennou adresou zařízení a jeho IP adresou, a to je právě *DNS (Domain Name System)*.

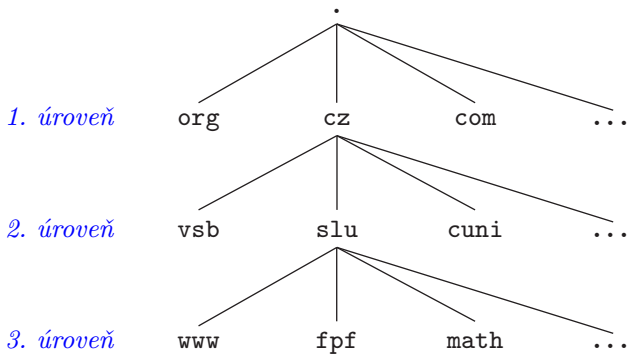
Definice (Doména, doménové jméno)

Doména je síť nebo skupina sítí pod společnou správou a sdílející společné (doménové) jméno. *Doménové jméno* musí splňovat tyto podmínky:

- může obsahovat písmena anglické abecedy, číslice a pomlčku, přičemž pomlčka nesmí být na začátku ani na konci,
- maximální délka jednoho jména je 63 znaků,
- maximální délka zřetězení doménových jmen je 255 znaků,
- v rámci domény jsou názvy subdomén jednoznačné.

Doména může mít přiřazeno více jmen. Jedno z nich je *kanonické jméno* (hlavní), ostatní jména nazýváme *aliasy*.

O adresním prostoru protokolu IP jsme si říkali, že je hierarchický, v adresách ve směru zleva (zařízení v téže síti mají levou část adresy shodnou), přičemž síť je členěna na podsítě a jejich vztah můžeme zakreslit pomocí stromu. Doménový adresní prostor je taky *hierarchický*, ale ve směru zprava – zařízení v téže doméně mají pravou část jmenné adresy shodnou, domény jsou členěny hierarchicky. Jejich vztah můžeme taky zakreslit pomocí stromu.



Obrázek 5.1: Ukázka stromu doménových názvů

Na obrázku 5.1 je ukázka takového stromu (samozřejmě hodně osekaná). Kořen stromu slučuje všechny podstromy do jediného stromu, směrem dolů postupujeme k podřízeným doménám. Listy stromu (zcela dole) jsou konkrétní zařízení, obvykle servery (jejich hostname).

Nejvyšší úrovně doménového stromu mají i své názvy:

- V kořeni stromu je *kořenová doména* (root domain).
- Domény typu *Top-Level Domain* (TLD) jsou v první úrovni stromu domén (například `.cz`).
- Dále hovoříme o doménách druhé (SLD – Second Level Domain), třetí úrovně atd.

Doménu nebo skupinu domén spravovuje konkrétní subjekt – *správce domény*. Podle našeho obrázku je například doména `.slu` včetně subdomén spravována Slezkou univerzitou, doména `.cz` je spravována hlavním registrátorem domén pro Českou republiku, organizací CZ.NIC.

TLD domény jsou několika typů:

- *národní domény* (ccTLD – Country Code TLD, dvoupísmenné) jsou pojmenovány podle mezinárodních kódů států (například `cz`, `uk` apod.), patří sem i doména `eu`, třebaže nesplňuje podmínku „národnosti“,

- *generické domény* (gTLD, minimálně třípísmenné) pojmenované podle svého zaměření:
  - `com` – komerční organizace,
  - `edu` – vzdělávací instituce,
  - `gov` – státní správa v USA,
  - `mil` – armáda USA,
  - `net` – organizace související se správou a standardizací sítí, například `ieee.org`,
  - `org` – ostatní organizace,
  - nové generické domény, například `audio`, `cafe`, `download`, `help`, `chat`, `training`, atd. (stovky).

✎ *FQDN* (Fully Qualified Domain Name) je plné jméno zařízení, úplná jmenná adresa. Tuto adresu sestavíme tak, že ve stromě jdeme od listu (dotyčného serveru) směrem nahoru ke kořeni a domény na cestě oddělujeme tečkou, například server `www` (tj. webový server) podle obrázku 5.1 má FQDN `www.slu.cz`. Maximální délka FQDN je 255 znaků.

5.1.2 Zóny a DNS servery

✎ Jednu nebo více domén sdružujeme do společné *zóny* (typicky to bývá část podstromu domén). Pro každou zónu je stanovena konkrétní *autorita*, tedy zodpovědná organizace. Členění na zóny (coby skupiny domén) souvisí s odpovědností a technickou správou, zóny se obvykle nepřekrývají, až na jejich hranice (mezi zónami musí existovat určitá komunikace). Například celý podstrom `slu` tvoří zónu ve správě Slezské univerzity, kdežto nadřízená doména `cz` patří do zóny spravované organizací CZ.NIC.

✎ Správu zóny zajišťujeme na *doménovém (jmenném) serveru – DNS serveru* (name serveru). Hlavními úlohami DNS serveru jsou:

- vést tabulku adres (tabulku hostitelů – host table), je uložena v *zónovém souboru*,
- odpovídat podle této tabulky na DNS dotazy, tedy překládat jmennou adresu na IP adresu.

V doméně (v zóně) musíme mít jeden *primární DNS server* a dále tam můžeme mít pomocné servery – *sekundární DNS servery* a *caching-only servery*. Ve vztahu k zónovému souboru:

- Primární DNS server vede a garantuje zónový soubor s tabulkou hostitelů, změny v tabulce provádíme vždy právě na tomto severu a záznamy na tomto serveru jsou vždy důvěryhodné, *autoritativní*.
- Sekundární DNS servery si v pravidelných intervalech kopírují zónový soubor z primárního DNS serveru (synchronizují, tomu se říká *zone transfer*), jejich účelem je rozložení zátěže, aby primární server nebyl příliš zatěžován. Jejich odpověď je také *autoritativní*.
- Caching-only servery neobsahují celý zónový soubor. Když takový server obdrží dotaz na adresu, „zeptá se“ primárního nebo sekundárního serveru, ale odpověď si na určitou krátkou dobu uloží (do cache – dočasné paměti) a pokud je v krátké době tázán na tutéž adresu, použije údaj z cache (takže se nemusí dotazovat dále). Odpověď caching-only serveru není autoritativní, ale většinou dostačuje.

✂ DNS server má ve svém zónovém souboru informace o uzlech nacházejících se v jeho zóně a dále zná DNS server z nadřízené zóny (tedy část adresy směrem od sebe vpravo). Dotaz na adresu obvykle znamená, že je zadána adresa FQDN a tazatel chce IP adresu. Pokud je DNS

server tázán na IP adresu zařízení, které patří do jeho zóny, může hned odpovědět. Pokud je v dotazu jméno, které v databázi nemá (protože není z jeho zóny), probíhá vyhodnocení takto:

- Pokud je dotazovaný název z podřízené zóny (pravá část adresy je stejná), najde jmenný server ve směru zprava doleva první doménu v FQDN, která není jeho. Má odkazy na všechny domény na hranici s podřízenou zónou (například v zónovém souboru pro `cz` je odkaz na DNS server domény `slu`), tedy předá dotaz DNS serveru této podřízené domény. V podřízené doméně je dotaz vyřízen nebo opět předán níže. Rekurzívně směrem dolů je nalezena odpověď, která je pak stejnou cestou distribuována zpět tazateli.
- Pokud dotazovaný název není z vlastní ani podřízené zóny, předá DNS server dotaz DNS serveru z nadřízené zóny. Ten dotaz buď vyřídí nebo předá některému podřízenému nebo nadřízenému uzlu. Po vyřízení je odpověď opět distribuována stejnou cestou k tazateli.

Příklad

Pokud se některé zařízení z domény `fpf.slu.cz` dotáže na IP adresu serveru `www.fpf.slu.cz`, DNS server domény `fpf.slu.cz` hned odpoví, protože tento webový server patří do jeho zóny.

Pokud se některé zařízení z domény `fpf.slu.cz` dotáže na IP adresu serveru `www.cuni.cz`, nejdřív se dotazem zabývá DNS server domény `fpf.slu.cz`. Adresa je z jiného podstromu, tedy dotaz přepošle do domény `slu.cz`. To je podobný případ, proto je dotaz přeposlán DNS serveru domény `cz`. Tento server již pozná, že dotazovaná adresa patří do podřízené domény `cuni.cz` (protože pravá část adresy – `cz` – je shodná), v zónovém souboru najde kontakt na DNS server z podřízené domény, který již najde odpověď – IP adresu webového serveru `www.cuni.cz`, protože ten patří do jeho zóny. Odpověď půjde tazateli zpět stejnou cestou.

DNS resolver (dotazovač) je komponenta, která zajišťuje dotazování a ve své cache paměti si po určitou dobu udržuje výsledky předchozích dotazů. Tuto komponentu najdeme v každém systému, který je připojen k síti používající jmenné názvy, tedy i v obyčejném počítači (včetně počítačů s Windows nebo mobilních zařízení). Z toho vyplývá, že i resolver ve Windows si po určitou dobu pamatuje výsledky proběhlých DNS dotazů, aby je nemusel zbytečně často opakovat.

Nejdůležitější informace, kterou DNS resolver potřebuje, je adresa příslušného DNS serveru. Právě tomuto serveru odesílá všechny své dotazy, které pak mohou být buď přímo zodpovězeny nebo v rekurzi předávány.

Postup (Možnosti vyhodnocení DNS dotazu na koncovém zařízení)

Zařízení v síti, na kterém pracuje pouze DNS resolver, si ve skutečnosti některé údaje taky ukládá, jak bylo naznačeno výše. Koncové zařízení s DNS resolverem při vyhodnocení dotazu postupně zkouší tyto možnosti:

- v souboru `/etc/hosts`, resp. `...\system32\drivers\etc\hosts` je seznam dvojic IP adresa + jmenná adresa, které tam lze „staticky“ uložit,
- odpověď na dotaz je v cache paměti, pokud jsme v nejbližší době pokládali stejný dotaz,
- poslední možnost je dotázat se DNS serveru.

Standardně je odpověď hledána právě v uvedeném pořadí.

5.1.3 Tabulka hostitelů

Klíčovým prvkem pro DNS překlad je *tabulka hostitelů* (host table) v zónovém souboru. Můžeme si ji představit jako tabulku, ve které má každá adresa svůj řádek, a na tom řádku máme tyto údaje (ve sloupcích):

- jmenná adresa,
- IP adresa,
- typ záznamu,
- TTL,
- třída (class) – většinou „IN“ jako Internet,
- případně další údaje k záznamu.

Údaj TTL je životnost záznamu v sekundách pro případ, že je tento záznam stažen do cache jiného zařízení (ať už koncového zařízení nebo caching-only serveru).

Z toho, že mezi údaji je i položka „typ záznamu“ (třetí odrážka předchozího výčtu), plyne, že existují různé typy DNS záznamů. Nejpoužívanější typy záznamů jsou:

- typ A – pro překlad jmenné adresy na IPv4 adresu,
- typ AAAA – pro překlad jmenné adresy na IPv6 adresu (čtyři „A“) jsou tam proto, že IPv6 adresy jsou čtyřikrát delší než IPv4 adresy),
- typ CNAME (Cannonical Name) – pro překlad aliasu na kanonické jméno dotyčného serveru,
- typ NS (Name Server) – v tomto záznamu se dozvíme IP adresu autoritativního DNS serveru pro danou doménu (okolní domény nebo takové, se kterými se v naší doméně hodně komunikuje),
- typ MX (Mail eXchanger) – adresa e-mail serveru pro danou doménu,
- typ SOA (Start of Authority) – administrativní informace o doméně,
- typ PTR – pro reverzní (obrácený překlad, kdy známe IP adresu, ale potřebujeme jmennou adresu, atd.

Většinou potřebujeme právě záznamy typu A nebo AAAA, kdy známe jmennou adresu a potřebujeme buď IPv4 adresu nebo IPv6 adresu.

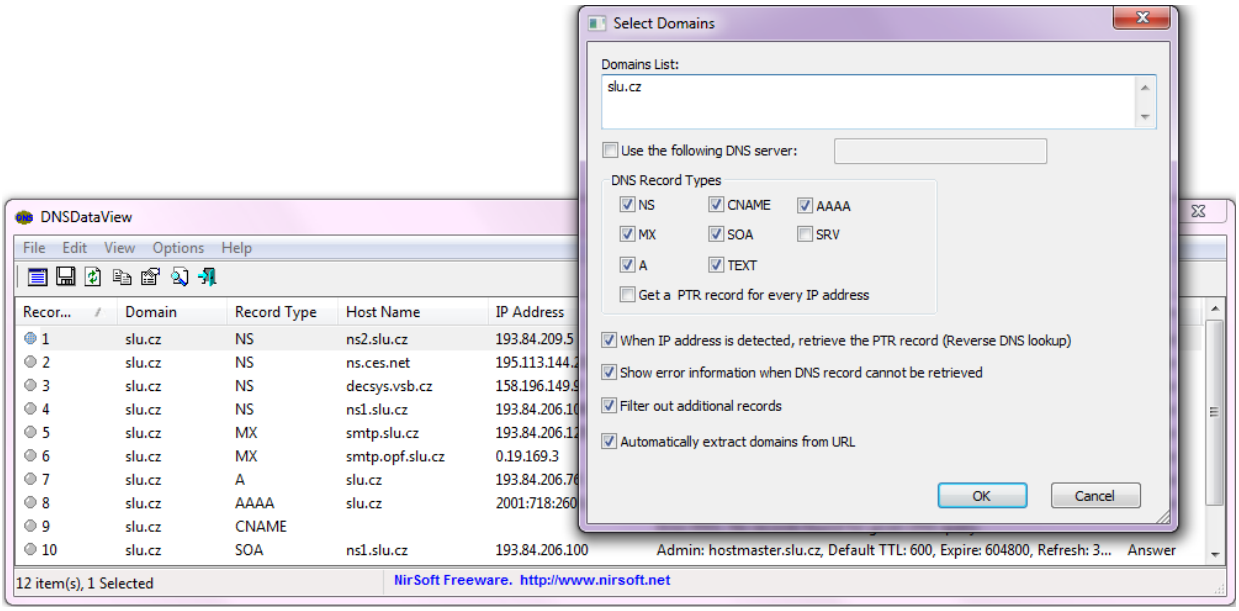
Příklad

Pro přístup k DNS záznamům můžeme použít příkaz `nslookup` nebo v UNIXových systémech lépe příkaz `dig`, ale zajímavou alternativou s grafickým rozhraním je aplikace *DNS Data View*[1] od společnosti Nirsoft. Okno s výstupem pro doménu `slu.cz` vidíme na obrázku 5.2.

5.1.4 Protokol DNS a DNS paket

DNS není jen služba, stejně se nazývá i aplikační protokol, který tuto službu poskytuje. Tento protokol stanovuje, jak má vypadat DNS paket, jak se s ním má zacházet, jak funguje celý systém DNS včetně práce se zónovými soubory, jejich transferu mezi DNS servery (zone transfer) a vzájemné komunikace mezi DNS servery.

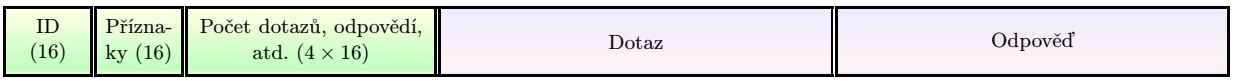
[1] Ke stažení na http://www.nirsoft.net/utils/dns_records_viewer.html



Obrázek 5.2: Program Nirsoft DNS Data View

Protokol DNS definuje komunikaci typu klient-server, tedy rozlišujeme *dotaz* (query question) a *odpověď* (query response). Pro dotaz i odpověď se používá tentýž *formát DNS paketu* (vlastně i pro další činnosti související s DNS). V záhlaví je

- *identifikátor* (podobný jako se používá u IP) sloužící ke spárování dotazu a odpovědi,
- *pole příznaků*, z nichž je nejdůležitější ten, který určuje, zda jde o paket s dotazem nebo s odpovědí, nebo například v případě odpovědi zda je dotazovaný server autoritativní,
- několik dvouoktetových číselných hodnot (počet dotazů, odpovědí, autoritativních odpovědí, dodatečných informací v tomto paketu),
- dotaz(-y) – pokud se jedná o dotaz na překlad, pak tu najdeme jmennou adresu, která má být přeložena, typ záznamu, podle kterého chceme přeložit (A nebo AAAA) a třídu (obvykle IN jako Internet),
- odpověď(-i) – plné znění záznamu nalezeného v zónovém souboru (jmenná adresa, IP adresa, typ, třída, TTL).



Obrázek 5.3: DNS paket

Do odpovědi patří i dotaz, na který se odpovídá, vytvoření paketu pro odpověď probíhá takto:

- použijeme stejný identifikátor jako v dotazu,
- v poli příznaků nastavíme příznak pro odpověď a pak ještě pár dalších podle okolností,
- pole pro počet dotazů přejmeme, do pole pro počet odpovědí vložíme počet nalezených záznamů ze zónového souboru,
- pole pro dotaz zkopírujeme z dotazu,
- pole pro odpověď naplníme obsahem nalezených záznamů ze zónového souboru.

DNS paket se zapouzdřuje do UDP nebo TCP segmentu, na straně DNS serveru komunikuje *na portu 53* (pro UDP i TCP). Jsou upřednostňovány UDP segmenty, protože pro tento typ komunikace je důležitá rychlost. TCP se používá typicky tehdy, když je DNS paket nutné rozdělit do více segmentů.

Protože na straně klienta může být více různých tazatelů komunikujících s DNS serverem, musí každý takový tazatel (například okno webového prohlížeče, poštovní klient, Skype apod.) mít vlastní dynamické číslo portu, aby bylo možné tyto komunikace rozlišit.

5.1.5 Databáze WHOIS

Jak bylo výše uvedeno, každá doména má svého odpovědného správce. Jak zjistit informace o doméně včetně odpovědnosti? K tomu slouží služba *WHOIS* (z angličtiny Who Is?).

Databáze WHOIS je distribuovaná, tedy rozložená mezi odpovědné organizace. Celá databáze je rozložena mezi jednotlivé RIR poskytovatele (každý si vede svou vlastní databázi pro svou doménu a subdomény), podobně se informace distribuují níže (LIR apod.).

Příklad

Pokud budeme chtít vyhledávat SLD doménu v rámci české TLD domény, hledáme ve WHOIS databázi vedené hlavním českým registrátorem CZ.NIC. Jestliže chceme vyhledávat TLD doménu z Evropy či většiny Asie, hledáme ve WHOIS databázi registrátora RIPE.

Jednodušší je však často využít „obecné služby“ serverů, které nejsou regionálně omezené, jako je například http://www.whois.com/whois/ nebo http://www.whois-search.com/.

Pokud chceme zjistit informace o určité konkrétní doméně, zadáme do příslušného vyhledávače FQDN název této domény (tj. včetně nadřízených domén v pravé části adresy). Volíme spíše obecnější název, protože bychom měli pro danou odpovědnou organizaci vyhledávat její nejvyšší doménu. Například místo `fpf.slu.cz` zadáme `slu.cz`. V odpovědi bychom měli zjistit časové údaje (platnost registrace apod.), kdo je registrátorem, DNS servery, kontaktní informace a další.

5.2 Služba WWW a protokol HTTP

Protokol HTTP (HyperText Transfer Protocol) známe především z adresního řádku webového prohlížeče, ale obecně je tento protokol používán pro přenos strukturovaných informací; přenos webových stránek ve formátu HTML či XML je jen jednou z mnoha možností jeho využití.

5.2.1 Adresace

URI (Uniform Resource Identifier) je sekvence znaků identifikující konkrétní zdroj. Může to být lokátor (stanovuje umístění zdroje) nebo název (bez lokace). Existují tedy dva druhy URI:

- *URL* (Uniform Resource Locator) je URI typu lokátoru,
- *URN* (Uniform Resource Name) je URI typu názvu.

Odlišnost je tedy v tom, zda v identifikátoru máme nebo nemáme zakódováno umístění zdroje. Například telefonní číslo je URI typu název (URN), protože neurčuje konkrétní lokaci.

URL je typ adresy používaný (kromě jiného) právě protokolem HTTP pro určení webového serveru. Je to řetězec začínající názvem protokolu (v tomto případě `http://` následovaným jmennou adresou serveru a podle potřeby určením konkrétní stránky v adresářové struktuře na serveru.

Plná specifikace URI je `protokol://server:port/cesta`, kde

- `protokol` určuje, přes který protokol se komunikuje (http, ftp apod.),
- `server` je adresa serveru, obvykle jmenná (hostname) nebo IP adresa,
- `port` je číslo portu TCP/UDP, přes který se má komunikovat (zadáváme, pokud se má použít jiný než je obvyklé),
- `cesta` určuje cestu k žádanému zdroji v rámci zadaného serveru (v adresářové struktuře).

Některé části jsou nepovinné. Pokud je nezadáme, je zvolena výchozí možnost. Například pokud je zadána adresa webového serveru a nepřipíšeme číslo portu, zvolí se port 80.

Příklad

V následujícím řetězci, který je příkladem URL, máme tři části:

`http://www.firma.cz/dokumenty/kontakty.html`

První část určuje protokol, druhá jmennou adresu serveru (která může být nahrazena jeho IP adresou, ať už ručně nebo překladem přes DNS), třetí cestu k dokumentu ve formátu HTML uloženém na serveru.

Kdybychom jmennou adresu serveru nahradili jeho IP adresou, taky se bude jednat o URL, a bude lokalizovat stejný zdroj.

5.2.2 HTTP zprávy

PDU protokolu HTTP se nazývají zprávy a zapouzdřují se do TCP segmentů (tj. spojová komunikace s potvrzováním). V naprosté většině případů se na straně serveru používá port 80 (můžeme se setkat také s portem 8080), na straně klienta máme opět pro každého tazatele nad aplikační vrstvou samostatné číslo portu z rozsahu pro dynamické porty. Celá komunikace se označuje jako *HTTP relace* (session) a patří do ní všechny zprávy v rámci daného spojení.

HTTP předepisuje komunikaci typu dotaz–odpověď. HTTP zpráva může být tedy buď dotazem (request) nebo odpovědí (response). Dotaz může být proveden jednou z těchto *metod*:

- HTTP GET – nejčastěji používaná metoda. Součástí URL je i specifikace konkrétních dat.
- HTTP POST – používá se pro odesílání vyplněných webových formulářů, posílaná data nedává do URL, ale do těla zprávy.
- Další metody – například PUT přenese data na server, DELETE umožňuje mazat objekty na webovém serveru, HEAD funguje jako GET, ale neposílá data (jenom poskytuje metadata v záhlaví), CONNECT přidává do adresy informaci o číslu portu.

První informace ze záhlaví je použitá metoda, zbytek záhlaví je posloupnost dvojic ve tvaru „položka: hodnota“. Položky takto předávané jsou například typ obsahu, označení serveru, konkrétní adresa objektu na serveru, použité kódování, v položce User-Agent zase najdeme co nejúplnější specifikaci programu, který na straně klienta žádá o webovou stránku.

Následuje tělo HTTP zprávy. Pokud je co přenášet, při odeslání od serveru klientovi tam většinou najdeme HTML kód žádané stránky.

5.2.3 Komunikace podle HTTP

Komunikace podle HTTP probíhá následovně:

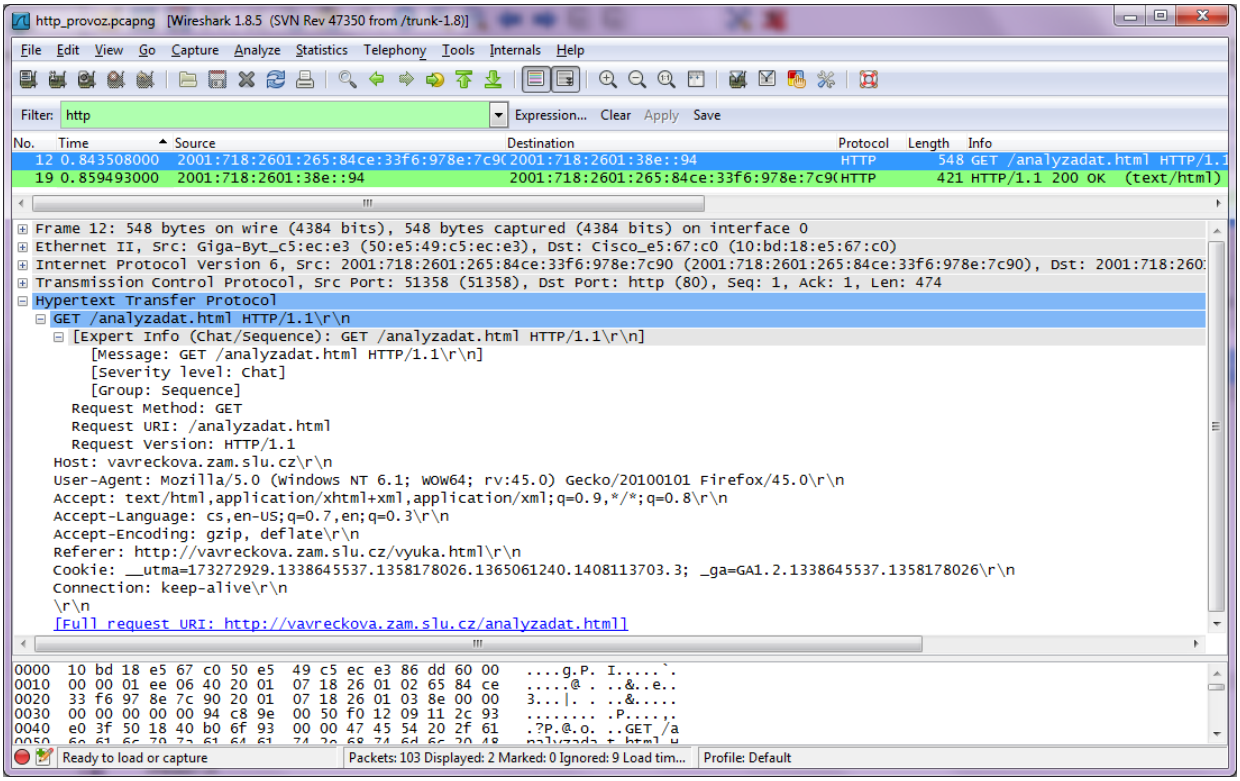
- Proběhne TCP Three-Way Handshake. V poli pro číslo portu na straně serveru je číslo 80, třebaže uvnitř těchto TCP segmentů nemáme zapouzdřenou žádnou HTTP zprávu.
- Následně klient odešle žádost serveru v TCP segmentu, který již obsahuje zapouzdřenou HTTP zprávu se záhlavím typu request.
- Server odpoví TCP segmentem se zapouzdřenou HTTP zprávou se záhlavím typu response.
- Podle potřeby se dotazy a odpovědi opakují.
- Následuje konec HTTP relace a ukončení TCP spojení (bez zapouzdřených HTTP zpráv).

Jestliže klient z předchozí odpovědi serveru zjistil, že na stránce, jejíž text již obdržel, se nachází adresa některého vnořeného objektu k načtení (obrázku, rámu apod.), pošle serveru nový HTTP dotaz, tentokrát na vnořený objekt. To se může opakovat i rekurzívně, také ve vnořeném objektu může být jiný vnořený objekt.

Samozřejmě kromě výše uvedených kroků je třeba udělat kroky „doprovodné“, například protokolem DNS zjistit IP adresu zadaného serveru (a až pak se provádí Three-Way Handshake). Ještě předtím může být třeba pomocí protokolu ARP nebo NDP zjistit MAC adresu brány.

Příklad

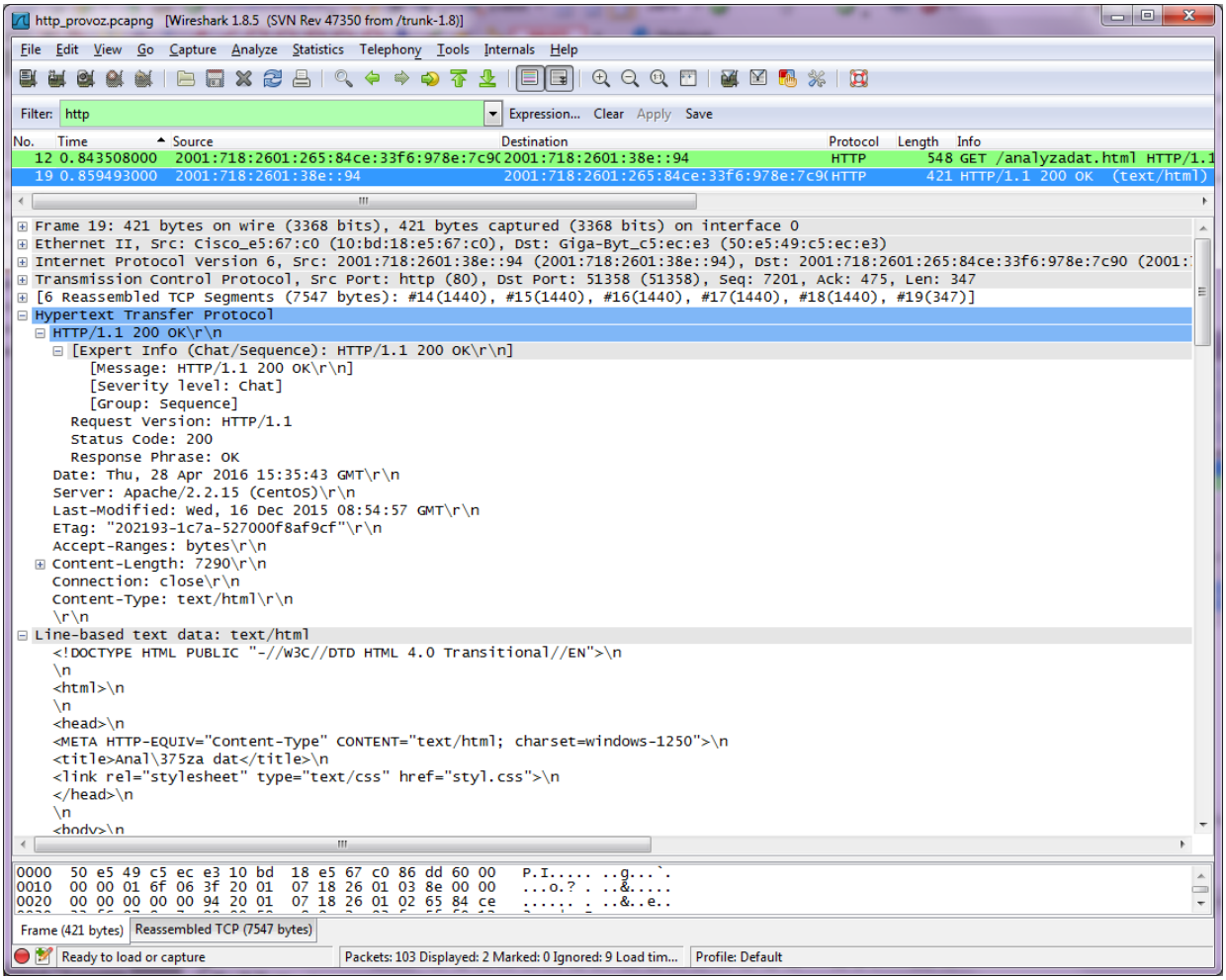
Ukážeme si na konkrétním příkladu (jen HTTP zprávy, žádný „kontext“ v podobě TCP či DNS), jak vypadá vyžádání stránky a její doručení.



Obrázek 5.4: Zpráva HTTP GET

Na obrázku 5.4 je HTTP zpráva s dotazem (žádostí) typu GET. Všimněte si, že za `GET` je vlastně poslední část URL, tedy konkrétní soubor (kdyby byl tento soubor zanořen v některém adresáři, byla by tu celá cesta). Až za údaji o dotazu GET je řádek `Host: ...`, kde je část URL odpovídající doménovému názvu. Jednotlivé záznamy v záhlaví jsou ukončeny dvojicí znaků `\r\n`, což je zařádkování. Na konci celého záhlaví (v případě dotazu to odpovídá i konci celé HTTP zprávy) je toto ukončení zdvojeno, čímž cíl pozná konec záhlaví.

Všimněte si také, že náš prohlížeč poněkud „bonzuje“ – nejen cíl, ale také kdokoliv na cestě se dozví, jaký máme webový prohlížeč a v jakém operačním systému pracujeme, včetně verzí (zde Firefox na 64bitových Windows verze 6.1, což jsou Windows 7).



Obrázek 5.5: Potvrzení zprávy HTTP GET

A co jsme se dozvěděli z odpovědi? Například to, že na serveru běží webový server Apache, a operační systém je CentOS (to je jedna z linuxových distribucí). Záhlaví opět končí dvojitým odřádkováním, ale tentokrát máme i datovou část – `Line-based text data`, kde je nejdřív uveden typ podle MIME (o MIME víc v následující sekci). Jak vidíme, následuje kód webové stránky ve formátu HTML.

Nejznámější webové (HTTP) servery jsou Apache, MS IIS a nginx, většina webových sídel používá Apache. Obvykle jde o službu nebo démona (démon je v UNIXových systémech obdoba

služby) – proces běžící na pozadí a neustále vyhodnocující požadavky, které přicházejí ze sítě.

Podle *Netcraftu* (známá britská analytická společnost) podíl Apache a IIS mírně klesá, podíl nginxu mírně roste. Analýza z března 2016 je na http://news.netcraft.com/archives/2016/04/21/april-2016-web-server-survey.html. U aktivních webových stránek má Apache podíl 49 %, nginx 17 %, IIS 10 %. U „nejrušnějších“ (busiest) stránek má Apache 45 %, nginx 26 % a IIS 11 %.

Poznámka:
Jak vlastně webový server (třeba Apache) pozná, že mu právě byla doručena HTTP zpráva? O to se musí postarat sám démon či služba. Říkáme tomu, že *naslouchá* na portu (obvykle na portu číslo 80), a kdykoliv přijde na tento port požadavek, naslouchající proces je informován.

Dnes často komunikujeme s webovým serverem zabezpečeně. To znamená, že mezi protokoly HTTP a TCP se „nasune“ jeden z protokolů SSL nebo TLS a zajistí šifrování komunikace. Kombinace HTTP a jednoho z těchto zabezpečujících protokolů se označuje *HTTPS*. Hned po navázání TCP spojení se dojednají parametry zabezpečeného připojení a od té chvíle je komunikace šifrovaná.

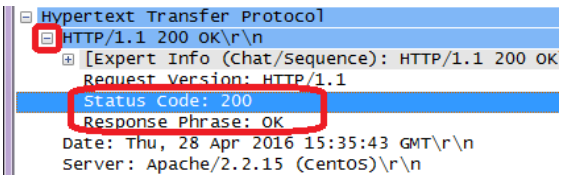
Komunikace podle HTTPS probíhá na jiném portu než 80, na straně serveru se používá port 443. Takže webový server vlastně naslouchá nejen na portu 80, ale také na portu 443 (a případně jiných portech podle konfigurace).

5.2.4 Informační kódy HTTP

Ne vždy se povede dostat webovou stránku ke klientovi tak, jak by to mělo být. Může nastat mnoho různých chyb, na straně klienta, serveru i po cestě. Uživatel na straně klienta pak může (nemusí) být informován webovým prohlížečem, že něco není v pořádku. Kromě chyb je samozřejmě klient informován obecně o stavu plnění požadavku.

Protokolem HTTP je klient informován tříciferným kódem. Kódy začínající určitou číslicí mají tento význam:

- 1xx – informační, server zpracovává požadavek, například kód 100 znamená, že server úspěšně přijal žádost a začal pracovat na jejím řešení,
- 2xx – oznámení o úspěšném zpracování požadavku nebo jeho části, například kód 200 znamená úspěšné dokončení operace,
- 3xx – pro úspěšné zpracování požadavku je třeba provést některou nestandardní operaci (například přesměrování na jinou adresu),
- 4xx – chyba na straně klienta,
- 5xx – chyba na straně serveru.



První a druhý typ kódu jsou informace o průběhu, od třetího typu dále již jde o informace o chybách.

Z chyb na straně klienta je asi nejběžnější chyba 404 (Not Found – požadovaný zdroj nebyl na zadaném serveru nalezen), což většinou znamená, že jsme se překlepli v poslední části URL (název souboru nebo některý adresář na cestě k němu), nebo tento soubor byl na serveru přesunut, přejmenován či smazán.

Chyba 401 je poslána klientovi v případě, že žádá přístup ke zdroji, ke kterému je požadována autentizace (tj. musíme zadat přihlašovací informace a pak bude zdroj zpřístupněn.

Ze serverových chyb se můžeme setkat třeba s chybou 500 (Internal Server Error), která nastává v případě, že proces běžící na straně serveru nereaguje tak, jak by měl (například zamrzl), nebo 503 (Service Unavailable), pokud je server odstaven (například pro účely údržby, asi na něm běží Windows).

Tento kód je součástí záhlaví HTTP zprávy. Všimněte si na obrázku 5.5 na straně 115 (také na obrázku na předchozí straně) řádku `Status Code: 200`, to je přesně ono, server sděluje klientovi, že je vše hotovo a v této zprávě posílá požadovanou stránku (obecně objekt).

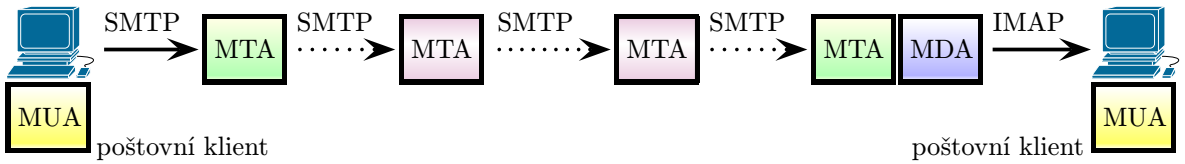
Další informace:

Na stránce https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html je popsáno, jak protokol HTTP funguje, včetně záhlaví s informačními a chybovými kódy.

5.3 Služby elektronické pošty

5.3.1 Infrastruktura

Jak jistě každý ví, svou e-mailovou schránku máme na poštovním (e-mail) serveru. Když odesíláme e-mail, pošleme ho právě tomuto serveru (ten zajistí přeposlání do schránky příjemce), na příchozí e-maily (jejichž jsme adresátem) se díváme do naší schránky. Takže tento server je pro nás poskytovatelem poštovní služby. Celá infrastruktura je však složitější než jen poštovní servery se schránkami, jak vidíme na obrázku 5.6.



Obrázek 5.6: Infrastruktura poštovních serverů

Součástí infrastruktury jsou následující prvky:

- *MTA* (Mail Transfer Agent, agent pro transfer zpráv) – jeho úkolem je přijmout e-mailovou zprávu, zkontrolovat a podle adresy určit, kam ji případně přeposlat. Roli MTA mohou plnit mail servery jako je MS Exchange, postfix, qmail, sendmail, atd.
- *MDA* (Mail Delivery Agent, agent doručení zpráv) – vede poštovní schránky uživatelů, přijímá od MTA zprávy určené do těchto schránek. Funkce MDA bývá obvykle zahrnuta v nástrojích pro MTA.
- *MUA* (Mail User Agent) – aplikace, která komunikuje přímo s uživatelem (obvykle lokálně na počítači) na jedné straně a s jeho schránkou (vedenou MDA) na druhé straně. Obvykle se jedná buď o klientský program (Outlook, Thunderbird, Evolution apod.) nebo o webový prohlížeč se spuštěným webovým rozhraním schránky.

Takže pokud posíláme e-mail, nejdřív naše MUA aplikace naváže spojení s naším mail serverem plnícím roli MTA. Odešle mu e-mail, ten je pak přeposlán dalšímu MTA na cestě, atd. (zpráva postupně prochází do nadřízených domén a sítí a pak v opačném směru v hierarchii sítě příjemce směrem dolů), v doménách, kterými prochází, se v DNS záznamech hledají záznamy typu MX. Poslední MTA již zprávu předá cílovému MDA, který zajistí umístění do schránky příjemce.

Poznámka:
Pokud to umíme, můžeme se obejít i bez MUA. Spojení s MTA se totiž dá navázat i přímo přes Telnet nebo SSH, přičemž postupně odesíláme ty informace, které by jinak odesílal MUA agent.

5.3.2 Protokoly

Pro práci s elektronickou poštou potřebujeme dva druhy protokolů:

- protokol zajišťující odeslání zprávy a její transfer až do schánky adresáta, dnes se prakticky výhradně používá protokol SMTP,
- protokol pro přístup ke zprávám ve své schránce, je na výběr mezi protokoly POP a IMAP.

To vše (i včetně schématu na předchozí straně) platí, pokud opravdu používáme e-mailového klienta. Jestliže komunikujeme se schránkou přes webové rozhraní, pak na prvním a posledním úseku cesty najdeme většinou protokol HTTP nebo lépe HTTPS.

Protokol SMTP (Simple Mail Transfer Protocol) tedy slouží k transferu (přeposílání) zprávy směrem k poštovním serverům. Právě podle tohoto protokolu také hovoříme o SMTP serverech (to jsou servery poskytující službu MTA). Pomocí SMTP komunikujeme se svým MTA serverem, když odesíláme e-mail, a stejně komunikují mezi sebou jednotlivé MTA servery.

SMTP komunikuje na portu 25, SMTP zprávy jsou zapouzdřovány do TCP nebo UDP segmentů (většinou TCP).

U protokolu HTTP jste si určitě všimli, že záhlaví je poměrně variabilní (podle směru a fáze komunikace). Stejně je to i SMTP. Mezi MUA a MTA je nejdřív navázáno TCP spojení (s číslem portu 25), dojednáno zabezpečení (abychom nepřenášeli přihlašovací údaje jako obyčejný text) a následně se odesílají „tematické“ SMTP zprávy s adresami odesílatele a příjemce, předmětem a dalšími částmi. Zprávu pak zkompletuje náš lokální MTA.

Na další cestě (když už na obou stranách spoje najdeme MTA) je už komunikace jednodušší. Zpráva se přenáší „v celku“, a na každém MTA se na její začátek přidá nový záznam – údaj o dotyčném MTA. Z toho vyplývá, že velikost SMTP zprávy po cestě postupně narůstá, a příjemce si pak v záhlaví může ověřit, přes které MTA (SMTP) servery zpráva šla.

Protokol POP (Post Office Protocol) ve verzi 3, tedy POP3, je již poněkud zastaralý. Jeho účelem je přístup do e-mailové schránky za účelem stažení zpráv do lokálního úložiště. Ve verzi 3 komunikuje na portu 110, používá protokol TCP.

Protokol IMAP (Internet Message Access Protocol) je již modernější náhradou protokolu POP. Umožňuje přistupovat do schránky, ale nejen pro stažení zpráv – nabízí službu on-line správy se vším všudy, tedy například:

- číst, kopírovat a mazat vybrané zprávy přímo ve schránce na MDA,
- třídit zprávy, označovat, volněji používat vyhrazený paměťový prostor ve schránce,

- prohlížení či stažení nejen celých zpráv, ale také jen záhlaví zpráv, čímž se šetří přenosová kapacita sítě,
- ke schránce a jednotlivým zprávám lze přistupovat z různých klientských zařízení a na každém mít vlastní kopii zpráv ve schránce,
- pokročilé možnosti automatizace, například automatické zálohování zpráv.

IMAP komunikuje na TCP portu 143. Dnes se téměř výhradně používá protokol IMAP, s protokolem POP3 se setkáváme výjimečně.

5.3.3 Komunikace a nastavení

V případě elektronické pošty jde také o spojovanou komunikaci, tedy protokol TCP, do kterého jsou zapouzdřovány SMTP a IMAP zprávy. To vždy znamená komunikaci typu klient-server.

U SMTP je klientem to zařízení, které odesílá SMTP zprávu. SMTP server (tedy MTA) naslouchá na portu 25 a očekává příchozí zprávy.

IMAP klient je příjemcem zprávy, a musí se (obvykle ve vhodných intervalech) dotazovat IMAP serveru na změny. IMAP server naslouchá na portu 143 a očekává právě tyto dotazy.

Když konfigurujeme MUA (poštovního klienta), musíme mu sdělit, kdo je jeho protějškem – zadáváme jmennou adresu SMTP serveru a číslo portu pro komunikaci s ním, a dále adresu IMAP serveru s číslem portu.

Pokud se jedná o pouhé webové rozhraní ke schránce, pak něco takového nemusíme dělat, protože konfigurace je součástí webové aplikace a souvisí s adresou, kterou píšeme do adresního řádku. Navíc je otázkou, zda opravdu jde o MUA, protože s SMTP/IMAP serverem komunikuje protokolem HTTPS.

Poznámka:

Dnes se již běžně s poštovními servery komunikuje zabezpečeně. Pozor, to neznamená, že by zprávy byly šifrovány bez přerušení po celé cestě nebo digitálně podepisovány. Pouze se šifruje spojení mezi dvěma komunikujícími zařízeními, aby například nebylo možné odposlechnout přihlašovací údaje nebo samotnou zprávu, SMTP server „vidí“, co píšeme.

Zabezpečená komunikace probíhá trochu jinak než jednoduchá nezabezpečená. Mezi protokol SMTP nebo IMAP a protokol TCP se vsouvá bezpečnostní protokol SSL nebo TLS, což mimo jiné znamená použití jiného čísla portu. Takže pokud komunikaci SMTP zabezpečujeme, jde o port 465, v případě protokolu IMAP to je port 993 a u POP3 port 995.

Poštovní protokoly používají adresaci založenou na podobném principu jako HTTP, tedy URL. Jako protokol se uvádí `mail:` a ve zbytku lokátoru je název schránky a jmenná adresa MDA.

Příklad

URL pro elektronickou poštu vypadá takto:

`mail:jan.novak@firma.cz`

Oproti HTTP URL je přehozen význam druhé a třetí části – v druhé části máme název schránky a až třetí část je jmenná adresa. Symbol zavináče `@` se obvykle čte „at“ (jan.novak at firma.cz).

Další informace:

- https://www.siteground.com/tutorials/email/pop3-imap-smtp-ports.htm
- https://www.port25.com/how-to-check-an-smtp-connection-with-a-manual-telnet-session-2/

Poznámka:

Relay servery (Open Mail Relay) se nazývají SMTP servery, které při komunikaci s uživatelem či MDA odesílajícím e-mail neověřují, zda souhlasí jméno a adresa odesílatele. Jsou často zneužívány k rozesílání spamu, proto síť s takovým serverem bývá považována za nedůvěryhodnou.

5.3.4 MIME

MIME (Multipurpose Internet Mail Extensions) je standard pro definování formátů posílaných dat. Nepoužívá vlastní PDU, ale určuje, jakým způsobem mají být reprezentovány různé typy dat v PDU jiných aplikačních protokolů. Tento standard používáme velmi často právě v e-mailech, kde bývá využit pro reprezentaci téže informace v různých formátech (čistý text, HTML apod.) a také pro reprezentaci vložených příloh.

Některé nejznámější MIME typy jsou:

- *text* – pro textové formáty (například čistý text/plain, html, rtf),
- *image* – pro obrázky (bmp, gif, png, jpeg apod.),
- *audio* – pro zvukové soubory (audio/mpeg, audio/mp4, audio/ogg, atd.),
- *video* – pro videosoubory (napříkad h263, h264, audio/mp4),
- *application* – pro soubory aplikací, moduly a datové soubory aplikací, jednoduše binární soubory (například java-vm, json, msword, octet-stream, pdf),
- *multipart* – „multiformát“, který může například indikovat, že následují tatáž data postupně v několika různých formátech (multipart/alternative), případně postupně za sebou více různých objektů, například že je přiložen digitální podpis, atd.

Pokud se použije MIME, je ve zprávě místo jednoduchého textu nejdřív (jedno či více) MIME záhlaví následované MIME obsahem. V MIME záhlaví najdeme verzi MIME, typ obsahu (Content-type), případně kódovací metoda pro následující data a další podle potřeby.

Další informace:

http://www.iana.org/assignments/media-types/media-types.xhtml

5.4 Souborové služby

5.4.1 Protokol FTP

Souborový server (file server) poskytuje službu úložiště dat a unifikovaného přístupu k těmto datům. Klient tedy na souborový server ukládá data (upload), stahuje si data (download), popřípadě data aktualizuje.

Protokol FTP (File Transfer Protocol) slouží ke komunikaci se souborovými servery a má vyhrazené porty 20 a 21, přičemž:

- port 21 je používán pro zasílání řídících informací (příkazy),
- port 20 je používán pro zasílání dat.

Takže FTP server naslouchá především na portech 20 a 21, přesněji naslouchá vždy jen na jednom z nich – standardně na 21 (pro příkazy), a pokud je přes port 21 vyjednán přenos dat, pracuje se pro změnu jen s portem 20. Co se transportních protokolů týče, lze sice používat TCP i UDP, ale prakticky je použitelná jen TCP komunikace, protože mnohé souborové servery vyžadují autentizaci, pro kterou potřebujeme navázat spojení.

Příklad

URL pro souborový server vypadá podobně jako u HTTP: `ftp://fileserver.firma.cz`

Zabezpečená komunikace s FTP serverem probíhá obdobně jako u předchozích protokolů, tedy mezi FTP a TCP nasuneme protokol SSL nebo TLS (označujeme jako FTPS), ale lepší alternativou je přímo použití zabezpečeného protokolu SSH (port 22). SFTP se označuje přenos FTP přes SSH, což je poněkud komplikovanější komunikace.

Protokol FTP se typicky používá ke stahování dat z webu (především multimediálních), protože pro protokol HTTP (který by to taky zvládl) bývá práce s velkými soubory zbytečně obtížná. Dalším typickým použitím je správa vlastních webových stránek – práce se soubory, které tyto webové stránky na serveru tvoří (na serveru se spuštěným HTTP serverem bývá i FTP server).

Zatímco na straně serveru musí běžet FTP server (coby proces, služba, démon), na klientském zařízení potřebujeme *FTP klienta*. FTP klienti bývají součástí webových prohlížečů a správců souborů (například Total Commander nebo Free Commander), nebo můžeme zvolit specializovaný program (FileZilla, český WinSCP apod.).

Abychom mohli daného FTP klienta používat pro přístup ke konkrétnímu serveru (resp. nám vyhrazené části), musíme si obvykle pro tuto komunikaci vytvořit profil (to není nutné, pokud se uživatelé nemusejí autentizovat). V profilu je třeba zadat název serveru (tj. doménové jméno), umístění (plná URL serveru včetně protokolu), číslo protokolu (většinou 21) a autentizační informace (jméno a heslo), případně lze tyto informace zadávat dynamicky vždy při přístupu na daný server.

S FTP serverem komunikujeme pomocí FTP příkazů. Pár nejzákladnějších:

- `open server` – otevře relaci k danému serveru, parametr je název souborového serveru,
- `get soubor` – chceme si stáhnout zadaný soubor (ze serveru k sobě),
- `send soubor` – odesíláme zadaný soubor od sebe na server,
- `!` – tímto příkazem se přepínáme mezi datovým prostorem na našem počítači a datovým prostorem serveru, platí pro pohyb v adresářové struktuře na našem počítači nebo serveru,
- `lcd adresář` – přesun do jiného adresáře buď na našem počítači nebo na serveru, místo působení se přepíná příkazem `!`, používají se stejné způsoby zadávání adresáře jako v příkazovém řádku (včetně `..` pro přesun o úroveň výše).

Další informace:

FTP příkazů je samozřejmě mnohem více. Stručný návod a přehled najdete například na http://www.computerhope.com/issues/ch001246.htm.

Poznámka:

Jednoduchého FTP klienta ve skutečnosti máme k dispozici v každém operačním systému, dokonce i ve Windows. Když na příkazovém řádku zadáme `ftp`, dostaneme se do příkazového prostředí tohoto klienta. Pak lze otevřít relaci s určitým serverem, přičemž jsme dotázáni na jméno a heslo, a po úspěšném přihlášení můžeme zadávat FTP příkazy. Takže začátek relace vypadá takto:

```
ftp
open fileserver.firma.cz
...
```

Pak jsme požádáni o jméno a heslo, následně už můžeme zadávat FTP příkazy. Relaci ukončíme příkazem `quit` nebo `bye`.

5.4.2 Sdílení prostředků v lokální síti

Pro sdílení prostředků v lokálních sítích (často typu peer-to-peer) se kromě FTP může používat *protokol SMB* (Server Message Block), také nazývaný CIFS (Common Internet File System). Je to aplikační protokol poskytující autorizovaný přístup ke zdrojům typu souborů, tiskáren apod. Zprostředkovává přístup k souborovým a tiskovým serverům.

SMB byl původně vyvinut společností IBM ve spolupráci s Microsoftem a dodnes je oblíbený zejména v sítích s Windows servery a klienty, ale existuje také open-source implementace pro jiné operační systémy nazvaná *Samba*.

Protokol definuje komunikaci typu klient-server – žadatel o prostředek je klient a poskytovatel prostředku je server. SMB zprávy se nazývají *bloky* (taky je to v názvu protokolu), jejich formát je stejný v obou směrech komunikace, jen do odpovědi se mohou přidat požadovaná data.

5.5 Přidělování IP adres a protokol DHCP

Každé zařízení v síti potřebuje nějak získat IP adresu. Můžeme ji buď na každém zařízení zadat ručně nebo nechat DHCP server, aby ji připojovaným zařízením přiděloval dynamicky.

IP adresa může být získána těmito způsoby:

- *dynamická alokace* – uživatel se nemusí o nic starat, při připojení zařízení do sítě je tomuto zařízení automaticky přiřazena IP adresa, může být pokaždé jiná,
- *staticky* – uživatel má předem určenou IP adresu, do konfigurace se dostane takto:
  - uživatel ji *ručně* zadá na patřičné místo v konfiguraci síťového rozhraní,
  - *statická alokace*: tato adresa je při připojení zařízení do sítě tomuto zařízení automaticky přiřazena (jako u dynamické alokace, ale adresa je rezervovaná pro toto zařízení).

Nás bude zajímat první případ a z druhého případu možnost statické alokace, což vše probíhá podle *protokolu DHCP* (Dynamic Host Configuration Protocol). Protokol DHCP tedy nabízí mechanismus distribuce konfigurace síťového rozhraní. Klientské zařízení může od DHCP serveru během dynamické nebo statické alokace získat různé informace:

- IP adresu, masku sítě,
- adresu výchozí brány,
- adresu DNS serveru,
- další informace.

Takže rozhodně nejde jen o IP adresu a masku sítě. Klient se taky dozví, kudy vede cesta ze sítě a kdo v síti umí překládat jmenné adresy na IP adresy. Podle potřeby může server distribuovat i další informace.

S přechodem od IPv4 na IPv6 byly vytvořeny nové verze i pro jiné protokoly (zatím jsme se s tím setkali u ICMPv4/v6), totéž potkalo i protokol DHCP. Změny v IP byly prostě tak rozsáhlé, že musel být značně pozměněn i tento protokol.

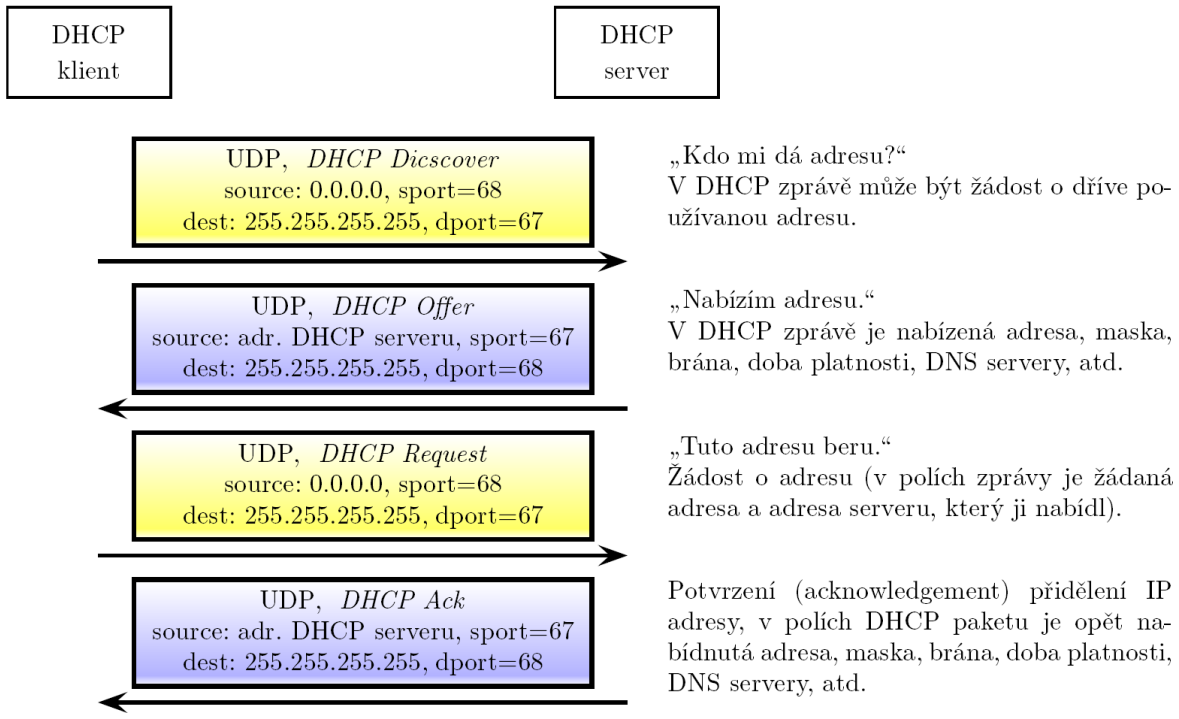
Starší DHCPv4 je jednodušší a jako svůj nosný protokol směrem k transportní vrstvě využívá protokol Bootstrap (Bootp). Celá struktura je následující: DHCP zpráva se zapouzdří do Bootstrap zprávy (resp. je její součástí) a výsledná zpráva se zapouzdří do UDP segmentu. Číslo portu je 67 na straně serveru a 68 na straně klienta.

DHCPv6 je již komplexnější a nevyužívá protokol Bootstrap, sám vytváří zprávy na aplikační vrstvě. Naproti tomu úzce spolupracuje s protokolem ICMPv6 (ale nikoliv ve smyslu zapouzdřování). DHCPv6 zprávy se také zapouzdřují do UDP segmentu, používají se čísla portů 547 na straně serveru a 546 na straně klienta.

Poznámka:
Všimněte si, že na rozdíl od předchozích aplikačních protokolů používá DHCP klient port z rozsahu „dobře známých“ portů a nikoliv dynamické porty. Uvědomte si, že DHCP sice pracuje na aplikační vrstvě, ale rozhodně nekomunikuje s (různými) aplikacemi a procesy v různých oknech či záložkách – nemůže nastat situace, kdy by dvě různé aplikace chtěly přidělit adresu. Proto nemusí klient využívat dynamické porty, kterými by odlišil jednotlivé aplikace – není co odlišovat.

Na obrázku 5.7 je naznačen postup získání IPv4 adresy. Komunikuje se pouze broadcastově, protože klient ještě nemá přidělenou IP adresu (v obou směrech – klient sice postupně zjistí adresu serveru, ale i tak mu posílá broadcasty). Jednotlivé kroky:

1. *DHCP Discover* (poptávka) – klient zjišťuje, kdo v síti mu může přidělit adresu (nezná adresu DHCP serveru). V žádosti posílá svou MAC adresu a seznam parametrů, které požaduje (Parameter Request List – masku sítě, adresy DNS serverů, adresu routeru/brány apod.).
2. *DHCP Offer* (nabídka) – DHCP server obdržel žádost a v odpovědi posílá nabízenou adresu (nebo jejich rozsah), dobu platnosti adresy (Lease Time), svou vlastní adresu, adresy DNS serverů apod. Pokud je v síti víc DHCP serverů, může klientovi přijít i víc než jedna nabídka. Nabídnutá adresa je na určitou krátkou dobu (například 1 minutu) dočasně rezervována.
3. *DHCP Request* (žádost) – klient přijme nabízenou adresu a zároveň znovu pošle seznam dalších požadovaných parametrů, stejně jako v poptávce. Pokud přišlo víc nabídek, odpoví jen na jednu.



Obrázek 5.7: Získání adresy přes DHCPv4

4. *DHCP Ack* (potvrzení) – server klientovi potvrdí přidělení IP adresy a (znovu) pošle hodnoty dalších požadovaných parametrů. Adresa je klientovi definitivně přidělena a DHCP server ji má zaznamenanou v databázi.

   Jestliže server zjistí na síti zprávu DHCP Request, která není určena jemu (ale jinému DHCP serveru), znamená to, že klient si vybral jiného poskytovatele, tedy dočasně rezervovanou adresu uvolní.

Pokud jde o *znovupřidělení téže adresy* (například tehdy, kdy klient má adresu přidělenou, ale blíží se vypršení doby platnosti – *Lease Time*), probíhají pouze poslední dva kroky, protože klient zná svého poskytovatele adresy (nemusí se poptávat) a nabídka už jednou taky proběhla. Takže jen žádost a potvrzení.

Až po DHCP Ack nemůže klient při prvním přidělování používat skutečnou IP adresu, do té doby má 0.0.0.0 (nedefinovanou). Ovšem při znovupřidělování toto neplatí – klient používá svou dříve přidělenou adresu a komunikace je unicastová (oba uzly se navzájem „znají“).

Při použití DHCPv6 je víc možností jak může komunikace proběhnout, tyto možnosti probereme v následující kapitole. Jedním z rozdílů oproti DHCPv4 je také to, že pro tento protokol je již definován vlastní formát zprávy, nepoužívají se již zprávy protokolu Bootstrap.

5.6 Další typy serverů

Databázový server je server, na kterém běží některý databázový systém (MySQL, PostgreSQL, Oracle, MS SQL apod.).

Pro databázové servery je typické, že uživatel obvykle nekomunikuje přímo s nimi, ale existuje zprostředkovatel – webový server poskytující uživatelské rozhraní k databázovému serveru. Takže

uživateli je zobrazena webová stránka, na které zadá, co od databáze vlastně chce, odešle se webovému serveru, ten vše zkontroluje a odešle dotaz do databáze na databázovém serveru. Odpověď jde opět přes webový server, který ji „zformátuje“ do webové stránky, aby byl výstup uživatelsky příjemnější.

Tiskový server je server poskytující tiskové služby (vede tiskové fronty připojených tiskáren, přijímá požadavky od klientů ze sítě a řadí je do front).

Pokud k běžnému počítači máme připojenou tiskárnu, kterou máme nasdílenou do sítě, pak náš počítač taky funguje jako tiskový server. V současné době je tiskový server přímo vestavěn především ve větších síťových tiskárnách, aby nebylo nutné vedle nich provozovat ještě další „zprostředkující“ zařízení.

Aplikační server je server poskytující do sítě služby určité (síťové) aplikace. Jedná se vlastně o proces (službu, démona) běžící na daném hardwarovém serveru, přičemž tento proces přijímá požadavky ze sítě.

V každém případě je potřeba, aby mezi klientem a aplikačním serverem byl webový server, který bude zprostředkovávat komunikaci. Není to jen kvůli pohodlí uživatelů, ale také z důvodu lepší ochrany aplikačního serveru. Samotné aplikační servery často dokonce ani nemají implementovaný protokol HTTP, ale komunikují (s webovým serverem) pomocí jiných protokolů na jiných než běžných portech.

Za speciální typ aplikačního serveru můžeme považovat například i databázový server, protože na něm běží databázový systém jako speciální proces a mezi klientem a databázovým serverem taky obvykle bývá webový server.

Poznámka:

Stejně funguje taky nám známý systém STAG. Databáze uživatelů, studijních plánů, předmětů, sylabů, hodnocení,... je na databázovém (aplikačním) serveru, ale uživatelé se přímo dostanou jen k webovému rozhraní (portálu), kterému rozhodně neposílají databázové (SQL) dotazy, ale pouze webové formuláře. Až webový server podle formulářů sestaví SQL dotaz, přepošle ho do databáze a v opačném směru pak podle výsledku dotazu sestaví webovou stránku, kterou následně pošle klientovi.

5.7 Vzdálená konfigurace

5.7.1 Telnet

Pokud potřebujeme konfigurovat zařízení, u kterého z nějakého důvodu nemůžeme přímo sedět (je daleko, je hůře dostupné, nemá klávesnici a obrazovku apod.), potřebujeme protokol, který by nám dokázal zprostředkovat vzdálený přístup na dané zařízení v takovém rozsahu, jako když u toho zařízení přímo sedíme. Při správě serveru si obvykle vystačíme s textovým rozhraním (ano, dokonce i Windows server lze z bezpečnostních a výkonových důvodů nainstalovat bez grafického rozhraní – instalace typu „Server Core“), kdežto při správě uživatelských zařízení na dálku se může hodit i přenos grafického rozhraní uživatele.

Protokol Telnet slouží pro interaktivní vzdálený (textový) přístup typu klient-server k zařízením přes síť, a to s možností autentizace (zadáváme jméno a heslo). V reálu provádí *emulaci terminálu* – terminál je specializované zařízení skládající se z obrazovky a klávesnice a připojené například přes síť k serveru. Pokud terminál emulujeme, znamená to, že se náš počítač dokáže chovat jako terminál, třebaže terminálem není.

Vpodstatě se Telnet dá provozovat na jakémkoliv zařízení (klientském i serverovém), ale většinou bývá zablokován či jiným způsobem je znemožněno jeho používání. V čem je problém? Telnet totiž vznikl v době, kdy síť byla bezpečným místem (a taky byla mnohem menší) a nebylo nutné nic šifrovat. Při autentizaci přenáší jméno a heslo v plain textu (čistý text) a kdokoliv, kdo se k síti dostane, si tyto údaje může odposlechnout.

Dnes se Telnet používá pouze v provozech (vnitřních sítích), o kterých si administrátor myslí, že jsou bezpečné (všimněte si, že tu nepíšu v bezpečných sítích – nic takového jako bezpečná síť totiž ve skutečnosti neexistuje). Logičtější použití je při ladění serverového softwaru, přičemž jsou klient i server odděleni od sítě a technicky se do komunikace nikdo nemůže dostat.

Telnet používá spojovanou komunikaci (protokol TCP) přes port 23 (klient používá dynamické porty). Nejdřív je navázána TCP komunikace na portu serveru 23 Pokud je vyžadována autentizace, jsou poslány a potvrzeny autentizační údaje, a dál už záleží, co konkrétně chceme dělat.

Poznámka:

Pokud se nám povede zpřístupnit Telnet na našem počítači, můžeme začít Telnet relaci takto:

```
telnet
open adresa.serveru port
...
```

Jako adresu serveru zadáme ten server, na který chceme vzdáleně (i přes Internet) přistupovat, číslo portu zadáváme jen tehdy, když má být jiné než 23. Například pokud chceme přistupovat k webovému serveru tím způsobem, že v příkazech budeme posílat části HTTP zprávy, použijeme port 80. Následně jsme požádáni o jméno a heslo (pokud je dotyčný server chráněn autentizací) a pak už můžeme zadávat příkazy.

V UNIXových serverech je naprosto běžné, že se k nim dá přistupovat vzdáleně formou emulace terminálu (ať už přes Telnet nebo některý bezpečnější protokol), takto můžeme zadávat prakticky kterýkoliv příkaz tak, jako bychom seděli přímo u dotyčného zařízení. Prostě se nám zobrazí prompt a my zadáváme příkaz. Ve Windows se tento vzdálený přístup poněkud omezenější, protože na rozdíl od UNIXových systémů existuje mnoho úloh, které v textovém režimu prostě nelze provést, a navíc Windows vzdálený přístup poněkud hůře zvládají.

Často (především při testování a ladění) se využívá toho, že mnoho aplikačních protokolů je textově orientovaných (ne binárních) a přes telnet tak můžeme posílat i jejich zprávy. Dá se to tak dělat například s protokolem HTTP, SMTP a dalšími.

Příklad

Pokud chceme přes Telnet otestovat webový server, postupujeme takto:

```
telnet www.server.cz 80
GET /index.html HTTP/1.1
```

Nejdřív jsme navázali spojení se zadaným serverem, a to na portu 80 (protože chceme poslat něco, co má být HTTP zprávou). Webové servery většinou nevyžadují autentizaci, tedy nebudou vyžadovány přístupové údaje. Dalším příkazem odešleme HTTP požadavek typu GET (pozor, musí být velkými písmeny, jinak bude hlášena chyba), žádáme o poslání souboru `index.html` z kořenového adresáře serveru, taky se přidává informace o verzi protokolu HTTP. V odpovědi se nám vypíše HTTP záhlaví a taky tělo zprávy (tedy HTML kód).

Podobně se dá komunikovat například i s SMTP serverem na portu 25, dokonce takto můžeme odeslat e-mail, jen je to trochu časově složitější než vyžádání webové stránky.

5.7.2 SSH

SSH (Secure Shell) je bezpečnější variantou protokolu Telnet, ale kromě samotného zabezpečení komunikace má oproti Telnetu i další přídavnou funkčnost. Slouží k zabezpečenému přístupu ke vzdálenému zařízení, a to v různých směrech. Kromě toho, že umožňuje bezpečnou konfiguraci, plní taky podobnou (ale zabezpečenou) úlohu jako FTP (přenos souborů), dovoluje monitorovat procesy a zdroje na vzdáleném systému, vytvářet šifrované VPN tunely (dlouhodobé zabezpečené spojení), atd. V současné době se používá SSH verze 2.

Jedná se o komunikaci typu klient-server přes TCP (navazuje se spojení), na straně serveru je port 22. Vzhledem k tomu, že zde jde především o bezpečnost, je lepší na serveru nakonfigurovat jiný port než 22. Postup této konfigurace záleží na konkrétním produktu, obvykle jde o změnu v některém konfiguračním souboru. Ovšem pokud na serveru změníme číslo portu pro SSH, klient musí při navazování spojení toto číslo použít (a tedy být o něm informován).

Nejznámější implementací protokolu SSH je open-source projekt *OpenSSH*. Pro tento produkt existuje klientská i serverová varianta a na UNIXových systémech včetně Linuxu a MacOS X již obvykle bývá nainstalován. Pro Windows existuje klientská varianta a slouží pro přístup k UNIXovým serverům, a v poslední době se dokonce objevila i serverová implementace pro Windows.

Pro Windows máme k dispozici i program *PuTTY*, který však existuje pouze v klientské variantě. Opět se používá pro vzdálený přístup k UNIXovým serverům.

SSH konverzace vypadá takto:

- Nejdřív je navázáno TCP spojení, začíná klient.
- SSH klient i server se „představí“ (vzájemně si sdělí svou verzi), začíná server.
- Server sdělí klientovi, které šifrovací algoritmy zvládá, klient si v odpovědi mezi nimi vybere.
- Následuje bezpečnostní procedura – dojde k výměně šifrovacího řetězce atd., podle zvoleného algoritmu. Účelem je zajistit, aby se v následující komunikaci nevmísil do spojení někdo, kdo by mohl odposlechnout provoz.
- Po dokončení bezpečnostní procedury je již další provoz šifrován, včetně případné autentizace, pokud je vyžadována.

Další informace:

- http://www.dsl.cz/jak-na-to/jak-na-ssh
- http://www.debianhelp.co.uk/ssh.htm

5.8 Správa sítě

Zde se nebudeme podrobně věnovat správě sítě, pouze si něco řekneme o protokolu, který k tomuto účelu slouží.

✎ *Protokol SNMP* (Simple Network Management Protocol) se používá pro sběr dat ze sítě pro účely správy (managementu) této sítě. Taktéž jde o komunikaci typu klient-server, přičemž klientská část je dnes implementována prakticky na každém zařízení v síti (počítače, aktivní síťové prvky, čidla, tiskárny, atd.).

Klient, kterému se v SNMP terminologii říká *agent*, shromažďuje data týkající se zařízení, na kterém běží, kdežto serverová část (*správce*, network manager) si tato data může kdykoliv vyžádat z různých zařízení v síti a analyzovat. Komunikace může být jednoho z těchto typů:

- *dotaz–odpověď*, přičemž správce posílá dotaz a agent odpovídá,
- *trap* – aktivita je na straně agenta (informuje správce o události na zařízení, na kterou je třeba reagovat).

V současné době se můžeme setkat se dvěma používanými verzemi – SNMPv2 a SNMPv3. Verze 2 již není považována za příliš bezpečnou.

✎ SNMP komunikuje na obvykle přes protokol UDP, ale může použít i TCP. Různé verze používají různé porty.

- Při komunikaci dotaz–odpověď (správce se ptá a agent odpovídá) agent podle verze SNMPv2 naslouchá na portu 161, podle SNMPv3 to je port 10 161. Správce použije některý dynamický port.
- Při komunikaci typu trap správce podle SNMPv2 naslouchá na portu 162, podle SNMPv3 na portu 10 162. Agent použije některý dynamický port.

5.9 Přehled protokolů a portů

V této kapitole jsme se dozvěděli, že aplikační protokoly využívají protokol TCP (pro spojovanou potvrzovanou komunikaci) nebo UDP (pro nespojovanou nepotvrzovanou, tedy datagramovou službu), což znamená, že jejich zprávy jsou zapouzdřovány do TCP nebo UDP segmentů a v případě TCP je také navazováno spojení.

V TCP a UDP segmentech se v záhlaví uvádí číslo portu, které na straně serveru určuje konkrétní službu, se kterou se komunikuje (taky může znamenat typ zapouzdřené zprávy, ale ne vždy uvnitř segmentu něco je), na straně klienta to bývá konkrétní aplikace či její část.

HTTP(S)	SMTP	IMAP	FTP	Telnet	SSH	DNS	DHCP	SNMP
80	**25**	143					**67, 68**	161, 162
443	465	993	**20, 21**	**23**	**22**	**53**	547, 546	10 161, 10 162
TCP							UDP	

Tabulka 5.1: TCP a UDP porty s protokoly

Kapitola 6

Síťové adresy a směrování

Rychlý náhled: V této kapitole se opět vrátíme k síťové vrstvě modelu ISO/OSI (internetové v modelu TCP/IP), tedy L3, ale především ze začátku se budeme pohybovat i na vrstvě L2 (linkové) nebo na hranici mezi nimi. Zaměříme se na práci s adresami, překlad adres, typy IP adres, práci s podsítěmi a také na základy směrování a používání směrovacích protokolů.

Klíčová slova: ARP, NDP, tabulka sousedů, třída, loopback, soukromá adresa, podsíťování (subnetting), VLSM, CIDR (nadsíťování, agregace cest), DHCP, dynamická alokace, autokonfigurace, EUI-64, SLAAC, privacy extensions, NAT, PAT, směrování, směrovací tabulka, metrika, autonomní systém, směrovací protokol, algoritmus vektoru vzdáleností, RIP, algoritmus stavu spoje, OSPF, BGP

Cíle studia: Po prostudování této kapitoly porozumíte problematice dosazování MAC a IP adres do rámce/paketu, budete umět pracovat s adresními rozsahy pro IPv4, zjistíte, jak se získává IPv4 a IPv6 adresa, porozumíte problematice překladu IP adres, naučíte se principu směrování paketů mezi routery, porozumíte fungování směrovacích algoritmů.

6.1 Adresy v rámcích a paketech

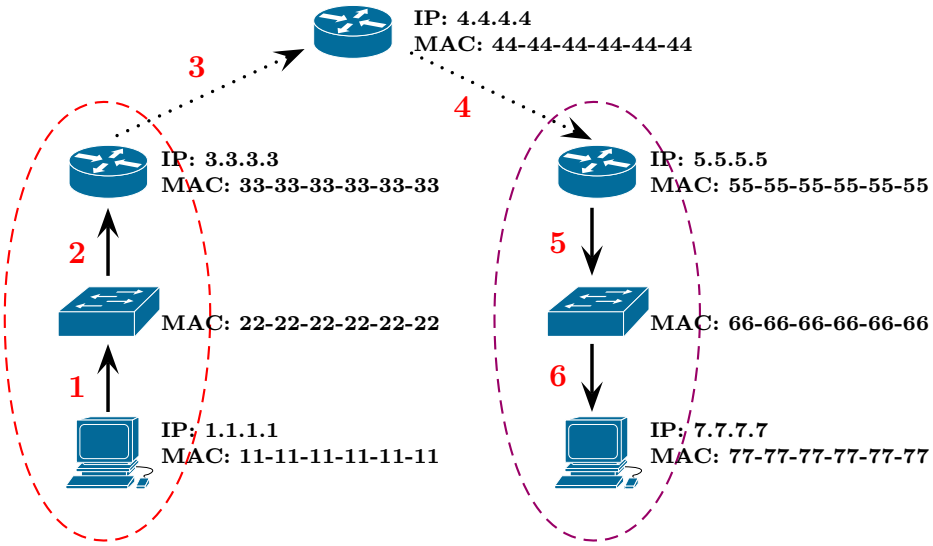
Když odesíláme IP paket, potřebujeme samozřejmě IP adresu zdroje (naši) a příjemce, ale musíme si uvědomit, že IP paket se má zapouzdřit do rámce (třeba ethernetového), a tam jsou uvedeny i MAC adresy zdroje a cíle.

Do rámce, který odesíláme, jako zdrojovou dáme vlastní MAC adresu, ale jakou dát tu cílovou? Určitě nemůžeme použít MAC adresu příjemce, protože tu neznáme. O příjemci (třeba serveru) máme totiž jen omezenou informaci (jenom IP nebo jmennou adresu), navíc pod toutéž jmennou či IP adresou může v reálu existovat více než jedno fyzické zařízení, například z důvodu vyrovnávání zátěže nebo kvůli údržbě. Jediná MAC adresa, kterou známe, je MAC adresa brány.

Příklad

Na obrázku 6.1 je naznačena struktura sítě, přes kterou máme poslat paket. Zdrojovým zařízením

je počítač s IP adresou 1.1.1.1, cílovým zařízením počítač s IP adresou 7.7.7.7.



Obrázek 6.1: Vztah mezi L2 a L3 adresováním

Jak to proběhne:

- Na prvním úseku cesty směřuje PDU k switchi, který vidí jen do rámce; rámec a paket zapouzdřený v rámci obsahují adresy:

Zdroj:	IP 1.1.1.1	MAC 11-11-11-11-11-11	adresy odesílatele
Cíl:	IP 7.7.7.7	MAC 33-33-33-33-33-33	IP adresa příjemce, MAC adresa brány

  Všimněte si, že IP a MAC adresa cíle k sobě nepatří, ale to vůbec nevadí.

- Na druhém úseku se PDU dostává k bráně a adresy se zatím nezměnily:

Zdroj:	IP 1.1.1.1	MAC 11-11-11-11-11-11	adresy odesílatele
Cíl:	IP 7.7.7.7	MAC 33-33-33-33-33-33	IP adresa příjemce, MAC adresa brány

- Třetí úsek cesty již vede mimo síť odesílatele. Router, který plní roli brány, přijal a rozbalil rámec, rozbalil vnořený IP paket a znovu ho zabalil, tentokrát s jinými MAC adresami.

Zdroj:	IP 1.1.1.1	MAC 33-33-33-33-33-33	IP adresa odesílatele, MAC adresa začátku úseku
Cíl:	IP 7.7.7.7	MAC 44-44-44-44-44-44	IP adresa příjemce, MAC adresa konce úseku

- Podobně to bude na čtvrtém úseku:

Zdroj:	IP 1.1.1.1	MAC 44-44-44-44-44-44	IP adresa odesílatele, MAC adresa začátku úseku
Cíl:	IP 7.7.7.7	MAC 55-55-55-55-55-55	IP adresa příjemce, MAC adresa konce úseku

- Pak již jsme v síti příjemce. Brána v síti příjemce „zná“ MAC adresy uzlů ve své síti.

Zdroj:	IP 1.1.1.1	MAC 55-55-55-55-55-55	IP adresa odesílatele, MAC adresa cizí brány
Cíl:	IP 7.7.7.7	MAC 77-77-77-77-77-77	adresy příjemce

- Na šestém úseku už jen zůstáváme na vrstvě L2 a nemá smysl adresy měnit.

Zdroj:	IP 1.1.1.1	MAC 55-55-55-55-55-55	IP adresa odesílatele, MAC adresa cizí brány
Cíl:	IP 7.7.7.7	MAC 77-77-77-77-77-77	adresy příjemce

Z toho plyne, že opravdu nemusíme znát MAC adresu cíle, stačí nám IP adresa, a cíl zase nemusí znát naši MAC adresu. Cíl sice dostane rámec obsahující „nějakou“ MAC adresu, ale ta by byla naše pouze v případě, že jsme ve stejné síti. Taky z toho plyne další informace – switch, ke kterému jsme přímo připojeni, je pro nás vlastně „neviditelný“, k jeho MAC adrese se v rámcích nedostaneme (k jeho IP adrese taky ne – žádnou nemá).

Definice (Internetworking)

Internetworking je mechanismus propojování sítí, tedy zajišťování komunikace mezi (inter) sítěmi (networks). Tento mechanismus probíhá na vrstvě L3, a to na zařízeních typu router (směrovač) nebo switch (přepínač) s funkcionalitou vrstvy L3.

6.2 Objevování sousedů

6.2.1 Tabulky sousedů

Představme si tuto situaci: známe svou IP adresu, masku (či prefix) a IP adresu brány. Potřebujeme poslat paket do cizí sítě, a tedy přes bránu. Jenže „kterým směrem“ je brána? Přes které síťové rozhraní je dostupná a jaká je její MAC adresa?

Vlastně podobný problém musíme řešit i tehdy, když chceme poslat IP paket někomu do vnitřní sítě. Třeba znám IP adresu dotyčného zařízení, ale jakou tedy má MAC adresu a přes který port je toto zařízení dostupné?

Co se týče „směru“, na ten máme jiné mechanismy, teď se zaměříme na mapování IP adresy na korespondující MAC adresu.

Mapování IP adres na MAC adresy provádí v případě IPv4 *protokol ARP* (Address Resolution Protocol) a v případě IPv6 *protokol NDP* (Neighbor Discovery Protocol). Základní funkce těchto protokolů jsou:

- vést tabulku sousedů,
- pomocí ARP/NDP dotazů tuto tabulku doplňovat.

V obou případech si zařízení vede *tabulku sousedů* s informacemi o nejbližším síťovém okolí (sousedech). V tabulce sousedů máme pro každého „souseda“ tyto informace:

- IP adresa (IPv4 nebo IPv6),
- MAC adresa,
- typ (statická/trvalá nebo dynamická/zjištěná).

První dva údaje nám mapují IP adresu souseda na jeho MAC adresu, třetí údaj určuje, jak se dotyčný údaj dostal do tabulky (zda byl staticky vložen nebo zjištěn dynamicky dotazováním).

Pro každé síťové rozhraní s vlastní IP adresou se vede jedna tabulka (včetně virtuálních rozhraní, pokud máme nainstalován virtualizační software).

Poznámka:

Koncová zařízení v síti mají v této tabulce především jeden důležitý záznam – adresu brány, na kterou směřují většinu svého provozu.

V případě Windows vyšších verzí máme ARP tabulku i NDP tabulku poněkud přeplácané – najdeme tam kromě brány taky multicast adresy a v případě ARP taky broadcast adresy včetně mapování na MAC adresy. Obsah tabulek zobrazíme takto:

- `arp -a` pro ARP tabulku v IPv4,
- `netsh interface ipv6 show neighbors` pro NDP tabulku v IPv6

(příkaz `netsh` se dá použít i pro zobrazení ARP tabulky v IPv4, pokud místo `ipv6` napíšete `ipv4`, ale kdo by se s tím vypisoval, když nemusí...).

Tyto příkazy se také dají použít k ovlivňování tabulky sousedů (přidávání nových záznamů nebo naopak jejich odstraňování), ve většině případů však tyto úlohy zvládají dotyčné protokoly dynamicky.

V Linuxu a jiných unixových systémech se pro zobrazení ARP/NDP tabulky používá příkaz

```
ip neigh show
```

(tedy můžeme napsat `ip neighbor show` nebo `ip n s` nebo něco mezi tím, nejlepší je volit kompromis – nepsat příliš a zároveň psát tolik, aby to bylo srozumitelné). Na výstupu máme obvykle pro každé síťové rozhraní právě jeden údaj – záznam pro bránu, nejsme zahlcováni dalšími informacemi. Také tento příkaz je možné použít pro ovlivňování obsahu tabulky.

Také v unixových systémech existuje příkaz `arp` (pro zobrazení tabulky stačí zadat bez parametrů), ten můžeme použít tehdy, když nám jde opravdu jen o IPv4 adresy.

6.2.2 K protokolům

Úkolem protokolů ARP a NDP není jen vést tabulku sousedů, ale také v případě potřeby se dynamicky poptávat na to, co by mělo v této tabulce být. Existují tedy pakety posílané dotyčným protokolem.

ARP pakety (tedy pro IPv4) se zapouzdřují přímo do rámců Ethernet II nebo rámců bezdrátové sítě Wi-fi. Rozlišujeme ARP dotaz a ARP odpověď.

Pokud zařízení potřebuje zjistit MAC adresu k určité IP adrese, vyšle ARP paket s dotazem „Who has xxx? Tell yyy.“ (Kdo má adresu xxx? Sdělte to na adresu yyy). Protože MAC adresáta neznáme (teprve ji zjišťujeme), dotaz se odesílá jako univerzální broadcast (tj. na adresu `FF-FF-FF-FF-FF-FF`).

Tážeme se na cizí IP adresu a chceme zaslat odpověď na naši IP adresu. Každý ARP paket je stejně dlouhý, ať už jde v kterémkoliv směru. V dotazu i odpovědi jsou všechna pole (vlastní IP a MAC adresa, dotazovaná IP a MAC adresa), ale v dotazu je to, co odesílatel neví, vyplněno nulami.

ARP dotaz (ve formě broadcastu, tedy pro všechny v síti) je zachycen kromě jiných i majitelem dotazované adresy, který doplní chybějící pole (tedy svou MAC adresu) a odešle zpět, tentokrát jako unicast (zná adresu tazatele).

NDP pakety (mechanismus pro IPv6) nemají vlastní specifikaci, posílají se jako ICMPv6 pakety se speciálními typy zpráv a zapouzdřují se do IPv6 paketů (jako každá ICMP zpráva). NDP je komplexnější protokol než ARP, plní více úloh. Co se týče ICMPv6 zpráv, se sousedy souvisejí především tyto:

- *Neighbor Solicitation* (ICMP zpráva číslo 135) – žádost o oznámení souseda, významem odpovídá ARP dotazu,
- *Neighbor Advertisement* (ICMP zpráva číslo 136) – odpověď na předchozí žádost.

Na rozdíl od ARP se formát a velikost dotazu a odpovědi liší, v dotazu najdeme opravdu jen žádanou adresu. V dotazu je jako adresa cíle použita multicast adresa odvozená od skupinové adresy pro směrovače. Odpověď je poslána buď jako unicast (pokud byla v dotazu IP adresa tazatele) nebo jako multicast na všechny uzly v síti (tj. na `ff02::1`).

Mechanismus NDP slouží i k jiným účelům – komunikaci s routery při zjišťování konfigurace sítě (adresa sítě, délka prefixu apod.), zjišťování duplicitních adres a dalším. Pro každý účel existují zvlášť čísla ICMPv6 zpráv.

Poznámka:

Otázkou je, na které vrstvě RM ISO/OSI, resp. síťového modelu TCP/IP, protokoly ARP a NDP vlastně pracují. V případě ARP není v standardu o nějaké vrstvě ani slovo. ARP pakety se zapouzdřují do rámců vrstvy L2, takže spíše patří na vrstvu L2 nebo na rozhraní mezi vrstvami L2 a L3, ale rozhodně ne na vrstvu L3.

Protokol NDP využívá ICMP zprávy, takže jednoznačně patří na vrstvu L3.

6.3 Adresní rozsah IPv4

6.3.1 Třídy

Adresa protokolu IPv4 zabírá 32 bitů (4 oktety), což by teoreticky znamenalo $2^{32} = 4\,294\,967\,296$ možných adres, ale ve skutečnosti je to mnohem méně – kromě jiného z „organizačních“ důvodů. Adresní rozsah je hierarchicky členěn na sítě a podsítě, aby se zjednodušilo směrování.

Původně se právě z důvodu snadnějšího škálování této hierarchie rozlišovalo *pět tříd adres* podle pozice hranice mezi síťovou a hostitelskou částí adresy:

- *Třída A* používá první oktet pro adresu sítě, zbytek je hostitelská část, adresa s prvním oktetem nulovým není platná,
- *Třída B* používá pro adresu sítě první dva oktety, zbytek je hostitelská část,
- *Třída C* používá pro adresu sítě první tři oktety, zbytek je hostitelská část,
- *Třída D* slouží pro skupinovou adresaci,
- *Třída E* slouží pro experimentální účely.

Struktura adres těchto tříd je naznačena v tabulce 6.1.

Poznámka:

Takže když vidíme adresu, jejíž první oktet je menší než 128, pak by nám mělo být jasné, že jde o adresu třídy A. Jestliže je první oktet menší než 192 (a větší nebo roven 128), jde o adresu třídy B a u prvního oktetu menšího než 224 adresu třídy C. Číslo 224 bychom si měli pamatovat obzvlášť dobře, protože tímto oktetem začíná většina skupinových adres.

Třída	Struktura adresy	První nibble	První oktet		Max. zařízení v síti
A	síť.uzel.uzel.uzel	`0xxx...`	1–127	1–7F	$2^{24} - 2$
B	síť.síť.uzel.uzel	`10xx...`	128–191	80–BF	$2^{16} - 2$
C	síť.síť.síť.uzel	`110x...`	192–223	C0–DF	$2^8 - 2 = 254$
D	skupinové	`1110...`	224–239	E0–EF	–
E	experimentální	`1111...`	240–255	F0–FF	–

Tabulka 6.1: Třídy IPv4 adres

Kdy kterou třídu volíme? To záleží na tom, jakým způsobem chceme členit síť do podsítí (k tomu se dostaneme později) a kolik adres pro zařízení v dané síti budeme potřebovat. Jestliže budeme mít v síti stovky zařízení, pak samozřejmě nezvolíme třídu C. Většinou se volí třída C pro velmi malé sítě a třída A pro větší sítě (nebo když počítáme s překotným růstem sítě).

Poznámka:

Všimněte si v tabulce 6.1, že v posledním sloupečku (Maximální počet zařízení v síti) je u tříd „použitelných“ pro zařízení vždy určité číslo −2. Proč? Protože odečítáme adresu, kde jsou všechny bity v hostitelské části nastaveny na 0 (tj. adresu sítě) a adresu, kde jsou všechny hostitelské bity na staveny na 1 (tj. broadcast adresu v síti). Tyto dvě adresy nemohou být přiděleny žádnému zařízení, proto jsou odečteny.[1]

Vraťme se ke speciálním adresám pro IPv4:

- Loopback (tj. `127.0.0.x`) je zjevně adresou třídy A.
- V každé třídě máme část rozsahu vyhrazenu pro soukromé adresy:
  - pro třídu A to jsou adresy `10.x.x.x`,
  - pro třídu B to jsou adresy `172.16.x.x` až `172.31.x.x`,
  - pro třídu C to jsou adresy `192.168.0.x` až `192.168.255.x`.

  Všimněte si, že tyto adresy opravdu patří do rozsahu pro danou třídu. Pokud máme nastaveno používání adres určité třídy, jsme odkázáni na soukromé adresy s touto třídou související.
- Broadcast pro síť s adresami dané třídy (A, B nebo C) má síťovou část nastavenou běžným způsobem, v hostitelské části (tj. poslední tři, dva nebo jeden oktet) jsou všechny bity nastaveny na 1.

  Takže pokud máme adresu sítě například `10.0.0.0`, bude broadcastovou adresou v dané síti adresa `10.255.255.255`.

Poznámka:

Všimněte si, že pokud bychom zůstali u tříd a třídního směrování, pak vlastně nepotřebujeme masku sítě ani délku prefixu. To, která část adresy je síťová a která hostitelská, nám zcela jednoznačně určuje prvních pár bitů prvního oktetu. Jenže my jsme u tříd nezůstali, a tedy budeme tyto „dodatečné údaje“ potřebovat.

[1] Ve skutečnosti to u některých routerů jde (i u Cisca), ale je to považováno za nestandardní operaci, která někdy může způsobit problémy.

6.3.2 Podsíťování

Jak jsme výše naznačili, třídy postupně přestaly stačit. Správci větších sítí prostě potřebovali vytvářet hierarchii, a to jen se třídami nešlo. Proto se začalo používat *podsíťování* (subnetting). Jednu síť rozdělili na několik dílů – podsítí (subnetů), přičemž příslušnost k určité podsíti se dá poznat podle konkrétních bitů adresy.

Podsíťování není nic jiného než administrativní zásah do hostitelské části adresy (s adresou sítě a jejím rozsahem se u subnettingu nic neprovádí). Takže bity v hostitelské části nejsou všechny vyhrazeny pro adresování stanice, ale část (zleva, hned za síťovou částí) vyhradíme pro adresu podsítě.

Takže adresa se při podsíťování skládá ze tří částí:

- síťová část adresy,
- podsíťová část adresy,
- hostitelská část adresy.

Součet bitů všech tří částí samozřejmě musí dávat 32. Zatímco počet bitů síťové části je jednoznačný (zatím jsme pořád u tříd), počet bitů podsíťové části už jednoznačný není, a tedy je nutné k adrese přidat masku podsítě (nebo délku prefixu).

Příklad

Pokud máme adresu sítě třídy C ve tvaru `192.168.48.0` (tj. tři oktety jsou adresou sítě) a chceme tuto síť rozdělit na podsítě, potřebujeme vědět především:

- kolik má být podsítí,
- kolik maximálně stanic v každé podsíti chceme mít.

Oba údaje navzájem souvisejí – když chceme víc podsítí, budeme muset vyhradit víc bitů pro podsíťovou část adresy a zůstane nám méně bitů v hostitelské části, a tedy méně stanic v každé podsíti.

Takže předpokládejme, že podsítí nechceme nijak moc – vystačíme si se dvěma bity (tj. maximálně $2^2 = 4$ podsítě).

Zatím jsme vypotřebovali $3 * 8 + 2 = 26$ bitů pro síťovou a podsíťovou část adresy, tedy pro hostitele nám zbývá $32 - 26 = 6$ bitů. Protože $2^6 - 2 = 62$, v každé podsíti může být 62 zařízení (pozor, jakýchkoliv zařízení včetně routeru). Rozvržení bitů v adrese – část pro síť, podsíť a hostitele:

`ssssssss.ssssssss.ssssssss.ppHHHHHH`

Jak tedy budou vypadat adresy jednotlivých podsítí: část adresy pro podsíť bude u různých podsítí nabývat hodnot binárně `00`, `01`, `10` a `11`. Pro každou podsíť stanovíme adresu sítě (bity v hostitelské části = 0) a broadcast adresu (bity v hostitelské části = 1), adresy mezi nimi pak budou patřit zařízením. V reálu:

- Podsíť 1: první dva bity posledního oktetu budou 00, tedy:
  - adresa podsítě je `192.168.48.0/26` (poslední oktet binárně `00000000`),
  - broadcast adresa pro tuto podsíť je `192.168.48.63/26` (poslední oktet `00111111`),
  - hostitelé mají adresy z rozsahu `192.168.48.1/26` až `192.168.48.62/26`.

- Podsíť 2: první dva bity posledního oktetu budou 01, tedy:
  - adresa podsítě je `192.168.48.64/26` (poslední oktet binárně `01000000`),
  - broadcast adresa pro tuto podsíť je `192.168.48.127/26` (poslední oktet `01111111`),
  - hostitelé mají adresy z rozsahu `192.168.48.65/26` až `192.168.48.126/26`.
- Podsíť 3: první dva bity posledního oktetu budou 10, tedy:
  - adresa podsítě je `192.168.48.128/26` (poslední oktet binárně `10000000`),
  - broadcast adresa pro tuto podsíť je `192.168.48.191/26` (poslední oktet `10111111`),
  - hostitelé mají adresy z rozsahu `192.168.48.129/26` až `192.168.48.190/26`.
- Podsíť 4: první dva bity posledního oktetu budou 11, tedy:
  - adresa podsítě je `192.168.48.192/26` (poslední oktet binárně `11000000`),
  - broadcast adresa pro tuto podsíť je `192.168.48.255/26` (poslední oktet `11111111`),
  - hostitelé mají adresy z rozsahu `192.168.48.193/26` až `192.168.48.254/26`.

Všimněte si, že adresa podsítě je o 1 vyšší než broadcastová adresa předchozí podsítě.

Proč to všechno?

- Podsítě nám zjednodušují směrování v rámci velké sítě (ve směrovacích tabulkách stačí mít pro celou jednu podsíť jediný řádek).
- Optimalizujeme tím síťový provoz, což je důsledkem zjednodušení směrování.
- Broadcastové domény můžeme omezovat i na podsíť, takže broadcasty nám nebloudí v celé síti, což opět znamená snížení provozu v síti.

Poznámka:

V třetí kapitole jsme se seznámili s VLAN (virtuálními lokálními sítěmi), což je obvykle chápáno jako mechanismus vrstvy L2. Ve skutečnosti se členění sítě na VLANy distribuuje i na vrstvu L3, a to právě pomocí podsítí – zařízení patřící do stejné VLANy budou v téže podsíti, zařízení z různých VLAN budou v různých podsítích.

6.3.3 VLSM

Masky podsítí s proměnnou délkou – VLSM (Variable Length Subnet Mask) – jsou mechanismem rozšiřujícím možnost podsíťování. Zatímco u podsíťování jsme si pevně stanovili, kolik bitů bude pro adresu podsítě, u VLSM se tento počet bitů může u různých podsítí měnit. Ještě pořád zůstáváme u třídního směrování, ale zvyšujeme pružnost návrhu.

K čemu je to dobré? Vpodstatě navyšujeme výhody, které přináší možnost podsíťování. Při podsíťování máme pro každou podsíť tentýž maximální počet hostitelů, při použití VLSM můžeme mít pro různé podsítě různý počet hostitelů, podle potřeby. Ale pozor, jednoznačnost je důležitá i zde, takže jednotlivé podsítě (jejich adresní rozsahy) se v žádném případě nesmí prolínat.

Poznámka:

Můžeme si to představit tak, že si vytvoříme vícestupňovou hierarchii podsítí, přičemž v některých „větvích“ zastavíme členění dřív (méně bitů pro podsíť, více bitů pro hostitele) a v jiných „větvích“ zastavíme členění později (více bitů pro podsíť a tedy více podsítí, méně bitů pro hostitele).

Příklad

Opět máme adresu sítě třídy C ve tvaru `192.168.48.0` a chceme tuto síť rozdělit na podsítě. Víme, že budeme potřebovat jednu velkou podsíť a dvě menší. Pro první podsíť si zvolíme délku prefixu 25 (tedy pro podsíť vyhradíme jeden bit), pro druhou a třetí 26 (dva bity pro podsíť).

- Podsíť 1: první bit posledního oktetu bude 0, tedy:
  - adresa podsítě je `192.168.48.0/25` (poslední oktet binárně `0`0000000),
  - broadcast adresa pro tuto podsíť je `192.168.48.127/25` (poslední oktet `0`1111111),
  - hostitelé mají adresy z rozsahu `192.168.48.1/25` až `192.168.48.126/25`.
- Podsíť 2: první dva bity posledního oktetu budou 10, tedy:
  - adresa podsítě je `192.168.48.128/26` (poslední oktet binárně `10`000000),
  - broadcast adresa pro tuto podsíť je `192.168.48.191/26` (poslední oktet `10`111111),
  - hostitelé mají adresy z rozsahu `192.168.48.129/26` až `192.168.48.190/26`.
- Podsíť 3: první dva bity posledního oktetu budou 11, tedy:
  - adresa podsítě je `192.168.48.192/26` (poslední oktet binárně `11`000000),
  - broadcast adresa pro tuto podsíť je `192.168.48.255/26` (poslední oktet `11`111111),
  - hostitelé mají adresy z rozsahu `192.168.48.193/26` až `192.168.48.254/26`.

Srovnejte s předchozím příkladem. Všimněte si, že v tomto příkladu nám první podsíť vlastně „spolkla“ první a druhou podsíť z předchozího příkladu (tj. obě původní podsítě začínající bitem `0` se staly jedinou podsítí). Jaký je důsledek? Zatímco v „menších“ podsítích máme k dispozici 62 adres pro zařízení ($2^5 - 2$), ve „velké“ podsíti je 126 adres pro zařízení ($2^6 - 2$). Kdybychom používali adresy třídy A, měli bychom ještě větší možnosti rozlišení.

Poznámka:

Na cvičeních se podíváme podrobněji na podsíťování i VLSM, včetně běžných postupů. Příklad na podsíťování nebo VLSM bude důležitou součástí zápočtové písemky.

6.3.4 CIDR

Beztřídní směrování – CIDR (Classless Inter-Domain Routing) již třídy plně odbourává v tom smyslu, že ruší závislost mezi tvarem prvního oktetu a nejlevější hranicí pro síťovou část adresy (což právě stanovují třídy). Hlavním motivem bylo zpřehlednění a zjednodušení směrování ve vyšších úrovních hierarchie, tedy u poskytovatelů síťového připojení na úrovni RIR a LIR. Dále budeme používat terminologii, se kterou jsme se seznámili v sekci 4.3 od strany 85.

Jak to funguje s CIDR: IANA přiděluje základní rozsahy jednotlivým RIR, tedy určitý počet bitů v adrese zleva. Tím je dána určitá základní délka prefixu. Každý RIR svým zákazníkům (tedy jednotlivým LIR) přidělí rozsah adres takový, že LIR „zdědí“ to, co má přiděleno RIR (tedy tuto část mají všichni zákazníci společnou), a k tomu jsou přidány další bity specifické pro dotyčného LIR – provede se podsíťování, každý LIR bude mít (vzhledem k RIR) jednu podsíť. Takže LIR má adresu s delším prefixem. LIR má také své zákazníky, kterým provede totéž – každý zákazník „zdědí“ prefix od LIR plus část specifickou pro daného zákazníka.

Z toho vyplývá, že CIDR vlastně buduje hierarchii sítí a podsítí navzájem vnořených do takové míry, jak je zapotřebí. Hlavním účelem CIDR je právě sladit hierarchii IP adres s hierarchií (fyzických) sítí a především routerů.

Poznámka:

Jak se CIDR projevuje na routerech: vraťme se k předchozímu příkladu. Na routeru, od kterého vede společná cesta k podsítím číslo 2 a 3, můžeme mít pro obě podsítě společný záznam ve směrovací tabulce – jednoduše pro adresu `192.168.48.128/25` (poslední oktet binárně `10000000`), protože prvních 25 bitů adresy mají pro obě podsítě shodných a až v dalším bitu se liší.

Ve skutečnosti router, který takto zjednodušeně adresuje podsítě 2 a 3, tyto dvě podsítě ani nerozlišuje (netuší, že rozsah `192.168.48.128/25` je nějak vnitřně členěn). Ale to je dobře, vnitřním členěním není nutné zbytečně zahlcovat vnější prvky.

Sumarizace (také agregace) cest, resp. *nadsíťování* (supernetting) je právě to, co je popsáno v poznámce. Pokud do více podsítí vede cesta přes tentýž port routeru, není důvod mít pro každou zvlášť ve směrovací tabulce samostatný řádek. Všechny budou shrnuty do jednoho řádku, přičemž se zkrátí prefix (posune se hranice síťové části adresy směrem doleva) tak, aby v sobě zahrnoval všechny tyto podsítě. Klidně i před původní hranici tříd A/B/C. Tomuto souhrnnému řádku reprezentujícímu více (pod)sítí se říká *CIDR blok*. Ovšem podmínkou je, aby doopravdy fyzické propojení sítí odpovídalo hierarchii IP adres.

Poznámka:

Rozdíl mezi podsíťováním (včetně VLSM) a nadsíťováním (CIDR) je kromě jiného v tom, že u podsíťování posouváme hranici mezi (pod)síťovou a hostitelskou částí směrem doprava, kdežto u nadsíťování ji posouváme doleva. Podsíťování se týká každého prvku hierarchie (firma dostane od ISP adresní rozsah a ten si může libovolně rozčlenit), nadsíťování se týká spíše ISP provozujících hodně vytížené směrovače (potřebují redukovat směrovací tabulky právě pomocí CIDR).

Další informace:

Zajímavým projektem, především pro výukové účely, je Subnet Calculator:
http://www.subnet-calculator.com/.

Právě systém CIDR přišel se zjednodušeným označováním hranice mezi síťovou a hostitelskou částí pomocí lomítka a délky prefixu, tedy v třídním směrování bychom teoreticky měli používat zápis pomocí masky. Zápisu s lomítkem a délkou prefixu se říká *CIDR notace*.

Poznámka:

Systém adresních rozsahů IPv6 je již přirozeně typu CIDR. Právě proto jsme s popisem hierarchie pro IANA, RIR, LIR apod. začali právě u popisu principu IPv6 adresování.

6.4 Jak získat IP adresu

Každé zařízení komunikující po síti potřebuje minimálně jednu IP adresu, jenže ji potřebuje odněkud získat. Mechanismy získávání IP adresy jsme ve skutečnosti už nakousli (v předchozí kapitole byla sekce o DHCP), teď si projdeme jednotlivé možnosti.

6.4.1 Jak získat IPv4 adresu

IPv4 adresu můžeme získat

- dynamickou alokací podle protokolu DHCP,
- staticky, přičemž buď adresu zadáváme do konfigurace ručně nebo použijeme automatickou statickou alokaci (také pomocí DHCP).

Statická IP adresa je tedy adresa pevně určená pro dané zařízení, zařízení jinou ani nedostává. Typicky se statické IP adresy používají pro servery, které mají být dostupné vždy pod stejnou adresou, aby se nemusely průběžně měnit směrovací tabulky. Naproti tomu *dynamická IP adresa* je vždy propůjčena (leased) jen na určitou dobu (třeba jeden den) a po této době může zařízení získat znovu tutéž adresu nebo úplně jinou.

DHCP server je server zajišťující přidělování a evidování dynamických IP adres a taky adres přidělovaných statickou alokací. Má přidělen rozsah adres, které může přidělovat (Address Pool, Address Stack) a v tomto rozsahu si poznamenává, komu (které MAC adrese) zatím přidělil kterou IP adresu a na jak dlouho (leased time). Tato tabulka je vlastně stavovou informací DHCP serveru (určuje stav přidělování adres), a tedy využití DHCP serveru pro přidělování adres označujeme jako *stavový režim DHCP*.

Co se týče *statické alokace*, probíhá tak, že na DHCP serveru máme nadefinováno přiřazení konkrétní IP adresy ke konkrétní MAC adrese. Jestliže zařízení s takovou MAC adresou žádá DHCP server o IP adresu, je mu přidělena vždy tatáž.

O protokolu DHCP a způsobu jeho používání je již psáno v předchozí kapitole o aplikačních protokolech, včetně DHCP komunikace. Je třeba si uvědomit, že DHCP zdaleka není jen o adresách, zprostředkovává taky další informace, například adresy DNS serverů.

6.4.2 Jak získat IPv6 adresu

V síti se zprovozněným IPv6 každý router v pravidelných intervalech rozesílá ICMPv6 zprávu *Router Advertisement – RA* (oznámení směrovače). Router (nebo switch s funkcionalitou L3) do těchto zpráv vkládá všechny důležité informace o sobě a o síti, například síťový prefix (tedy adresa sítě), délka prefixu, adresa brány, hodnota MTU apod.

Pokud zařízení chce doprovodné údaje k IPv6 adrese, buď naslouchá na síti nebo se aktivně zeptá ICMPv6 zprávou *Router Solicitation – RS* (dotaz na oznámení směrovače), kterou donutí router vyslat RA.

Poznámka:

Zprávy RA (ICMP zpráva 134) a RS (ICMP zpráva 133) jsou součástí mechanismu protokolu NDP (objevování sousedů).

Podle IPv6 máme tyto možnosti získání IP adresy:

- dynamickou alokací pomocí DHCP (stavový režim DHCP),
- autokonfigurací stanice s využitím EUI-64 (SLAAC – Stateless Address Autoconfiguration),
- autokonfigurace s Privacy Extensions,
- statická konfigurace.

První možnost – *dynamická alokace pomocí DHCP* – je prakticky stejná jako u IPv4 (rozdíl je především v typu a formátu zasílaných zpráv). Jako jediná je *stavová*, tedy DHCP server si ukládá stavové (proměnlivé, dynamické) informace o přidělování adres, ostatní možnosti jsou bezstavové (nikam se takové záznamy neukládají).

Autokonfigurace s využitím EUI-64 (SLAAC) má ulehčit DHCP serverům a zejména síti od nadbytečného provozu. Hostitelskou část adresy si zařízení sestaví samo, jen musí získat ostatní informace (síťovou část adresy apod.). Jak to probíhá:

- Hostitelskou část adresy (zde přesně polovinu, 64 bitů) tvoří EUI-64, které se vyrobí z MAC adresy. MAC adresa má 48 bitů, takže ještě musíme 16 bitů přidat (2 oktety). Upravíme:
  - přesně do poloviny MAC adresy vnoříme dva oktety `FF-FE`,
  - nastavíme U/L bit na 1 (v prvním oktetu druhý bit zprava).
- Síťovou část adresy (zbývajících 64 bitů) musíme získat jinak, například z RA zprávy routeru.

Příklad

Ukážeme si vytvoření EUI-64 a jeho začlenění do IPv6 adresy. Předpokládejme tyto údaje:

- MAC adresa je `50-E5-49-C2-AC-27`,
- přes RA jsme zjistili, že prefix (síťová část adresy) je `2001:718:2601:265`.

Nejdřív vytvoříme EUI-64. MAC adresa se skládá z šesti oktetů, tedy mezi třetí a čtvrtý (tj. doprostřed) vnoříme dva oktety `FF-FE`:

`50-E5-49-FF-FE-C2-AC-27`

Pak v prvním oktetu nastavíme U/L bit – binárně je první oktet původně `01010000`, po změně `01010010`, což je hexadecimálně `52`. Takže výsledné EUI-64 je:

`52-E5-49-FF-FE-C2-AC-27`

Zkompletujeme celou adresu, tedy EUI-64 upravíme, aby „opticky“ odpovídalo části IPv6 adresy (skupiny, dvojtečky) a předsadíme prefix.

`2001:718:2601:265:52E5:49FF:FEC2:AC27/64`

Jenže to ještě není všechno. Velkou výhodou sice je, že stanice nemusí kvůli získání adresy komunikovat s DHCP serverem (jen musíme mít v síti takový router, který dokáže v RA odesílat síťovou část adresy), ale jak se stanice dozví například IP adresy DNS serverů? Teoreticky je víc možností:

- *Přidat info o adresách DNS serverů do RA.* Tato možnost je standardizována od roku 2010 v RFC 6106 a podporují ji skoro všechny systémy (u klientských zařízení prakticky všechny běžné unixové systémy včetně Linuxu, Android, systémy od Applu), kromě těch od Microsoftu (podpora musí být jak na straně routeru, tak na straně žadatele). Očekávalo se, že RFC 6106 bude podporováno ve Windows 10 a Windows Server 2016, ale není.

- *Speciální anycast adresy pro DNS servery.* Standardizace nebyla nikdy dokončena, protože se počítalo se site-local adresami, které byly ze standardů o IPv6 vypuštěny. Implementaci můžeme najít v některých systémech od Microsoftu.
- *Použití DHCPv6.* Adresu si sice stanice zkompletuje přes EUI-64 a pomocí RA, ale adresy DNS serverů si zjistí od DHCP serveru. Toto řešení vyžaduje, aby v síti byl alespoň jeden DHCP server, třebaže nebude fungovat stavově a provoz na síti nebude tak velký jako při dynamické alokaci.

Autokonfigurace s Privacy Extensions je třetí možností získání IPv6 adresy. Funguje to vpodstatě podobně jako autokonfigurace s EUI-64, jen hostitelskou část adresy vytváříme jinak.

- Hostitelská část adresy (taky přesně polovina, 64 bitů) je dynamicky generována z různých hardwarových a softwarových charakteristik systému a každých několik dnů se mění.
- Síťová část adresy je získávána například z RA zprávy routeru.

Pro získání adres DNS serverů platí totéž co u EUI-64. Zatímco EUI-64 se používá především v unixových systémech, Privacy Extensions jsou typické pro Windows.

Poznámka:

Účelem je co nejvíc ztížit identifikaci koncových zařízení v síti, ale ve skutečnosti tento postup ztěžuje práci i správcům. Správce sítě potřebuje mít přehled o tom, co se děje v síti, jaká zařízení má momentálně připojená a kam se obvykle připojují, IP adresa je důležitým identifikátorem v monitorování a evidenci. Pokud se tento identifikátor často mění, způsobuje to chaos v síti.

Statická konfigurace je poslední možností získání adresy, znamená situaci, kdy zařízení dostane adresu předem přidělenou. Stejně jako u IPv4, i zde ji lze provést ručně nebo zajistit pomocí statické alokace (zařízení se dotáže DHCP serveru, ale získává vždy tutéž IP adresu).

Staticky lze konfigurovat buď celou adresu nebo jen její hostitelskou část, přičemž síťovou část si zařízení doplní stejně jak bylo popsáno u předchozích možností.

Poznámka:

Statická alokace je *bezstavové využití DHCP*. Sice musí být uložena informace o tom, kterou IP adresu je třeba dotyčnému zařízení s určitou MAC adresou přidělit, ale nejedná se o dynamickou (proměnlivou) informaci.

6.5 NAT a soukromé adresy

O *soukromých adresách* už něco víme – jsou pro ně v každé třídě vyhrazeny určité rozsahy a pakety s touto adresou nesmí „za router“. K čemu tedy jsou?

Jejich hlavním účelem je šetření rozsahů veřejných IP adres (většina koncových zařízení nepotřebuje veřejnou IP adresu, stačí soukromá, která není globálně unikátní). Druhým účelem je mírné zvýšení zabezpečení, protože soukromé adresy je třeba na hranici sítě překládat na veřejné (nebo na soukromé fungující v nadřízené síti), což může být chápáno jako dodatečný ochranný mechanismus.

Soukromé adresy je třeba překládat na hranici sítě, ale ve skutečnosti tento překlad můžeme využívat i u jiných typů adres. Překlad v základu spočívá v tom, že v paketu zaměníme jednu IP adresu za jinou.

Poznámka:

Abychom si nepletli pojmy:

- veřejná adresa je globálně platná, může do Internetu (resp. za router),
- soukromá adresa je pouze lokálně platná, nemůže za router.

Dále rozlišujeme staticky a dynamicky přidělenou adresu:

- statická adresa je pevně určena danému zařízení, jiné zařízení ji nemůže získat,
- dynamická adresa je momentálně danému zařízení přidělena, ale zítra ji může mít úplně jiné zařízení.

Adresa zařízení může být soukromá a zároveň dynamická nebo soukromá a zároveň statická. Veřejné adresy bývají statické, ale ne každá statická adresa je veřejná.

NAT (Network Address Translation) je mechanismus překladu IP adres na hranici sítě. V paketu můžeme překládat jen zdrojovou adresu nebo jen cílovou adresu nebo obojí, obvykle se kombinuje překlad zdrojové adresy v jednom směru a překlad cílové adresy v druhém směru (když se nad tím zamyslíme, zjistíme, že se vlastně jedná o adresu příslušející témuž zařízení).

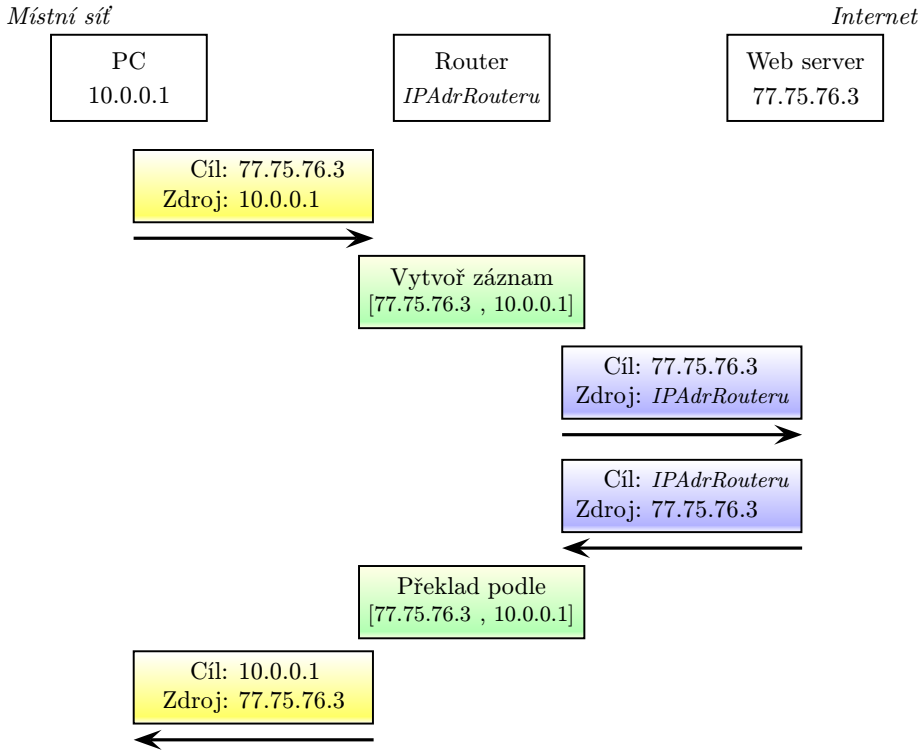
Rozlišujeme tyto druhy NAT:

- Statický NAT (bezstavový) – je pevně určeno, kterou adresu z vnitřní sítě překládáme na kterou adresu platnou ve vnější síti a naopak.
- Dynamický NAT (stavový, Masquerade) – dynamicky se určuje dvojice adres pro překlad, vyžaduje uložení záznamu o překladu, který se využije pro zpětný překlad.
- PAT (Port Address Translation, přetížení NAT) – přidává k dynamickému NAT dodatečný mechanismus rozlišení jednotlivých konverzací.

Statický NAT znamená, že v odchozím provozu zaměníme v paketu zdrojovou IP adresu našeho zařízení za jinou, pod kterou má být toto zařízení viditelné „venku“, a naopak v příchozím provozu provedeme na (tentokrát cílovou) IP adresu tohoto zařízení zpětný překlad. Statický NAT vyžaduje, abychom měli pro dané zařízení staticky vyhrazeny dvě adresy – jednu pro vnitřní síť a druhou pro vnější síť.

Typicky se používá pro skrývání IP adres našich serverů. Server je „venku“ viditelný pod jinou IP adresou než jakou má ve skutečnosti. Dalším důvodem využití statického NATu je dočasná změna v síti třeba z důvodu údržby.

Dynamický NAT (Masquerade, českým patvarem „maškaráda“) je již stavová konfigurace. Používá se v případě, že vnitřní adresy, které překládáme, jsou dynamické, a tedy si je nelze „staticky zapamatovat“. Abychom ušetřili adresní rozsah, máme ve vnitřní síti dynamicky přidělované adresy a směrem ven je překládáme na jedinou adresu (společnou pro všechno dynamické z vnitřní sítě). NAT server (to bývá router provádějící NAT překlad) si vede tabulku spojení, ve které si pro každé spojení mezi vnitřní a vnější sítí zaznamenává adresu zařízení ve vnitřní síti a adresu



Obrázek 6.2: Zjednodušené schéma dynamického NAT

zařízení z vnější sítě (typicky některého serveru na Internetu, se kterým je spojení navazováno). Princip je naznačen na obrázku 6.2.

Jednoznačnost je zajištěna v případě, že každé zařízení z vnitřní sítě komunikuje s jiným serverem. Problém nastává tehdy, když víc zařízení z vnitřní sítě chce komunikovat s tímtéž „vnějším“ serverem, pak by záznamy v tabulce spojení přestaly být jednoznačnými. Ve směru od klienta k serveru by se paket poslat dal, ale odpověď od serveru bychom nemohli doručit příslušnému klientovi, protože bychom nevěděli, kterému.

PAT (překlad portů, přetížený NAT) řeší právě tento problém. Do IP paketů se obvykle zapouzdřují TCP nebo UDP segmenty, ve kterých používáme zdrojové a cílové číslo portu. Klientská čísla portů bývají dynamická, tj. je velká pravděpodobnost, že dva různí klienti z naší sítě komunikující s tímtéž serverem budou mít tato čísla odlišná.

Takže místo abychom do tabulky ukládali jen IP adresy, přidáme i čísla portů (připomeňme si, že například socket je identifikován dvojicí IP adresa + číslo portu). Pokud by přesto nastala situace, že by dva klienti z vnitřní sítě při komunikaci s tímtéž serverem použili stejné zdrojové číslo portu, na NAT serveru by byla přeložena nejen IP adresa, ale i číslo portu tak, abychom zajistili jednoznačnost.

Další informace:

Mechanismus NAT je obvykle chápán jako berlička pro IPv4, ale i v IPv6 pro něj mohou existovat důvody: http://www.lupa.cz/clanky/10-duvodu-proc-mit-nat-na-ipv6/

Poznámka:

Shrňme si dosud probírané mechanismy pro překlad adres:

- protokoly ARP a NDP provádějí překlad logické IP adresy na fyzickou MAC adresu,
- protokol DNS provádí překlad jmenné (doménové) adresy na IP adresu,
- mechanismus NAT překládá mezi vnitřními a vnějšími IP adresami.

6.6 Směrování

6.6.1 Jak směrujeme

Tématu směrování (routing) jsme se už párkrát dotkli, teď se na ně zaměříme podrobněji. Směrování probíhá na vrstvě L3 a vlastně znamená proces určení cesty pro IP paket (tedy přes který hardwarový port tento paket poslat).

Definice (Směrovací tabulka)

Na routeru (nebo jiném zařízení vrstvy L3) je vedena minimálně jedna *směrovací tabulka* (routing table), ve které máme pro každý záznam především tyto informace:

- adresa sítě (plus maska nebo délka prefixu),
- určení síťového rozhraní pro odeslání,
- brána (v případě, že nejde o vlastní podsíť),
- metrika (kvalita určené cesty), případně další informace.

Tuto tabulku je třeba nějak naplnit a dále aktualizovat podle momentální situace (některá cesta přestane být platná, nebo naopak jiná cesta je zpřístupněna, mění se metriky či přidáváme novou podsíť). To se buď provádí ručně (staticky) nebo to můžeme nechat na *směrovacích protokolech* (dynamická údržba).

Z toho vyplývá, že směrování může být

- statické (static routing) – správce sítě ručně vkládá záznamy do směrovacích tabulek,
- dynamické (dynamic routing) – na směrovačích v síti běží směrovací protokoly, které si udržují přehled o topologii sítě (jak je co propojeno) a aktualizují směrovací tabulky.

V praxi se obojí kombinuje, tedy některé (blízké či nějakým způsobem citlivé) cesty vkládáme ručně a zbylé necháme doplňovat a aktualizovat dynamicky.

Metrika (kvalita cesty) je číslo, které bylo vypočteno podle určitých kritérií. Každý směrovací protokol používá trochu jiná kritéria a taky trochu jiným způsobem z nich metriku počítá. Jako kritéria se například mohou použít:

- propustnost cesty (například zda jde o Fast Ethernet nebo Gigabit Ethernet),
- počet routerů na cestě,
- spolehlivost cesty (nízká chybovost, málo zahozených paketů apod.),
- vytíženost cesty, atd.

Poznámka:

Obvykle platí, že čím menší číslo metriky, tím lepší cesta. Přímo připojené adresy mívají nízkou metriku.

Autonomní systém (podle jiných protokolů *proces*) je skupina směrovačů patřících pod správu téže organizace. Každý autonomní systém má přiděleno své číslo.

- *Vnitřní směrovací protokoly* (interior routing protocols) dokážou směrovat jen uvnitř vlastního autonomního systému (cestu ven neznají, všechno „cizí“ posílají prostě na jedinou bránu), nerozlišují jiné autonomní systémy (rozlišují jen „vlastní“ a „cizí“).
- *Vnější směrovací protokoly* (exterior routing protocols) zvládají směrování i mezi různými autonomními systémy (dokážou pracovat s číslem autonomního systému, ve směrovací tabulce máme v každém záznamu i číslo směrovací oblasti), dokážou rozlišovat mezi různými „cizími“ cestami.

Není nutné mít vlastní autonomní systém (ostatně, není to zadarmo). Domácí sítě a sítě malých firem jsou součástí autonomního systému svého ISP.

Každý směrovací protokol se řídí nějakým *směrovacím algoritmem*. Obvykle se jedná o jeden z následujících dvou algoritmů (případně po určitých úpravách) nebo jejich kombinaci:

- algoritmus vektoru vzdáleností,
- algoritmus stavu spoje.

6.6.2 Směrování algoritmem vektoru vzdáleností

Algoritmus vektoru vzdáleností (Distance-Vector Algorithm) určuje metriku cesty především podle vzdálenosti (počtu routerů na cestě). Router pracující podle tohoto algoritmu (protokolu) má nejdřív v tabulce své sousedy (routery, ke kterým vede přímá cesta) a od nich postupně dostává informace o jiných routerech a jejich podsítích. Pokud od svého souseda dostane paket s informací o síti s konkrétní metrikou (podle toho souseda), přidá si tuto síť do směrovací tabulky:

- adresu (pod)sítě zjistil od souseda,
- rozhraním bude adresa či port dotyčného souseda (protože přes něj je ta síť dostupná),
- jako metriku použije číslo o 1 (nebo jinou malou konstantu) větší než jaká platí pro souseda (tj. k cestě od souseda k síti je třeba započítat i úsek od souseda ke mně).

Každý router v pravidelných intervalech posílá svou směrovací tabulku všem sousedům, čímž je informuje o aktuálně dosažitelných sítích a jejich metrikách, což může poměrně hodně zahltit síť.

Jak vidíme, může tady vzniknout podobný problém jako v síti switchů, kdy jsme museli pro odstranění smyček použít protokol STP. Smyčky mohou vzniknout i zde (máme síť routerů používajících daných algoritmus/protokol), přičemž důsledky by mohly být podobné. Proto je definován maximální počet přeskoků, který určuje maximální velikost sítě (v počtu routerů na cestě). Pokud od souseda dojde informace o síti, která je dosažitelná více přeskoky než jaký je maximální počet, pak je tento záznam ignorován (síť je považována za nedosažitelnou).

Typickým směrovacím protokolem používajícím algoritmus vektoru vzdáleností je RIP (Routing Information Protocol) používající jako metriku počet skoků k cíli (tj. počet routerů na cestě).

První verze RIPv1 je v současné době nepoužitelná (pouze třídní směrování, nepodporuje masky ani délky prefixů, počítá jen s velmi malými sítěmi, posílá se jako broadcast, pouze IPv4).

RIPv2 v aktualizačních paketech posílá záznamy o sítích včetně masky, a tedy podporuje beztřídní směrování. Dnes používá jen v malých sítích, protože na jednu stranu je vpodstatě jednoduchý, ale na druhou stranu velmi pomalu konverguje (aktualizace sítě je časově náročná, během ní se nesměruje) a standardně povoluje maximální počet přeskoků jen 15.

Další verzí je RIPng, což je vlastně RIPv2 s přidanou podporou IPv6 adres.

Poznámka:

Typickou vlastností protokolů implementujících algoritmus vektoru vzdáleností je, že přímo „znají“ pouze své sousedy a o topologii sítě nemají žádnou informaci. Pouze vědí, že ke konkrétní (pod)síti se dá dostat přes konkrétního souseda a cesta trvá konkrétní počet přeskoků.

Dalším protokolem využívajícím algoritmus vektoru vzdáleností je IGRP od společnosti Cisco (proprietární) a jeho následovník EIGRP (jeho specifikace byla zveřejněna). IGRP se už nepoužívá (pracoval se třídami IP adres), ale EIGRP je dosud často používán. Na rozdíl od RIP je metrika protokolu EIGRP poměrně složitá a je považována za velmi kvalitní.

6.6.3 Směrování algoritmem stavu spoje

Algoritmus stavu spoje (Link-state Algorithm, algoritmus nejkratší cesty) vyžaduje budování *topologické databáze sítě* (jakési mapy sítě). Router s protokolem podle tohoto algoritmu si vede tři tabulky:

- tabulku sousedů, kterou si vytvoří jako první hned po zapnutí,
- tabulku reprezentující topologickou databázi sítě, tu si po zapnutí vytvoří postupně podle aktualizačních informací ostatních routerů,
- směrovací tabulku, kterou si vytvoří podle topologické databáze.

Každý router si neustále hlídá kvalitu (stav) spojů vedoucích k sousedům (to se týká první zmíněné tabulky). Pokud zjistí změnu (některý spoj/soused se stane nedostupným nebo změní některý parametr), informuje o tom své okolí. Routery, které dostanou tuto informaci, si změnu zanesou do topologické databáze (druhé tabulky) a podle ní propočítají novou cestu k cílům (aktualizují směrovací tabulku).

Pro protokoly podle algoritmu stavu spoje je typické, že mají přehled o topologii sítě, méně zahlcují síť (posílají se jen změny ke konkrétním spojům, a většinou jen při změnách topologie) a doba konvengence je výrazně kratší (aktualizace se jen přeposílají, změnu si propočítává každý router sám, nemusejí na sebe navzájem čekat).

Samotné propočítávání tras a zjišťování nejkratší (nejoptimálnější) cesty k dané (pod)síti se provádí podle topologické databáze *Dijkstrovým algoritmem* (algoritmem nejkratší cesty, shortest path first) nebo jeho modifikací. Jedná se o jeden z nejznámějších grafových algoritmů, jehož autorem je holandský informatik Edsger Dijkstra.

Nejznámějším protokolem stavu spoje je *protokol OSPF* (Open Shortest Path First). Pracuje přesně tak jak je popsáno výše, a jeho velkou výhodou je, že se jedná o otevřený protokol (výrobce,

který ho chce ve svém zařízení implementovat, nemusí platit licenční poplatky). V současné době to je nejrozšířenější směrovací protokol pro vnitřní směrování, navíc je do určité míry použitelný i pro vnější směrování (umí pracovat s čísly autonomních oblastí).

OSPF podporuje beztřídní směrování, nemá problém s IPv6 a dovoluje (na rozdíl od RIP) existenci i více než jedné trasy k témuž cíli. Používá složitější metriku, kde zohledňuje především šířku přenosového pásma na cestě, takže metrika více odpovídá skutečné propustnosti cesty. Konvergence je velmi rychlá (což souvisí s použitým algoritmem).

Poznámka:

Typickou vlastností protokolů algoritmu stavu spoje je, že mají přehled o své síti nebo její podstatné části (autonomní systém může být rozdělen na relativně samostatné *oblasti*, které jsou navzájem propojeny hraničními routery, a pak stačí, aby měl protokol přehled o oblasti, ve které se nachází). Další typickou vlastností je rychlá konvergence a menší náchylnost k zacyklení.

Dalším používaným protokolem využívajícím algoritmus stavu spoje je například IS-IS, který je oblíbený zejména u poskytovatelů síťové konektivity (ISP).

6.6.4 Protokol BGP

Zatímco předchozí jmenované protokoly jsou spíše vnitřními směrovacími protokoly (směrují převážně uvnitř svého autonomního systému a o „zbytku světa“ mají jen hrubou nebo vůbec žádnou představu), teď se zaměříme na jeden vnější směrovací protokol.

Protokol BGP (Border Gateway Protocol) je vnějším směrovacím protokolem, tedy je typicky používán pro směrování *mezi* autonomními systémy, nikoliv uvnitř nich. Používá se především ve WAN sítích, které slouží k propojování LAN, MAN a menších WAN sítí, a tedy BGP směruje mezi autonomními systémy těchto sítí. Vlastně se jedná o nejpoužívanější vnější směrovací protokol na Internetu.

BGP samozřejmě podporuje beztřídní směrování (CIDR) a IPv6, jinak by v dnešních WAN sítích nebyl použitelný.

Je implementací *algoritmu vektoru cest*, který je hybridem algoritmu vektoru vzdáleností a algoritmu stavu spoje. Pokud k některému autonomnímu systému vede více cest, vybere tu, která vede přes méně „průchozích“ autonomních systémů.

Poznámka:

Zatímco pakety předchozích jmenovaných směrovacích protokolů (RIP, IGRP, EIGRP, OSPF, IS-IS) jsou zapouzdřovány přímo do IP paketů (vůbec se netýkají transportní vrstvy) a tedy využívají datagramovou službu, BGP pakety dělají „okliku“ vrchem přes transportní vrstvu a zapouzdřují se do TCP segmentů (tj. routery v BGP síti spolu komunikují spojovaným způsobem) a následně do IP paketů. Přesto je BGP řazen na vrstvu L3.

Důvodem využití spojované komunikace je charakter WAN sítí, ve kterých se používá – tam se jinak než spojovaně nekomunikuje.

Kapitola 7

Bezdrátové sítě

Rychlý náhled: V této kapitole se vracíme na vrstvy L1 a L2, ovšem tentokrát se budeme věnovat bezdrátovým lokálním sítím. Nejdřív se obecně seznámíme s vlastnostmi bezdrátového přenosu a souvisejícími pojmy, následně se zaměříme na Wi-fi na vrstvách L1 a L2 včetně samotného přenosu a antén. Poslední sekce je o zabezpečení Wi-fi sítí.

Klíčová slova: Wi-fi, rádiové vlny, frekvenční pásmo, ad-hoc, infrastruktura, access point, repeater, extender, AP klient, WDS, WISP, BSA, ESA, SSID, BSSID, ESSID, Beacon rámec, přístupová metoda CSMA/CA, RTS/CTS, IEEE 803.11, OFDM, QAM, signál, anténa, diverzita, MIMO, MU-MIMO, AAA, WEP, WPA, WPA2, IEEE 802.1X, Radius, WPS

Cíle studia: Po prostudování této kapitoly porozumíte rozdílům mezi „kabelovými“ a bezdrátovými lokálními sítěmi, průběhu komunikace v bezdrátové síti a využívání rádiového signálu pro přenos dat, fungování Wi-fi na vrstvách L1 a L2, a principu zabezpečení bezdrátové sítě.

7.1 Bezdrátové technologie

Přenosové technologie dělíme podle přenosového prostředku do dvou kategorií:

- *kabelové* (wired, „drátové“) – přenos probíhá přes kabel (optický nebo metalický),
- *bezdrátové* (wireless) – přenos probíhá bez použití kabelu.

V této kapitole se budeme věnovat bezdrátovým technologiím.

Jaké přenosové médium tedy u bezdrátových technologií používáme? Mohli bychom říci, že vzduch, víceméně je to pravda, ale skutečným nosným médiem pro signál je to, na co signál modulujeme (ano, u bezdrátových technologií obvykle signál modulujeme). Jsou tyto možnosti:

- *rádiové vlny* o určité frekvenci – Wi-fi, WiMAX, Bluetooth, NFC, ZigBee apod.,
- *zvukové vlny* (sonická bezdrátová technologie) – například ultrazvuk,
- *světlo* (optická bezdrátová technologie) – laser, infračervené záření (IR), mávání vlajkou…

Nás bude zajímat především první možnost. Sítě založené na přenosu po rádiových vlnách mohou být různě velké – od osobních (PAN, například Bluetooth nebo NFC) přes lokální (LAN, například Wi-fi) a metropolitní (MAN, například WiMAX) až k rozlehlým (WAN, mobilní sítě).

WLAN (Wireless LAN) – touto zkratkou označujeme bezdrátové lokální sítě. Pozor, nepleťte si tuto zkratku s VLAN, jsou to dva naprosto rozdílné pojmy.

Bezdrátové sítě můžeme také rozdělit podle míry mobility:

- *mobilní* – připojená klientská zařízení i při relativně rychlém pohybu neztratí signál, sem řadíme především mobilní WAN sítě typu GPRS, EDGE, LTE apod.,
- *fixní* – připojená klientská zařízení mohou při rychlejším pohybu ztratit signál, což se stává například u Wi-fi.

Nic není jen černé nebo jen bílé. I pro Wi-fi existují standardy, které tuto technologii přibližují k těm mobilním, ale v základu se pořád jedná o fixní bezdrátovou technologii.

Co se týče frekvenčních pásem, ve kterých bezdrátové rádiové sítě vysílají, většina využívá bezlicenční pásmo, ale některé vysílají v licencovaném frekvenčním pásmu.

Bezlicenční frekvenční pásmo je frekvenční pásmo takové, že pokud zařízení vysílá v tomto pásmu, nemusí jeho provozovatel žádat o povolení, pokud dodržuje určitá pravidla stanovená v generální licenci – tzv. *všeobecné oprávnění*. V oprávnění není zakotvena právní ochrana před rušením, jen pravidlo stanovující, že „se to nesmí“.

Licencované frekvenční pásmo – uživatel takového zařízení si musí zajistit *individuální oprávnění* k vysílání v daném frekvenčním pásmu v dotyčné oblasti, tato služba může být placená. Má také právo na ochranu v případě, že v jeho přiděleném frekvenčním pásmu vysílá někdo jiný a tedy ruší signál.

Licencované frekvenční pásmo	**Bezlicenční frekvenční pásmo**
individuální oprávnění pro jednoho žadatele	všeobecné oprávnění (nepodáváme žádost)
obvykle zpoplatněno	obvykle bez poplatku za frekvenci
ochrana před zneužitím, rušením	bez právní ochrany před zneužitím, rušením
obvykle je možné poskytovat i garanci kvality služeb	obvykle se neposkytuje garance kvality služeb

Tabulka 7.1: Srovnání licencovaného a bezlicenčního frekvenčního pásma

Poznámka:

Určení bezlicenčních a licencovaných frekvenčních pásem je v různých zemích různé, u nás je stanovuje a dozoruje Český Telekomunikační úřad (ČTÚ).

7.2 Wi-fi

Technologie Wi-fi (Wireless Fidelity) je bezdrátovou technologií pracující v bezlicenčním pásmu, a to kolem 2,4 GHz a 5 GHz (podle konkrétního podstandardu).

Základním standardem pro Wi-fi je IEEE 802.11, podstandardy (dodatky) pak k tomuto označení přidávají jedno- či dvoupísmenné zkratky (abeceda je krátká, jednopísmenné už nestačily). O standard a kompatibilitu zařízení (včetně certifikace) se stará Wi-fi Alliance.

✎ Wi-fi síť může mít jednu ze dvou podob, resp. pracuje v jednom ze dvou režimů:

- *ad-hoc* – propojíme přímo dvě koncová zařízení, nepoužijeme žádný aktivní síťový prvek (DCE),
- *infrastruktura* – použijeme (minimálně jeden) aktivní síťový prvek (DCE), žádná dvě koncová zařízení nekomunikují přímo, veškerá komunikace jde přes DCE.

7.2.1 Access point

✎ Aktivním síťovým prvkem je *access point* (AP, přístupový bod). V základu toto zařízení pracuje na vrstvě L2 (a taky samozřejmě L1), tedy protokol IEEE 802.11 se svými podstandardy implementuje vrstvy L1 a L2 (jako Ethernet). Počítá se s podsunutím pod TCP/IP či jiný síťový zásobník pro vrstvy L3–L7.

✎ Ovšem DCE může podporovat i protokoly vyšších vrstev než jen L2, pak samozřejmě poskytuje další funkce související právě s vyššími vrstvami, nebo naopak může implementovat jen vrstvu L1. Takže DCE ve Wi-fi síti může pracovat v těchto *režimech*:

- *Access Point* (AP) – základní varianta, jednoduše implementuje vrstvy L1 a L2. Je to obdoba switche z ethernetových sítí.
- *Wi-fi router* – je to AP s přidanou funkcionalitou vrstvy L3. Díky tomu, že „vidí“ i do záhlaví IP paketů, dokáže směrovat a nabízí funkce typu překladu adres (NAT), DHCP serveru, hardwarového firewallu (filtruje podle IP záhlaví) apod.
- *Wi-fi repeater, extender* – pracuje jen na vrstvě L1, tedy žádné rámce. Přijme signál a zesílený ho znovu odvysílá, případně znovu vygeneruje. Tento typ zařízení má předně problémy s kompatibilitou (zejména při komunikaci mezi zařízeními různých výrobců), a navíc snižuje propustnost sítě (přijímá a vysílá obvykle v tomtéž pásmu, takže propustnost se může snížit i na polovinu). V domácnostech je to celkem užitečný prvek pro zvýšení dosahu signálu.
- *AP klient* – komunikuje bezdrátově s plnohodnotným AP, k němu jsou připojeni Wi-fi klienti obvykle kabelem. Na rozdíl od předchozího řešení zde tedy máme kabely, vzhledem k AP se tváří jako klientské zařízení a tedy jen zprostředkovává komunikaci „pravých“ klientů s AP. Nedochází k tak velkému snížení propustnosti sítě.
- *WDS* (Wireless Distribution System) – propojíme více AP do sítě, abychom pokryli větší prostor, pohybující se klient je „předáván“ mezi propojenými AP. Není to standardizované řešení, jsou problémy s kompatibilitou zařízení od různých výrobců. Navíc se podstatně snižuje propustnost sítě, podobně jako u repeateru.
- *Brána* – zprostředkovává komunikaci s jinou sítí pracující s odlišnými protokoly. Ve formě modulu bývá tato funkcionalita často přítomna v ADSL/VDSL Wi-fi routerech, přičemž modul propojuje místní síť (WLAN) s přístupovou sítí (ADSL nebo VDSL). Zařízení má tedy WAN port (tj. port vedoucí do přístupové sítě) daného typu, například ADSL (to by byl port s rozhraním RJ-11, kabel by vedl do zásuvky na pevnou linku).
- *WISP AP* (Wireless Internet Service Provider AP) – bezdrátový je WAN port (tj. se svým poskytovatelem internetu komunikujeme bezdrátově), ktežto porty pro lokální síť jsou například ethernetové.

AP (ať už jakéhokoliv typu) vysílá signál, oblast pokrytou tímto signálem nazýváme *buňka* nebo také *základní oblast služeb* (BSA – Basic Service Area). Pokud propojíme víc AP do distribučního systému (WDS), pak oblast pokrytou signálem zapojených AP nazýváme *rozšířená oblast služeb* (ESA – Extended Service Area).

Klientská zařízení potřebují identifikovat síť, do které se připojují nebo v ní pracují. Od toho máme tyto *identifikátory*:

- *SSID* (Service Set ID) je identifikátor celé sítě, název sítě. Je to řetězec o délce maximálně 32 znaků, který musí znát každé zařízení, které se do sítě přihlašuje.
- *BSSID* (Basic Service Set ID) je identifikátor access pointu. Obvykle se jedná o MAC adresu tohoto AP, tedy délka je 6 oktetů. Každé zařízení v buňce musí znát BSSID svého AP, protože tuto adresu umísťuje do záhlaví odesílaných rámců.
- *ESSID* (Extended Service Set ID) je identifikátor WDS, pokud máme víc AP propojených do distribučního systému.

Definice (Beacon rámec)
Každý AP v pravidelných intervalech vysílá *Beacon rámec* („majákový“ rámec), ve kterém informuje o svých parametrech. Kromě jiného je v Beacon rámci řetězec SSID.

AP poskytuje v bezdrátové síti tyto služby:

- *autentizace* – rozhoduje, zda do své buňky a tím i do sítě pustí konkrétní koncové zařízení,
- *asociace* (přidružení) – pokud je koncovému zařízení povolen přístup, je třeba vytvořit vazbu mezi AP a tímto zařízením, tedy technicky zajistit zařízení možnost komunikace v buňce,
- *de-asociace* – zrušení této vazby.

7.2.2 Wi-fi na vrstvě L2

IEEE 802.11 implementuje jen spodní podvrstvu L2 – MAC podvrstvu. Horní podvrstvu nechává na protokolu LLC (IEEE 802.2), což znamená, že používáme LLC rámce (jak bylo naznačeno v druhé kapitole od strany 42), které zapouzdřujeme do MAC rámce podle IEEE 802.11.

Rozlišujeme tři typy MAC rámců podle IEEE 802.11:

- *datové rámce* (Data Frame) – obsahují payload v LLC rámci,
- *řídící rámce* (Control Frame) – používají se například pro potvrzení přijatých rámců,
- *rámce pro správu* (Management Frame) – sem řadíme Beacon rámce, rámce pro dojednání připojení klientského zařízení do sítě (asociace) či odpojení zařízení, apod.

Zaměřme se na záhlaví MAC rámce podle IEEE 802.11. Je dlouhé 30 oktetů, což je poněkud více než u Ethernetu (navíc se také přidává zápatí s kontrolním součtem, stejně jako u Ethernetu). Nebudeme je probírat do podrobností, ukážeme si jen práci se dvěma příznaky a s adresami. Všechny tyto údaje se v záhlaví mění podle toho, kterou částí Wi-fi sítě právě rámec prochází. Rozlišujeme tyto případy:

1. Jsme v ad-hoc síti, tedy spoje vedou vždy právě mezi dvěma klientskými zařízeními, žádný AP na cestě není.

2. Jsme v síti typu infrastruktura, rámec je na spoji mezi zdrojem rámce (klientským zařízením) a některým AP.
3. Jsme v síti typu infrastruktura, rámec je na spoji mezi dvěma AP.
4. Jsme v síti typu infrastruktura, rámec je na spoji mezi některým AP a cílem rámce (klientským zařízením).

Takže záleží, na jakém typu spoje se právě rámec nachází, resp. mezi jakými dvěma typy uzlů v síti právě prochází (kdo je momentálním odesílatelem a příjemcem).

V záhlaví máme tyto dva příznaky (tj. jednobitové hodnoty):

- `FromDS` – je nastaven na 1, pokud na daném spoji je momentálním odesílatelem AP (resp. DCE), a je nastaven na 0, pokud je odesílatelem klientské zařízení (DTE).
- `ToDS` – je nastaven na 1, pokud na daném spoji je momentálním příjemcem AP (DCE), a je nastaven na 0, pokud je příjemcem DTE.

Zkratka „DS“ v názvech příznaků je *distribuční systém*.

Z toho vyplývá, že v nastíněných čtyřech případech jsou tyto příznaky nastaveny takto:

1. V ad-hoc síti jsou oba příznaky vždy nastaveny na 0, protože žádné AP na cestě ani být nemohou.
2. Na spoji mezi zdrojem rámce (DTE) a nejbližším AP je `FromDS=0` (odesílatel je DTE), `ToDS=1` (momentální příjemce je AP).
3. Na spoji mezi dvěma AP jsou oba příznaky nastaveny na 1: `FromDS=1`, `ToDS=1`. Na obou koncích spoje jsou AP, a tedy jsme uvnitř distribučního systému.
4. Na spoji mezi některým AP a cílem rámce (DTE), tedy v závěru cesty, je `FromDS=1`, `ToDS=0`.

Nyní krátce k *adresám*. V ethernetovém rámci máme vždy dvě adresy – MAC adresu cíle a MAC adresu zdroje. Ve Wi-fi rámci sice také používáme MAC adresy (stejný typ a struktura adresy jako u Ethernetu), ale jejich počet se dynamicky mění podle momentální pozice v síti, podobně jako zmíněné dva příznaky.

V ad-hoc síti máme jen MAC adresu cíle a MAC adresu zdroje, nic jiného nepotřebujeme. Ale v režimu infrastruktury se přidávají adresy nejbližších AP na daném úseku cesty – nejvíc adres máme v záhlaví v případě, že rámec právě „putuje“ mezi dvěma AP (v záhlaví jsou adresy zdroje, cíle a obou okolních AP). Počet a pořadí adres se v záhlaví postupně mění, ale vždy platí, že první dvě adresy v pořadí se týkají právě aktuálního úseku cesty a a na tomto úseku je vždy první v pořadí adresa konce úseku, pak teprve adresa začátku úseku.

7.2.3 Přístupová metoda

Standard IEEE 802.11 definuje také přístupovou (kolizní) metodu, která je používána na podvrstvě MAC vrstvy L2. Na rozdíl od Ethernetu je v bezdrátových sítích používána neustále, protože při sdíleném médiu se nikdy nedá zcela vyhnout kolizím.

Definice (Přístupová metoda CSMA/CA, mechanismus RTS/CTS)

Podle IEEE 802.11 se používá přístupová metoda *CSMA/CA*:

- CS (Carrier Sense) – naslouchámé na nosné,
- MA (Multiple Access) – vícenásobný přístup (sdílené přenosové médium),

- CA (Collision Avoidance) – kolizím se vyhýbáme, tedy jim předcházíme (nestačí je jen detekovat).

Každé zařízení, které chce vysílat, musí nejdřív požádat AP o povolení k vysílání (to je krátký správní rámec) a až ho získá, může odesílat data. CSMA/CA se dá implementovat více různými způsoby, dnes je běžný mechanismus *RTS/CTS* (Request to Send, Clear to Send):

- DTE vyšle signál RTS (v krátkém řídícím rámci), čímž požádá o povolení k vysílání dat,
- pokud je možné vysílání povolit, AP odešle žádajícímu DTE signál CTS (v řídícím rámci),
- DTE může vyslat datový rámec (MAC podle IEEE 802.11 s LLC rámcem, v něm jsou data),
- zpět je doručeno potvrzení (ACK), také v řídícím rámci.

Jak vidíme, ve Wi-fi se v takovém případě používá potvrzovaný přenos bez navázání spojení.

Ovšem i přes takto určenou přístupovou metodu ke kolizím může docházet, protože přenosové médium je sdílené. Dochází k nim například tehdy, pokud dvě zařízení zároveň vyšlou signál RTS nebo se takto potkají rámce s RTS a začátkem vysílání datového rámce.

Problém skrytých stanic nastává tehdy, když je buňka rozsáhlejší a dvě stanice patřící do téže buňky se navzájem „nevidí“ (ke každé z nich sice dosáhne signál z centrálního AP, ale nikoliv jejich signály navzájem). To je vcelku běžné, stanice mohou být od sebe příliš vzdáleny nebo navzájem zastíněny nábytkem či zdí, signál koncových zařízení taky bývá typicky slabší než signál AP.

Poznámka:
Mechanismus RTS/CTS víceméně řeší problém skrytých stanic – počet kolizí snižuje na minimum tím, že u krátkých řídících rámců je mnohem menší pravděpodobnost kolize než u velkých datových, a AP taky nepošle signál CTS s povolením vysílání, pokud zrovna na dané frekvenci komunikuje jiný (žadateli skrytý) uzel v síti.

7.2.4 Fyzická vrstva

Na fyzické vrstvě se rozlišuje více různých podstandardů s hodně odlišnými vlastnostmi, přičemž čas od času přibude nový podstandard.

Specifikace	Frekvence	Max. propustnost	Ozn.
IEEE 802.11	2,4 GHz	2 Mb/s	
IEEE 802.11a	5 GHz	54 Mb/s	Wi-Fi 1
IEEE 802.11b	2,4 GHz	11 Mb/s	Wi-Fi 2
IEEE 802.11g	2,4 GHz	54 Mb/s	Wi-Fi 3
IEEE 802.11n	2,4 GHz, 5 GHz	250 Mb/s na anténu, celkem až 600 Mb/s	Wi-Fi 4
IEEE 802.11ac	5 GHz	433 Mb/s na anténu a stream	Wi-Fi 5
IEEE 802.11ad	60 MHz	cca 7 Gb/s, podle použitých kanálů	WiGig
IEEE 802.11ax	2,4 GHz, 5 GHz	jednotky Gb/s	Wi-Fi 6

Tabulka 7.2: Podstandardy pro fyzickou vrstvu Wi-fi

V tabulce 7.2 je přehled existujících standardů pro fyzickou vrstvu (ve skutečnosti je jich mnohem více, ale zbývající jsou spíše doplňkové podstandardy).

Údaj v sloupci o propustnosti je třeba brát s rezervou, někdy se tento údaj dá považovat dokonce za marketingový. V reálu se rychlost snižuje například tím, že je nutné komunikovat v obou směrech, což je při využití téže frekvence problém, může být asociováno větší množství stanic, svůj vliv má také vzdálenost mezi komunikujícími uzly a případné překážky. Záleží taky na počtu antén, používané šířce kanálů, počtu streamů, atd.

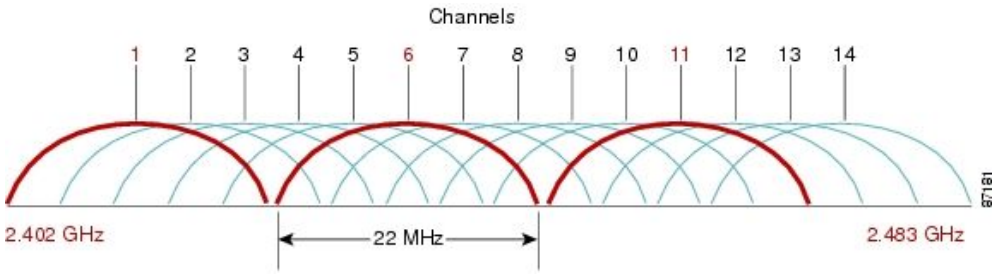
Zařízení podporující standard pro Wi-fi mívají v označení určeno, které frekvence a další vlastnosti podporují. Například IEEE 802.11b/g/n znamená, že zařízení má implementovány podstandardy b, g, n z výše uvedených, kdežto IEEE 802.11a/n/ac má implementovány podstandardy a, n, ac.

✎ **IEEE 802.11.** Původní znění standardu z roku 1997 bylo již po dvou letech aktualizováno, na trhu se objevilo jen málo těchto produktů.

✎ **IEEE 802.11b.** Tento předpis stanovuje komunikaci na frekvencích kolem 2,4 GHz, přičemž propustnost může být až 11 Mb/s (z toho 30–40 % je režie). Dnes se v zařízeních objevuje jeho implementace jen z důvodů kompatibility s IEEE 802.11g/n.

Na fyzické vrstvě se používá multiplexovací metoda *DSSS* (Direct Sequence Spread Spectrum) určující úpravu posloupnosti bitů s modulací *QPSK* (Quadrature Phase Shift Keying), třebaže se objevila zařízení podporující modulaci 64-QAM. Metoda je robustní, data se přenášejí redundantně, aby při jejich poškození bylo co nejjednodušší zjistit chybu.

Na obrázku 7.1 vidíme frekvenční rozsah využívaný podle IEEE 802.11b. Rozhodně nejde pouze o frekvenci 2,4 GHz, ale o interval frekvencí v blízkém okolí.



Obrázek 7.1: Frekvenční spektrum pro IEEE 802.11b[1]

✎ Celé spektrum se dělí na *kanály*, šířka kanálu je 22 MHz. Zařízení (DTE i AP) má určen konkrétní kanál, na kterém komunikuje. AP si svůj kanál obvykle volí sám podle analýzy zarušení spektra, ale v některých případech je třeba tuto hodnotu určit ručně (AP totiž nemůže „tušit“, které kanály jsou rušeny na straně klientů).

Ovšem pozor, v různých zemích mohou být stanoveny odlišné „povolené“ kanály. Zařízení, které je homologováno pro provoz v ČR, by mělo komunikovat pouze na kanálech povolených ČTÚ, což jsou ty s čísly 1–13.

Pokud zařízení komunikuje na konkrétním kanálu, ve skutečnosti je signál i na okolních kanálech. Teoreticky je zabráno celkem pět kanálů (například pokud zařízení komunikuje na kanálu 6,

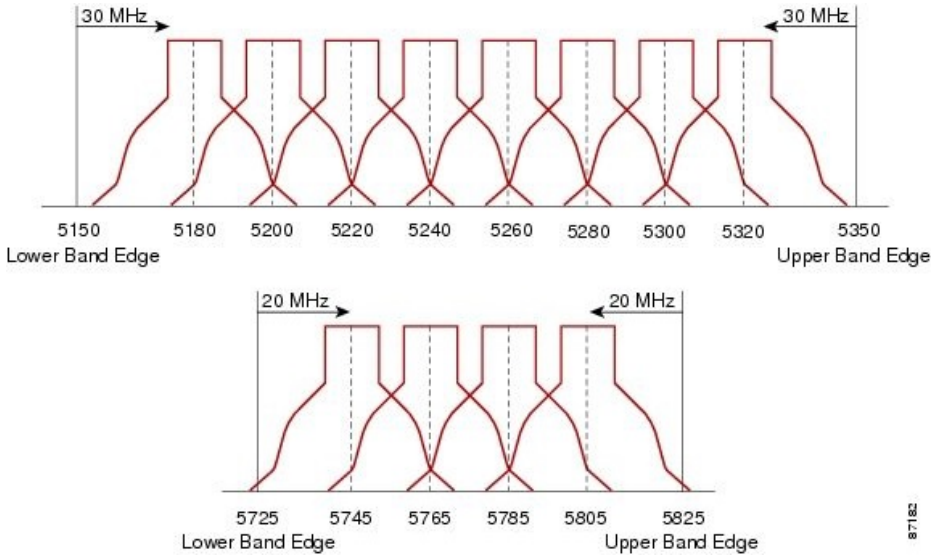
[1] Zdroj: http://www.cisco.com/en/US/docs/solutions/Enterprise/Mobility/emob30dg/RFDesign.html

pak zabírá kanály 4–8, podle „obloučku“ na obrázku), ale praxe bývá trochu drsnější – na vzdálenějších frekvencích síla signálu rozhodně neklesá tak prudce jak oblouček předepisuje, slabé rušení je ještě i o několik kanálů dál. Kanály se překrývají v praxi poněkud víc než jak říká teorie.

Poznámka:
Takže kolik zařízení může ve vzájemném dosahu komunikovat v pásmu 2,4 GHz bez toho, aby se navzájem rušila? Teoreticky tři (kanály 1, 6 a 11). Prakticky se v tomto poměrně úzkém spektru mohou navzájem rušit dokonce i pouhá dvě zařízení, záleží na síle jejich signálu a dalších vlastnostech antény.

Ve stejném frekvenčním spektru pracují také mikrovlnné trouby (pokud si myslíte, že při provozu je mikrovlnka tak důkladně izolovaná, že ven se žádný signál nedostane, pak věřte, že jste na omylu) a většinou i dětské chůvičky, takže vzájemné rušení se ani zdaleka netýká pouze Wi-fi zařízení.

IEEE 802.11a. Přesuňme se nyní do pásma 5 GHz. Zařízení s anténami pracujícími v tomto frekvenčním pásmu mají výhodu v tom, že toto pásmo není tak zarušeno jako 2,4 GHz. Zařízení podle standardu IEEE 802.11a mohou komunikovat teoreticky rychlostí až 54 Mb/s (opět jde o relativní údaj, to bude pro všechny specifikace stejné).



Obrázek 7.2: Frekvenční spektrum pro IEEE 802.11a[2]

Nevýhodou je, že přidání podpory 5 GHz znamená podstatné navýšení ceny zařízení. Specifikace je z roku 1999, ale první zařízení s její podporou se objevila až na konci roku 2001.

Na obrázku 7.2 je naznačeno rozdělení frekvenčního spektra pro IEEE 802.11a do kanálů. Kanály o šířce 20 MHz se překrývají trochu jiným způsobem (méně) než u IEEE 802.11b a spektrum je rozděleno do dvou oddělených částí.

Používá se multiplexovací metoda OFDM, která je efektivnější než DSSS. Vyšších rychlostí oproti IEEE 802.11b je dosaženo především díky efektivnější modulaci a přechodu na vyšší frekvence (při dvojnásobné frekvenci dokážeme za stejnou časovou jednotku přenést dvakrát víc dat, pokud jsou všechny ostatní parametry shodné).

[2]Zdroj: http://www.cisco.com/en/US/docs/solutions/Enterprise/Mobility/emob30dg/RFDesign.html

Jak bylo uvedeno, používá se metoda OFDM, přičemž na subnosných se moduluje některou z více různých metod (podle toho, jestli se upřednostňuje robustnost nebo rychlost) – BPSK, QPSK nebo QAM, pro QAM se dají použít různé varianty (robustnější 16-QAM, rychlejší 64-QAM).

IEEE 802.11g. Tato specifikace je z června 2003 a je zpětně kompatibilní s IEEE 802.11b v tom smyslu, že se používá stejné frekvenční spektrum (viz obrázek 7.1). Šířka kanálů je 20 MHz (pozor, změna), kanály jsou více „nahuštěny“ než u IEEE 802.11b. Pro multiplexování se používá metoda OFDM, subnosné se také modulují stejně jako u IEEE 802.11a.

Jak vidíme, mezi IEEE 802.11a a IEEE 802.11g je rozdíl především v použitém frekvenčním pásmu, jinak rychlost a modulace jsou podobné. Je tady však ještě jeden podstatný rozdíl – podpora u výrobců a cena. Géčko se stalo cenově mnohem lépe dosažitelným a v minulosti jej implementovala téměř všechna bezdrátová síťová zařízení.

Co se stane, když se do buňky pracující podle IEEE 802.11g asociuje zařízení zvládající pouze IEEE 802.11b? To záleží na konfiguraci AP v centru buňky.

- Výchozí bývá *Legacy Mode* fungující takto:
  - ta zařízení, která mají implementovány oba podstandardy, přejdou na IEEE 802.11b, čímž se sníží propustnost v celé buňce,
  - ta zařízení, která mají implementován IEEE 802.11g, ale ne „béčkový“ podstandard, budou odpojena.
- Pokud máme nastaven *Mixed Mode*, všechna zařízení pracují na nejvyšším možném podstandardu, který je na nich implementován.
- Třetí možnost je, že starým zařízením bez podpory „géčka“ nebude asociace povolena, což je asi nejlepší možnost vzhledem k zachování maximální propustnosti sítě.

Toto chování se nastavuje v konfiguraci AP, a není řečeno, že dotyčný AP podporuje opravdu všechny tyto možnosti.

IEEE 802.11n (Wi-Fi 4). Tato specifikace jako první umožnila pracovat v obou frekvenčních pásmech. Maximální propustnost se zvýšila až na 600 Mb/s (součet přes všechny antény), čehož bylo dosaženo takto:

- typicky se používá více než jedna anténa,
- používáme obě frekvenční pásma, a to i paralelně,
- obvyklá šířka kanálu je 40 MHz,
- kvalitnější modulace (OFDM, ale s jinými parametry),
- změny jsou i na vrstvě L2.

Čím víc antén, tím větší propustnosti teoreticky dosahujeme (prakticky však nemůžeme brát jen součet přes všechny antény).

Poznámka:

Implementace na straně výrobců hlavně ze začátku poněkud pokulhávala – například existují zařízení, která sice implementují IEEE 802.11n, ale pouze v pásmu 2,4 GHz, případně zařízení s pouze jednou anténou, nebo taková zařízení, která implementují IEEE 802.11n v obou pásmech, ale nedokážou je používat paralelně v jednom okamžiku (tj. pracují buď na 2,4 GHz nebo na 5 GHz, ale ne v obou pásmech zároveň).

Zatímco specifikace IEEE 802.11a/b/g se liší opravdu jen na fyzické vrstvě (podvrstvu MAC vrstvy L2 mají stejnou), specifikace IEEE 802.11n mění i podvrstvu MAC.

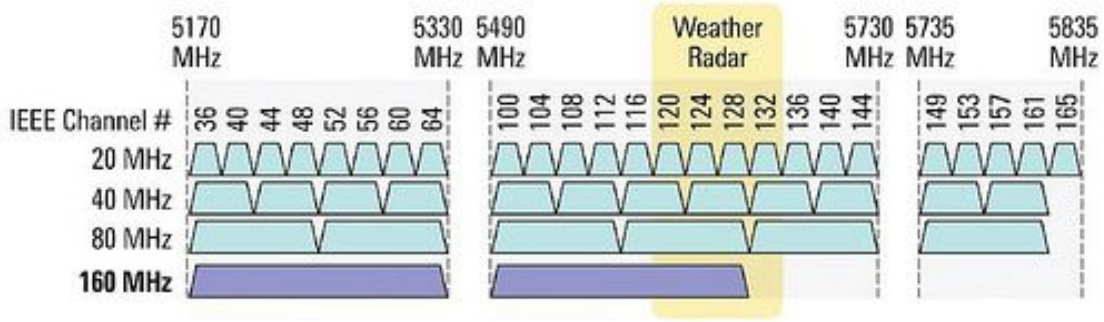
Při souběhu zařízení podporujících různé podstandardy (v tomtéž frekvenčním pásmu) může AP běžet v jednom z těchto tří módů:

- *Legacy Mode* – používá se jen starší standard (z průniku podporovaných podstandardů přes všechna zařízení v buňce). Obvykle rychlost spadne na tu nejnižší společnou.
- *Mixed Mode* – každé zařízení v buňce funguje podle svého nejlepšího podstandardu, pokud ovšem to dovoluje množství a nastavení antén AP.
- *Greenfield* – opak Legacy módu, používá se jen IEEE 802.11n a všechna zařízení, která ho nepodporují, mají smůlu.

Záleží samozřejmě, co vše dotyčný AP dovoluje nastavit. Z hlediska propustnosti sítě je nejlepší třetí možnost.

IEEE 802.11ac (Wi-Fi 5). Tento standard již plně přešel do pásma 5 GHz. Používá multiplexování OFDM s vnitřní modulací QAM, ale opět s jinak nastavenými parametry, aby se co nejvíc zvýšilo množství dat přenesených za časovou jednotku. Vliv na propustnost samozřejmě má i výhradní použití vyšších frekvencí.

Frekvenční pásmo je ve skutečnosti o něco širší než u IEEE 802.11a. Šířka kanálů může být 20 MHz, 40 MHz, 80 MHz nebo 160 MHz, konkrétní hodnotu volíme podle očekávaného množství klientů.



Obrázek 7.3: Frekvenční spektrum pro IEEE 802.11ac[3]

Podstandard IEEE 802.11ac navyšuje rychlost oproti starším následovně:

- používá lepší modulaci,
- dovoluje více antén (až 8) a zachází s nimi mnohem optimálněji (víc v další sekci),
- pracuje ve frekvenčním pásmu 5 GHz, přičemž šířka kanálu může být určena adaptivně podle vytížení sítě.

Používá se multiplexování OFDM, subnosné jsou pak zpracovány modulací QAM, a to až 256-QAM. V jednotlivých generacích se tento parametr měnil následovně:

- 802.11g: 16-QAM (jeden symbol = 4 bity, $2^4 = 16$ možností pro hodnotu symbolu),
- 802.11n: 64-QAM (jeden symbol = 8 bitů),
- 802.11ac: 256-QAM (16 bitů).

[3] Zdroj: http://www.dailywireless.org/2012/04/page/3/

Důsledkem bylo postupné zvyšování efektivity přenosu (víc přenesených dat za jednotku času).

Uvedené platí pro víceméně ideální podmínky. Při rušení nebo větší vzdálenosti, kdy se zhoršuje kvalita doručeného signálu a dochází k častějšímu ztrácení rámců, se přechází na robustnější modulaci (symbol je kódován do více bitů).

IEEE 802.11ad (WiGig) je tak trochu úkrok stranou, není to ani tak další generace, jako spíše specialita pro specifické použití. Hlavním účelem bylo vytvořit bezdrátovou náhradu HDMI kabelu, tedy vysokorychlostní přenos multimediálních dat na krátkou vzdálenost v řádech metrů. Typický dosah je do 10 metrů, a to bez překážek. Podle novější specifikace je k dispozici až 6 kanálů (každá země to má jinak, například Čína pouze 2 kanály, USA všech 6, v Evropě používáme 4 kanály) o šířce 2,16 GHz.

Použití frekvenčního rozsahu kolem 60 GHz totiž má jak kladné, tak i záporné následky:

+ teoretická propustnost až v jednotkách GHz, což postačuje pro přenos nekomprimovaného UHD videa,
– vyšší frekvence jsou náchylnější na rušení a navíc špatně procházejí překážkami, dokonce i vzduch má na signál tlumící účinek.

Typické použití je bezdrátové propojení výpočetního zařízení (počítač, notebook, HTPC apod.) se zobrazovacím zařízením (televize, projektor apod.)., ovšem často narážíme na to, že zařízení tento standard nepodporují.

IEEE 802.11ax (Wi-Fi 6) používá frekvenční spektrum kolem 2,4 GHz a 5 GHz, stejně jako IEEE 802.1n. Kanály jsou široké až 160 MHz (taky záleží, ve které části frekvenčního spektra), modulace je až 1024-QAM s multiplexem OFDMA (stejně jako u LTE), počet paralelních streamů se navýšil na 8 (to vše bude diskutováno dále). Pro zabezpečení se počítá s WPA3.

Maximální teoretická propustnost je oproti Wi-Fi 5 několikanásobně vyšší, ale samozřejmě záleží na počtu antén, šířce a využití kanálů, atd. Zatímco maximum pro Wi-Fi 5 je 3,5 Gb/s (160MHz kanály, 4 streamy), u Wi-Fi 6 to je 9,6 Gb/s (souhrn za všechny antény, 160MHz kanály, 8 streamů).

W-Fi 6 má ještě jednu vlastnost navíc – počítá se se zapojením IoT zařízení (Internet věcí), čemuž je přizpůsobeno mnohé, včetně nové funkce *TWT* (Target Wake Time) – inteligentní plánování uspávání a probouzení IoT zařízení.

Další informace:

- https://www.wi-fi.org/discover-wi-fi (v menu vpravo jsou různé standardy a další témata)
- https://www.cisco.com/c/en/us/products/collateral/wireless/white-paper-c11-740788.html (srovnání IEEE 802.11ax a IEEE 802.11ac)

7.2.5 Signál a antény

Existují různé druhy antén – všesměrové s různými způsoby formování signálu či směrové určené pro point-to-point spoje. Antény mohou být buď *interní* (například v notebooku jsou antény obvykle vedeny podél displeje) nebo *externí* (vyměnitelné jsou obvykle připojeny koaxiálovým konektorem typu N nebo BNC).

U běžných malých všesměrových antén je signál nejsilnější v rovině kolmé na osu antény, tedy nakloněním antény můžeme například ovlivnit to, zda bude signál dostatečně silný v okolních podlažích budovy. Pokud chceme mít signál jen v rámci jednoho patra, necháme anténu směřovat nahoru nebo dolů, kdežto když má signál dosáhnout do horního nebo spodního patra, nakloníme anténu do úhlu 45 stupňů. Pozor, ve směru, do kterého anténa míří, je signál nejslabší!

Zařízení s jedinou anténou může fungovat jen v *poloviční duplexu*, tedy buď vysílat nebo přijímat, ale nikoliv obojí zároveň. Pokud ale má alespoň dvě antény, může pro každý směr použít jednu z nich, přičemž budou pracovat na různých kanálech, čímž implementujeme *plný duplex*.

Ve specifikaci IEEE 802.11a/b/g se setkáváme s *diverzitou*, tedy možností využití více antén pro frekvenční pásmo 2,4 GHz. Diverzita funguje tak, že při komunikaci s konkrétním zařízením se nejdřív přijímá všemi, a po vyhodnocení kvality přijatého signálu ze všech antén je pak určena jedna, která bude pro dané zařízení využívána.

Ve specifikaci pro IEEE 802.11n se objevuje nástupce diverzity – *MIMO* (Multiple Input, Multiple Output) [maimo] – určující využití více antén a algoritmus pro kombinování signálu z těchto antén. Na rozdíl od diverzity mohou (téměř) všechny antény fungovat zároveň.

Při použití MIMO jsou antény rozděleny na Tx (odesílající) a Rx (přijímající), abychom zajistili komunikaci v plném duplexu, a dalším parametrem je počet paralelních streamů (tedy kolik nezávislých kanálů se nám vejde do spektra). Zapisuje se takto:

Tx		Rx		streamy
3	×	3	:	2

což znamená 3 odesílající antény, 3 přijímající antény a 2 streamy.

Technologie MIMO má jednu slabinu – v jednom okamžiku může AP komunikovat maximálně s jediným klientem, třebaže více anténami.

To řeší technologie *MU-MIMO* (Multi-User MIMO) používaná ve specifikaci IEEE 802.11ac, a také například u metropolitních sítí WiMAX. V jednom okamžiku může AP paralelně komunikovat s více různými klienty (maximum je 8 streamů, pro každého klienta 2, tedy maximálně 4 klienti).

V implementaci je bohužel trochu zpoždění, tedy když kupujeme zařízení podle IEEE 802.11ac podporující MU-MIMO, může jít o jednu ze tří fází, obvykle druhou:

- Wave 1 – SU-MIMO (Single-User MIMO), tedy žádná paralelní komunikace, max. 3 streamy,
- Wave 2 – MU-MIMO (více klientů paralelně), max. 3–4 streamy,
- Wave 3 – max. 8 streamů (s tím se obvykle u Wi-Fi 5 nesetkáme).

V IEEE 802.11ax (Wi-Fi 6) se přechází z multiplexování OFDM na OFDMA. Co to znamená? Ve skutečnosti se liší i význam „konců“ těchto zkratek: OFDM je Orthogonal Frequency Division Multiplexing, OFDMA je Orthogonal Frequency-Division Multiple Access. Při OFDM je sice možné používat MU-MIMO, ale pouze ve formě, kdy různé kanály jsou přiřazeny různým koncovým zařízením. OFDMA jde dál: kanály dělí na subkanály (nazývají se RU = Resource Unit). Také v rámci subkanálů se používá ortogonální dělení, aby se jednotlivé subkanály daly jednoduše oddělit.

Zatímco IEEE 802.11ac (a také starší IEEE 802.11n) reálně používá max. čtyři streamy (Wave 2), u IEEE 802.11ax se setkáme s až osmi streamy. Počet streamů udává maximální počet

(sub)kanálů, které lze používat paralelně (počítají se zvlášť pro různé směry). Například pokud jsou u Wi-fi routeru pracujícího podle IEEE 802.11ac k dispozici 4 streamy a v síti máme dva aktivní klienty, pak ke každému z nich mohou vést dva streamy (jeden pro každý směr). Ovšem pozor, klienti by v tom případě taky museli podporovat alespoň dva streamy.

Další informace:

- http://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/802-11ac-solution/q-and-a-c67-734152.html
- https://www.extremenetworks.com/extreme-networks-blog/ofdm-and-ofdma-subcarriers-what-are-the-differences/

7.3 Zabezpečení bezdrátové komunikace

7.3.1 AAA

AAA (Authentication, Authorization, Accounting) je termín označující řízení přístupu, v našem případě přístupu do bezdrátové sítě. Jak zkratka napovídá, zahrnuje tři fáze:

- *Autentizace* (Authentication) – klient (a v některých případech i AP) má prokázat svou totožnost, jde tedy o kontrolu identity. Přístupové údaje se také označují anglickým termínem *credentials*.
- *Autorizace* (Authorization) – když je identita klienta zjištěna, je třeba mu přidělit přístupová oprávnění, vymezit povolenou činnost.
- *Účtování* (Accounting) – evidujeme stanovené činnosti klienta v síti (podle konfigurace), případně reagujeme při pokusech o překročení oprávnění přidělených ve fázi autorizace.

Každá z následujících možností zajišťuje šifrování jak v první fázi komunikace (autentizace a asociace), tak i při běžném provozu. Ovšem používají se různé postupy a různé (i vzhledem k bezpečnosti) šifrovací algoritmy.

Krátká poznámka k šifrování. U jakéhokoliv šifrování potřebujeme *šifrovací klíč*, který mají mít obě strany komunikace (odesílatel pro šifrování a příjemce pro dešifrování).

Existují dva základní druhy šifer – *symetrické a asymetrické*. Vzhledem k šifrovacím klíčům:

- Symetrické šifry používají tentýž klíč pro šifrování i dešifrování. Hlavním problémem (a taky nevýhodou) je, jak *bezpečně* dostat klíč druhému komunikujícímu.
- Asymetrické šifry používají pro každý směr komunikace pár klíčů – soukromý a veřejný, přičemž co zašifruji veřejným klíčem, musí být dešifrováno soukromým (a naopak – ve funkčnosti jsou zaměnitelné). Uživatel vygeneruje svůj pár klíčů, soukromý si uloží a veřejný předá (jakkoliv) druhé straně. Druhá strana udělá totéž se svým párem klíčů. Když chci něco odeslat, zašifruji data veřejným klíčem adresáta, pošlu a adresát si to dešifruje svým soukromým klíčem.

Nevýhodou symetrických šifer je tedy problém předání klíče protistraně, ale výhodou velká rychlost šifrování a dešifrování. U asymetrických šifer je to přesně naopak – výhodou je možnost transportu veřejného klíče i nezabezpečeným kanálem, nevýhodou velká výpočetní náročnost (hodně zdržuje přenos).

Proto se v praxi oba přístupy některým způsobem kombinují – používáme *hybridní šifrování*. Po navázání spojení se přenášená data šifrují symetricky, aby nebyla komunikace příliš zdržována, ale předem se symetrický klíč předá pomocí asymetrického šifrování (místo dat prostě zašifrujeme symetrický klíč, čímž ho dokážeme relativně bezpečně předat protistraně).

Další informace:

Pro ty, kteří cítí potřebu navýšit své znalosti o šifrování – různé šifrovací algoritmy jsou popsány například ve skriptech jiného předmětu: http://vavreckova.zam.slu.cz/obsahy/analyzadat/analyzadat.pdf.

Princip šifrovacích algoritmů je vcelku podrobně popsán v knize [?].

Pro zvýšení zabezpečení se kromě (symetrického) statického šifrovacího klíče pro každý odeslaný rámec používá také dynamicky měnící se dodatek – *IV vektor* (inicializační vektor). Účelem je maximálně ztížit odposlech komunikace. Jenže když se něco neustále mění, musíme mít mechanismus, jak dát adresátovi vědět, jakou podobu má IV vektor pro ten konkrétní rámec (je pěkné, že hacker náš rámec nedešifruje, ale adresát by toho měl být schopen). Pro to jsou dva přístupy: buď mají obě strany algoritmus, kterým pro daný rámec IV vektor určí, nebo momentální IV vektor prostě adresátovi pošleme. První možnost je samozřejmě mnohem bezpečnější.

WEP (Wired Equivalent Privacy). Jedná se o mechanismus zabezpečení, který už rozhodně bezpečným nazývat nelze. Jedinou výhodou je plná kompatibilita se vším, co má implementováno IEEE 802.11.

Používá se stejné šifrování pro první fázi komunikace i pro běžný provoz (pokud vůbec nějaké). Šifruje se pouze symetrickým algoritmem, tedy klíč musí být nějak předán adresátovi (klientovi). IV vektor se v naprosté většině případů přenáší jako součást (nešifrovaného) záhlaví.

WEP nabízí tři režimy:

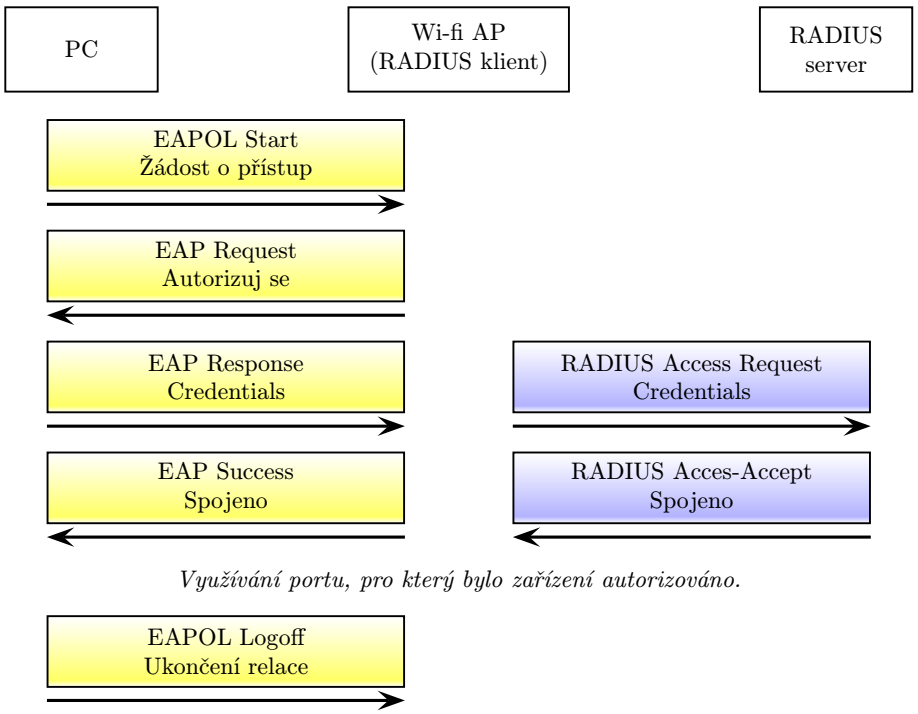
- otevřená síť (bez šifrování),
- symetrická šifra RC4 s 64bitovým klíčem, přičemž IV vektor tvoří jeho 24 bitů (takže 40 bitů je statická část klíče),
- symetrická šifra RC4 se 128bitovým klíčem, přičemž IV vektor taky tvoří jeho 24 bitů (104 bitů je statická část klíče).

V každém případě pro cracknutí klíče stačí několik minut (s použitím programu AirCrack), takže WEP je považován za naprosto nevyhovující způsob zabezpečení.

WEP a IEEE 802.1X. Mechanismus WEP byl brzy shledán nedostatečným, ale jiná možnost dlouho neexistovala. Proto se ve firemních sítích začala používat dodatečná možnost zabezpečení přihlašování ve formě implementace protokolu IEEE 802.1X (pozor, ne 11). Spočívá v zajištění autentizace a autorizace pomocí autentizačního serveru, který si vede databázi „povolených“ uživatelů s přístupovými údaji pro autentizaci (credentials) a údaji pro autorizaci.

Existují dva používané typy autentizačních serverů – RADIUS a TACACS. Pro RADIUS existují volně dostupné implementace (nejoblíbenější je open-source projekt FreeRADIUS), kdežto TACACS je proprietární řešení (Cisco).

Princip je takový, že uživateli nestačí jen vlastnit klíč, musí mít i své přihlašovací údaje (například jméno a heslo nebo třeba hardwarový token). Na obrázku 7.4 je naznačena komunikace s RADIUS serverem – klient se sice přihlašuje do některé buňky k příslušnému AP, ale místo toho,



Obrázek 7.4: Zjednodušená komunikace v síti při použití RADIUS serveru

aby AP o připojení rozhodoval přímo, pouze přihlašovací údaje přepošle RADIUS serveru a při autentizaci a autorizaci vlastně funguje jen jako zprostředkovatel. Kromě přihlašovacích údajů (credentials) si RADIUS server může navíc vyžádat další údaje, podle kterých pak rozhodne o autentizaci a konkrétních parametrech autorizace (přístupových oprávněních).

RADIUS síť obsahuje jen dva uzly – RADIUS server a RADIUS klienta, kterým je dotyčný AP, spojení je point-to-point. Toto spojení by mělo být dobře zabezpečeno (žádná Wi-fi, zde by měl být kabel, a taky šifrování). Samotný klient (žadatel, suplikant) není součást RADIUS sítě.

WPA (Wi-fi Protected Access). Mechanismus WPA byl původně dočasným řešením, vznikl při čekání na standard WPA2. Dnes je považován za *něco mezi* nezabezpečeným WEP a bezpečným WPA2, přičemž na rozdíl od WPA2 je vhodný i pro starší zařízení, na kterých nelze WPA2 použít. Na takovém zařízení stačilo upgradovat firmware (tedy softwarová záležitost), WPA nevyžadoval úpravu hardwaru. Zjednodušeně můžeme WPA brát jako WPA2 osekaný o hardwarově závislé součásti.

Základ je podobný jako u WEP – šifra RC4 se 128bitovým klíčem, jehož IV vektor je 48bitový. Ovšem správa dynamických IV vektorů je prováděna protokolem TKIP (Temporal Key Integrity Protocol), takže IV vektor se už nedá odposlechnout z nešifrovaného záhlaví rámce. Navíc protokol TKIP přidává k rámci 64bitový kontrolní součet (spíše digitální podpis rámce), který nazýváme MIC (Message Integrity Check, lidově „Michael“), podle kterého se pozná, jestli nebyl rámec cestou poškozen či záměrně upraven.

WPA pracuje ve dvou variantách:

- WPA-Enterprise – pro korporátní sféru, vpodstatě pokračovatel WEP+RADIUS, v síti musí být autentizační server,
- WPA-Personal (WPA-PSK) pro domácnosti a malé firmy, pro autentizaci se používá sdílený klíč (PSK = Pre-Shared Key), nemáme autentizační server, ale uživatel i tak potřebuje

autentizační informace (heslo 8–64 znaků nebo 64 hexadecimálních číslic), ze kterých se dynamicky určuje klíč.

WPA2 (IEEE 802.11i). Pro tento mechanismus existuje vlastní standard. Oproti WPA se místo protokolu TKIP používá protokol CCMP (Counter-Mode CBC MAC Protocol) s šifrováním AES, a právě použití šifrování AES vyžaduje hardwarovou podporu na zařízeních (dnes již většina procesorů obsahuje AES instrukce, takže obvykle nebývá problém se zprovozněním).

Stejně jako u WPA, i zde potřebuje uživatel přihlašovací údaje, které jsou využity buď pro autentizační server nebo pro protokol PSK, tedy opět máme dvě možnosti:

- WPA2-Enterprise – pro korporátní sféru, s autentizačním serverem,
- WPA2-Personal (WPA2-PSK) pro domácnosti a malé firmy.

V roce 2017 se objevila zpráva, že mechanismus WPA2 je náchylný na útok KRACK (Key Reinstallation Attacks), pomocí kterého lze obejít zabezpečení pomocí šifrování spojení.

WPA3 byl uveden roku 2018. Opět existují dva režimy – WPA3-Enterprise a WPA3-Personal. Pro Enterprise použití se předepisuje šifrování AES-256 v módu GCM s hashováním HMAC SHA-384. V Personal režimu lze použít AES-128 s protokolem CCMP jako u WPA2, ale výměna klíčů probíhá s využitím algoritmu SAE (Simultaneous Authentication of Equals), což je varianta výměny klíčů Dragonfly Key Exchange (funguje podobně jako Diffie-Hellman).

Příchod WPA3 byl uspíšen objevením závažných bezpečnostních chyb ve WPA2, nicméně bezpečnostním chybám se nevyhnul ani samotný mechanismus WPA3. Například výše zmíněný algoritmus Dragonfly (ve WPA3 se používá při navázání spojení pro bezpečnou výměnu klíčů) má bezpečnostní slabinu DragonBlood, která je zneužitelná rovnou několika různými způsoby.

Další informace:

- https://www.root.cz/clanky/sifrovani-wpa2-bylo-prolomeno-wi-fi-site-je-mozne-odposlouchavat/
- https://www.root.cz/zpravicky/dragonblood-bezpecnostni-chyba-ve-wpa3-umoznuje-ziskat-heslo-k-wi-fi/
- https://www.wi-fi.org/discover-wi-fi/security
- https://medium.com/@reliancegcs/wpa3-explained-wi-fi-is-getting-major-security-update-2b6dca8f3aff

7.3.2 WPS

Mechanismus WPS (Wi-fi Protected Setup) se objevil s tím, jak přibývalo nejrůznějších typů zařízení, která se měla připojovat do Wi-fi sítě (zařízení internetu věcí, „chytrá“ domácnost, W-fi extendery, atd.). Mnohá zařízení nemají klávesnici ani pořádný display, takže případné zadávání hesel a klíčů by bylo problematické. WPS právě slouží ke zjednodušené autentizaci a asociaci takovýchto zařízení do Wi-fi sítě.

Existují dvě možnosti použití mechanismus WPS – WPS tlačítko nebo WPS PIN.

WPS tlačítko (PBC – Push Button). Na AP zmáčkneme tlačítko označené WPS (nebo QSS či PBC) nebo v administraci AP klepneme na „softwarové“ tlačítko. V průběhu několika desítkem sekund AP skenuje svou buňku a hledá nová zařízení. Pokud se v tomto časovém intervalu (kterékoliv) zařízení v buňce začne hlásit (čehož docílíme tak, že na na dotyčném zařízení taky

zmáčkneme tlačítko nebo provedeme něco podobného podle návodu), je považováno za důvěryhodné a AP s ním dojedná asociaci (takže nemusíme zadávat žádná hesla).

Problémy jsou především tyto:

- V uvedeném časovém intervalu je naše síť prakticky nechráněná, cokoliv se může přihlásit a dostat se dovnitř.
- Musíme stihnout na asociovaném zařízení včas zmáčknout tlačítko, takže v některých případech to je spíše práce pro dva.

✎ **WPS PIN.** Pro zařízení, které chceme takto dostat do sítě, musíme zjistit jeho PIN – 8místné číslo, které najdeme buď na nálepce přímo na zařízení nebo někde v dokumentaci. PIN pak naťukáme v administraci svého AP, načež si opět AP se zařízením vyjednají veškeré parametry a provede se asociace.

Ovšem problémy jsou i s touto metodou:

- AP s podporou WPS PIN *naslouchá neustále*.
- PIN je 8místné číslo, přičemž poslední číslice se vypočítává z předchozích, je obdobou kontrolního součtu. Pokud PIN do administrace zadáme špatně, AP vrací informaci, ze které se dá odvodit, která polovina je chybná – pokud je chybná první polovina, stačí uhodnout tři číslice, což je pro průměrného hackera časově nenáročná hra.

Pokud náš Wi-fi AP má tuto funkci, je lepší ji vypnout a zapínat opravdu jen tehdy, když chceme takto asociovat nové zařízení.

7.3.3 Jak tedy zabezpečit bezdrátovou síť

Takže co udělat, aby naše Wi-fi síť byla co nejlépe zabezpečená? Předně nastavit šifrování WPA2, pokud to všichni naši klienti zvládají. A co dál?

✎ **Credentials do administrace.** Přístupové údaje do administrace AP jsou z výroby nastaveny na určité standardní hodnoty (typicky admin–admin nebo něco takového). V žádném případě bychom to tak neměli nechat! Takže hned po zakoupení během počáteční konfigurace bychom jako první měli nastavit vlastní přístupové údaje do administrace AP.

✎ **Skrytí SSID.** AP vysílá v pravidelných intervalech Beacon rámec, kde je mimo jiné SSID (tedy název sítě). SSID je potřeba pro přihlášení do sítě, a tedy když AP nebude vysílat Beacon rámce s SSID, teoreticky bychom tím zabránili pokusům o přihlášení osob, které v síti nemají co dělat. Prakticky se však SSID dá odposlechnout v autentizační komunikaci „povolených“ zařízení, takže stačí donutit některé takové zařízení k znovupřipojení (shodíme jeho komunikaci).

Navíc skrytí SSID může způsobovat problémy při hledání zdroje rušení jiné sítě – může nastat situace, kdy vidíme, že nám něco ruší síť, ale nevidíme, co to je, protože dotyčný má skryté SSID.

✎ **Filtrování MAC adres.** V administraci AP si můžeme vytvořit black list (seznam zakázaných) nebo white list (seznam povolených, nikdo jiný nebude asociován) MAC adres. Ale stačí si uvědomit, že MAC adresa se dá pozměnit (v Linuxu na to stačí jeden příkaz, ve Windows úprava registru nebo speciální program), takže opět bezzubé opatření.

Kapitola 8

Rozlehlé sítě a přístupové sítě

WAN (Wide Area Network) je rozlehlá síť zabírající obvykle rozlohu státu, kontinentu či celého světa. Propojuje nikoliv koncová zařízení, ale různé sítě typu LAN, MAN či menší WAN sítě.

Přístupové sítě slouží k napojení klientských lokálních sítí do MAN/WAN sítí. Existují různé druhy přístupových sítí, my se zaměříme na přístupové sítě využívající telekomunikační infrastrukturu. Pro přístup k WAN mohou sloužit i další typy sítí, například bezdrátová metropolitní síť WiMAX také může být považována za přístupovou síť, a přístup do mobilních sítí (GPRS, LTE apod.) je víceméně zprostředkováván samotnými WAN sítěmi.

Rychlý náhled: Tato kapitola je úvodem k WAN sítím a přístupovým sítím. V prvních dvou sekcích najdeme stručný popis typické fyzické a komunikační struktury WAN sítí a přehled nejznámějších WAN sítí. Následuje úvod do telekomunikačních sítí a dále charakteristika přístupových sítí ADSL a VDSL.

Klíčová slova: WAN, okruh, DTE, DCE, SLA, spojový protokol, i-frame, s-frame, u-frame, Frame Relay, DLCI, ADM, VPI, VCI, MPLS, label, telekomunikační síť, MTO, UTO, TTO, PBX, PSTN, POTS, ADSL, VDSL, modem, splitter, agregace, DSLAM

Cíle studia: Po prostudování této kapitoly porozumíte principu fungování WAN sítí a zjistíte, jak pracují aktivní síťové prvky v nejběžnějších typech WAN sítí. Také se naučíte principu činnosti běžných telekomunikačních přístupových sítí.

8.1 WAN sítě

8.1.1 Struktura WAN sítí

Jakou topologii vlastně používají WAN sítě? Záleží, jaký pohled použijeme.

- WAN sítě jsou navzájem uspořádány do (převážně) hierarchické struktury velmi podobné hierarchii podle protokolu IP. Souvisí to s tím, že lokální a regionální poskytovatelé konektivity tuto konektivitu také odněkud musejí získat, od velkých nadnárodních poskytovatelů. Celková struktura WAN sítí (při pohledu zvenčí) má charakter *páteřní sítě*.

- Když se podíváme na vnitřní strukturu WAN sítě, obvykle zjistíme, že používá architekturu *mesh*: redundantní spoje, které zajišťují vysokou dostupnost, odolnost proti výpadku.

Zatímco u lokálních sítí víceméně platí, že data po těchto sítích posílaná souvisejí s vlastníkem sítě, u MAN a zejména WAN sítí tomu tak není. Provozovatelé WAN sítí jsou prostě majiteli síťové infrastruktury, kterou pronajímají svým zákazníkům, je to páteř, na kterou jsou napojeny zákaznické sítě.

Síťová konektivita (případně v určité definované kvalitě) je jedinou službou, kterou WAN poskytují, funkčnost je podřízena také tomu, aby bylo možné poskytnuté služby tarifikovat. Dokonce může být problém se směrováním na jiné než unicast adresy, protože pro WAN je typická spojovaná komunikace přes virtuální okruhy.

Ve WAN sítích obvykle rozlišujeme tyto typy zařízení:

- *DCE (Data Circuit Equipment)* jsou zařízení uvnitř WAN sítě (jádro). Obvykle přímo nekomunikují s ničím, co by do WAN nepatřilo, prostě dokážou přes sebe navázat komunikační okruh (circuit) a pak přes něj přenášet data. Pro zákazníka jsou „neviditelné“. Jejich typickou vlastností je rychlost přepínání dat (obvykle s hardwarovou podporou), je to obdoba switchů v LAN sítích.
- *DTE (Data Terminal Equipment)* jsou zařízení na rozhraní mezi WAN sítí a tím, co je k ní připojeno, jsou to tedy hraniční zařízení. Potřebují „rozumět“ jak protokolům pracujícím ve WAN, tak i protokolům z připojené sítě (vzhledem k připojené síti obvykle potřebují funkcionalitu vrstvy L3, IP). Pracují jako překladatelé mezi dvěma druhy sítí. Paket příchozí „zvenčí“ oklasifikují, určí adresu, která bude používána ve WAN síti, přidají záhlaví podle protokolu používaného ve WAN síti a pošlou vzniklou PDU dovnitř.

Poznámka:

V některých WAN se používá trochu jiná terminologie:

- zařízení P (Provider) odpovídají zařízením DCE, jsou tedy uvnitř WAN sítě,
- zařízení PE (Provider Edge – na hranici poskytovatele) jsou na hranicích WAN sítě,
- zařízení CE (Customer Edge – na hranici zákazníka) jsou na straně zákazníka a zprostředkovávají pro něj přístup k WAN síti.

Zařízení typu PE a CE tedy mohou dohromady tvořit DTE.

U pronajímaných sítí včetně WAN a MAN uzavírá zákazník s pronajímatelem smlouvu označovanou zkratkou *SLA* (Service Level Agreement) – dohodu o úrovni služby, ve které jsou stanoveny dohodnuté parametry včetně garance datového toku, kvality služby.

8.1.2 Protokoly vrstvy L2 pro WAN sítě

Pro protokoly pracující na vrstvě L2 se vžil název *spojové protokoly*. Kromě těch, které jsme probírali dříve, zde najdeme také protokoly pro WAN sítě (protože většina WAN sítí je implementována především na této vrstvě).

Obvykle platí, že každá WAN síť má „svůj“ spojový protokol, nicméně u těchto protokolů můžeme vysledovat určité společné vlastnosti (protože ve skutečnosti jsou mezi nimi vztahy předek-potomek).

✎ **Rámce spojového protokolu.** Rozlišujeme několik typů rámců:

- *I-frame* (Information Frame, informační, datový) – nese informace z vyšší vrstvy a případně řídicí informace, tyto rámce podporují řazení, řízení toku, detekci chyb, zotavení. Určení: přenos dat.
- *S-frame* (Supervisory Frame, také služební či dohlížecí rámec) – obsahuje řídicí informace, je používán pro požadavek zahájení nebo ukončení spojení, hlášení o stavu, potvrzení přijetí I-rámce.
- *U-frame* (Unnumbered Frame, nečíslovaný) – pro řídicí informace (dohoda režimu provozu), na rozdíl od předchozího nepodporují číslování rámců do posloupnosti, a může také obsahovat data z vyšší vrstvy. Obecně: pro to, co se vejde do jediného rámce.

Rámce typu I-frame používají čísla *Sequence Number* pro oba směry (podobný mechanismus jako u TCP), S-frame mají čísla pouze pro jeden směr (přenášený obsah má totiž smysl pouze pro jeden ze směrů komunikace), U-frame nepoužívá žádná taková čísla (je nečíslovaný, nedělí komunikaci na části, které by bylo třeba kompletovat).

8.2 Nejznámější WAN sítě

8.2.1 Frame Relay

Specifikace Frame Relay původně vznikla v rámci specifikace ISDN, ale protože se ukázala dobrá životaschopnost tohoto řešení, v roce 1989 se osamostatnila.

Na vrstvě L2 pracuje protokol LAPF (ten je přenášen přes WAN síť), do rámce LAPF se zapouzdřuje IP paket ze sítě zákazníka. Komunikuje se po virtuálních okruzích, obvykle přepínaných (SVC).

Obecně platí, že typickým použitím Frame Relay je z principu propojování lokálních sítí, implementace páteřní sítě. Počítá se s přenosem po kvalitním a rychlém vedení, tedy Frame Relay poskytuje sice službu se spojením, ale nespolehlivou (poskytuje detekci chyb, ale bez možnosti opravy).

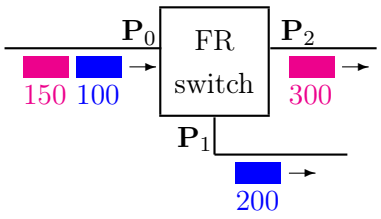
✎ **Adresace a přepínací tabulky.** Po navázání okruhu se používá lokální adresace pomocí *čísel DLCI* (Data Link Connection Identifier). Ve skutečnosti nejde o identifikátory zařízení, jsou to identifikátory okruhů. Hodnoty DLCI pro DTE v naší LAN získáme od poskytovatele služby Frame Relay (typicky telekomunikační společnost), samozřejmě pokud si FremeRelay implementujeme sami, tak si DLCI taky sami přidělíme.

V celé síti WAN (propojující různé lokální sítě) však může existovat více okruhů se stejným DLCI, navíc se hodnota DLCI při průchodu přes přepínače mění, na každém konci mívá okruh jiné DLCI.

Některé hodnoty DLCI jsou vyhrazeny pro účely signalizace stavů, problémů apod., především z rozsahu 0–16. Dále rozmezí 1019–1022 je určeno pro skupinové vysílání.

Na jednotlivých zařízeních jsou rámce přepínány podle *přepínacích tabulek*. Uvnitř FR sítě najdeme FR switche (to jsou DCE), jejich tabulka je vcelku jednoduchá, jak vidíme na obrázku 8.1 (případně by byl ještě další sloupec – informace o tom, zda je daná cesta aktivní).

	Přijato z		Přepnout na		
	IN_Port	IN_DLCI	OUT_Port	OUT_DLCI	
■	P_0	100	P_1	200	■
■	P_0	150	P_2	300	■
	⋮				



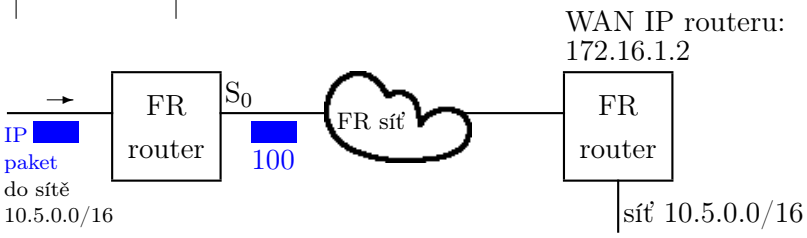
Obrázek 8.1: Ukázka jednoduché přepínací tabulky Frame Relay switche

Routing Table:

Network	Next Router	Interface
10.5.0.0/16	172.16.1.2	S_0
10.27.0.0/16	172.16.1.8	S_1
⋮		

FR Map:

Next router	DLCI
172.16.1.2	100
172.16.1.8	200
⋮	



Obrázek 8.2: Směrování na FR routeru (DTE), paket přichází do FR sítě

Díky jednoduchosti mechanismu je přepínání velmi rychlé.

Na hranici sítě jsou FR routery (v roli DTE), které na jedné straně komunikují se zařízeními lokální sítě zákazníka, na druhé straně komunikují s FR switchi. Tato zařízení obsahují dvě tabulky:

- směrovací tabulku určující, kam se rámec směřující do sítě s danou IP adresou má poslat,
- tabulku mapování DLCI.

Obsah a význam obou tabulek vidíme na obrázku 8.2. Na třetí vrstvě je paket nasměrován na daný DCE (na druhé straně FR sítě) a příslušné rozhraní, na druhé vrstvě je nastaveno DLCI.

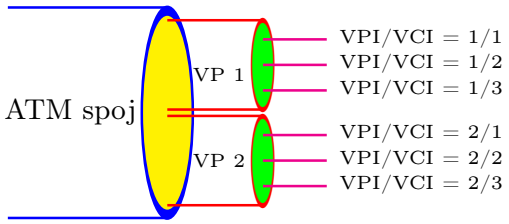
8.2.2 ATM

Sítě ATM (Asynchronous Transfer Mode) byly původně považovány za velmi nadějnou technologii, nicméně v praxi byly postupně vytlačeny jednodušími a pružnějšími sítěmi MPLS. „Asynchronní“ v názvu neznamená asynchronní přenos, ale nepravidelné zasílání „plných“ buněk na cestě, buňky jsou obsazovány daty podle potřeby, ne podle algoritmu.

ATM pracuje na principu komunikace se spojením, nespolehlivá služba (tj. bez oznámení chyb), na plném duplexu, v statistickém multiplexu. Důležitou vlastností je garance kvality služeb pro

různé typy dat – vlastně jednou z nejdůležitějších charakteristik ATM jsou právě široké možnosti řízení kvality služeb.

Adresace okruhů je na rozdíl od FR dvouhodnotová. *Virtuální cesta* a *virtuální kanál* jsou obdobou DLCI. V jedné fyzické přenosové cestě může vést více virtuálních cest, v jedné virtuální cestě vede víc virtuálních kanálů. Číslo virtuální cesty označujeme VPI (Virtual Path Identifier), číslo virtuálního kanálu označujeme VCI (Virtual Channel Identifier). *Hodnota VPI/VCI* plně identifikuje přenosovou cestu mezi dvěma sousedními uzly, na každém ATM přepínači se mění.



Obrázek 8.3: Virtuální cesty a virtuální kanály v ATM (zjednodušeně)

ATM switch si vede *tabulku připojení*, ve které jsou přidruženy příchozí a odchozí VPI/VCI, záznam se tvoří během navázání spojení (vytvoření okruhu).

Na obrázku 8.4 je zjednodušená a zkrácená přepínací tabulka ATM switche. Jsou v ní tři virtuální cesty. Každý řádek obsahuje jednu přepínací informaci, například v prvním řádku zjistíme, že buňka přicházející z portu A po cestě 1 v kanálu 29 je přepnuta na port B cestu 2 kanál 42.

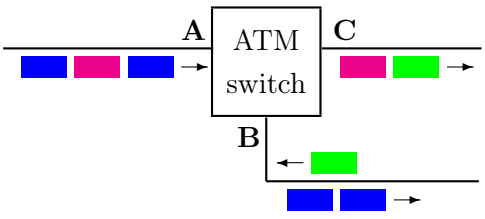
PDU se nazývají *buňky*, jsou velmi malé s konstantní délkou 53 oktetů (z důvodu jednoduchého a rychlého přepínání). Aby se do buněk vůbec vešla nějaká data, je záhlaví velmi malé, pouze 5 B, na data zbývá 48 oktetů. V záhlaví máme kromě adres VPI/VCI také například jednobitový příznak CLT (Cell Loss Priority, priorita ztráty buňky, 1 bit), který plní stejnou roli jako u FR bit DE.

Protože se komunikuje v statistickém multiplexu, jsou v přenosu definovány sloty, do kterých se „sázejí“ buňky. Zde je právě výhodou, že buňky mají konstantní velikost odpovídající velikosti slotů.

8.2.3 MPLS

MPLS (MultiProtocol Label Switching) je síť založená na *přepínání značek* (label, také návěští). Účelem je přesunout co nejvíce režie směrování (včetně administrace, QoS apod.) na okraj sítě

	Přijato z			Přepnout na			
	Port	VPI	VCI	Port	VPI	VCI	
■	A	1	29	B	2	42	■
■	A	3	42	C	1	8	■
■	B	1	18	C	2	36	■
	⋮						



Obrázek 8.4: Ukázka jednoduché přepínací tabulky ATM switche

tak, aby vnitřní oblast sítě byla co nejrychlejší. MPLS dokáže velmi rychle přenášet nejen běžná data, ale také hlas a video, a to se zajištěním QoS. K výhodám sítě ATM se tedy přidává pružnost a vyšší rychlost. Další výhodou je snadnější implementace virtuálních sítí.

Obrovskou výhodou MPLS je mnohotvárnost. Nemá přímo definovánu adresaci ani směrování, (možná proto) dokáže spolupracovat s více odlišnými protokoly. Výhody MPLS můžeme shrnout takto:

- rychlost přepínání,
- zajišťování řízení provozu (vyvažování zátěže) a QoS (odlišné zacházení s různými PDU),
- podpora VPN,
- schopnost spolupráce s mnoha různými protokoly, pod MPLS mohou fungovat různé technologie.

V ISO/OSI modelu bychom mohli MPLS zařadit někam mezi druhou (spojovou) a třetí (síťovou) vrstvu, také se označuje jako vrstva 2.5 nebo 2+.

Technologie byla představena firmou Ipsilon Networks pod názvem *IP Switching*. Později společnost Cisco vytvořila proprietární standard *Tag Switching*, který síť zbavil závislosti na ATM, podobný, ale otevřený standard později vydalo sdružení IETF.

Každý paket, který vstupuje do MPLS sítě, je na okrajovém směrovači opatřen jedním nebo více *MPLS záhlavími* uspořádanými do *zásobníku* (stack), a to na rozhraní mezi záhlavími druhé a třetí vrstvy. Záhlaví protokolu MPLS má tuto strukturu:

- samotná značka (návěští, label, 20 bitů),
- Qos informace (3 bity), v případě MPLS se setkáme s názvem CoS (Cathegory nebo Class of Service) nebo také ve významu Experimental (Exp.),
- příznak konce zásobníku (1 bit); pokud nenásleduje další záhlaví, je nastaven na 1 (to znamená, že následuje už přímo záhlaví IP paketu),
- TTL (Time to Live, 8 bitů).

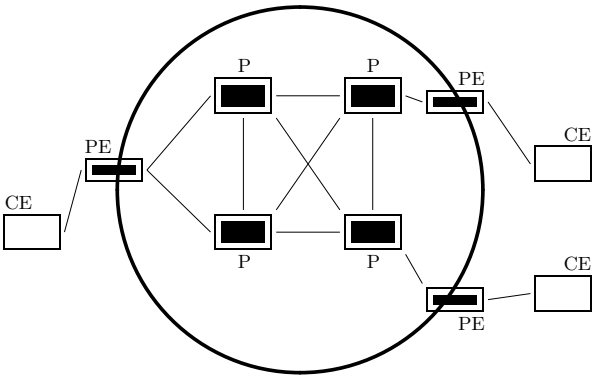
Značka uložená v MPLS záhlaví jednoznačně určuje směrování paketu na cestě k následujícímu směrovači, v podobném smyslu jako DLCI u FR nebo VPI/VCI u ATM.

Záhlaví rámce 2. vrstvy	MPLS Label 1, ES bit = 0	MPLS Label 2, ES bit = 0	MPLS Label 3, ES bit = 1	Záhlaví paketu 3. vrstvy (např. IP)	Datová část paketu

Tabulka 8.1: Zásobník značek v MPLS

Vnější (nejvrchnější) záhlaví na zásobníku je určeno právě ke stanovení cest. Nenabízí vpodstatě o moc víc než IP záhlaví (s jedním důležitým rozdílem – rychleji se zpracovává). Další záhlaví hlouběji v zásobníku mají trochu jiný, specifický, účel, například záhlaví VPN pro určení virtuální sítě, záhlaví pro QoS nebo záhlaví pro řízení provozu.

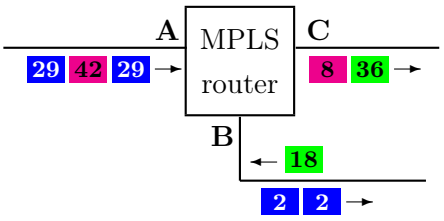
Na obrázku 8.5 je naznačena obvyklá struktura MPLS sítě s uzly P, PE a CE. CE je součástí lokální sítě zákazníka, k jednomu PE může být připojeno více CE (tj. více LAN).



Obrázek 8.5: Struktura MPLS sítě

Směrování (přepínání značek) probíhá podobně jako v ATM, ale ještě jednodušeji. Směrovače si vedou jednoduchou *tabulku*, která je podobná tabulce pro ATM, ale místo dvojice VPI/VCI je zde pouze hodnota značky. Jedná se jen o určení vazby mezi příchozí a odchozí značkou, cesty jsou jednosměrné.

	In port	In label	Prefix adresy	Out label	Out port	
■	A	29	128.95.0.0/16	2	B	■
■	A	42	128.101.0.0/16	8	C	■
■	B	18	141.62.0.0/16	36	C	■
	⋮					



Obrázek 8.6: Ukázka jednoduché přepínací tabulky MPLS směrovače

Na obrázku 8.6 je ukázka tabulky na vnitřním uzlu sítě. Značka uložená v MPLS záhlaví se při průchodu směrovači neustále mění, podle toho, jak to bylo nastaveno při navazování spojení, podle obsahu tabulek na průchozích uzlů. Tento proces se nazývá *Label Swapping* (výměna značek/návěští).

8.3 Telekomunikační sítě

Telekomunikační (telefonní) síť je řešena hierarchicky:

- *účastnické zařízení* (telefon, modem apod.), připojuje se přes *účastnickou přípojku*,
- (veřejná) *místní telefonní ústředna* (MTO) – v pravidelných intervalech na ulicích,
- *uzlová telefonní ústředna* (UTO), *tranzitní telefonní ústředna* (TTO).

V rámci firemního areálu může také fungovat *pobočková ústředna* (PBX, Private Branch Exchange), která bývá připojena na některou veřejnou místní telefonní ústřednu (tj. funguje jako brána).

Při přenosu dat přes telefonní síť, dimenzovanou pro přenos zvuku (především hlasu), bylo hlavní motivací využití již existující husté sítě těchto linek, tedy snadná dostupnost koncových bodů.

✎ **Pojmy.** Abychom snáze pochopili následující text, vysvětlíme si několik pojmů souvisejících s telekomunikační sítí.

- PSTN (Public Switched Telephone Network) – běžná telefonní síť dimenzovaná pro přenos hlasu, původně čistě analogová s frekvenčním multiplexem, po digitalizaci se používá časový multiplex s rezervací komunikačního pásma. Pozor, digitalizace se týká především telekomunikačních ústředen, část spoje u zákazníka je analogová.

  Po digitalizaci bylo nutno (v principu analogový) hlas v MTO digitalizovat, používal se některý kodek, tedy CODEC (COder-DECoder). Podobný princip se dnes používá i ve VoIP přímo v koncovém zařízení.
- POTS (Plain Old Telephone Service) je technologie využívající PSTN pro přenos analogového hlasu a digitálních dat. Protože PSTN je zkraje analogová a až od ústředny digitální, je třeba digitální data modulovat na analogový signál, používá se MODEM (MODulator-DEModulator).
- Modem je tedy zařízení, které moduluje digitální signál na analogový a v cíli ho zpětně demoduluje do digitální podoby. Modemy byly dříve analogové, dnes používáme digitální modemy (pro ADSL, VDSL apod.) – telekomunikační linka na straně zákazníka je prostě pořád analogová.
- ISDN (Integrated Services Digital Network) byla prvním pokusem o digitalizaci přístupových sítí, ale tak trochu na půl cesty. Šlo o vytáčené připojení (tarifikace podle času), protože správa sítě stavěla na původní POTS, vlastně to byla jen digitální nástavba POTS.
- xDSL je skupina technologií, které se oprostily od většiny součástí POTS, z telekomunikační sítě se využívá pouze spoj mezi účastnickou přípojkou a nejbližší ústřednou MTO. Pak komunikační cesta odbočuje do plně digitální počítačové WAN sítě poskytovatele.

8.4 Přístupové sítě

Přístupové sítě jsou sítě sloužící pouze a jenom k napojení lokálních (nebo jiných menších) sítí na WAN/MAN sítě (třeba patřící některému ISP) a zprostředkovaně do Internetu. Také hovoříme o sítích *první míle* (First Mile) nebo *poslední míle* (Last Mile), což vlastně znamená jakousi hranici mezi menší sítí a WAN/MAN sítí. V anglické literatuře se spíše setkáme s „optimističtějším“ Fist Mile.

Pro přístupové sítě existují různé technologie – může jít o bezdrátové sítě (například WiMAX) nebo telekomunikační sítě (většinou ADSL/VDSL nebo podobná technologie), v některých případech se přístupová síť dokonce jeví jako speciální součást WAN sítě (mobilní sítě, třeba LTE).

Specifikem přístupové sítě je, že spojuje velice odlišné technologie vzhledem ke způsobu komunikace (v LAN sítích máme typicky paketový přenos dat, kdežto ve WAN sítích se komunikuje přes okruhy), a také vzhledem k fyzické vrstvě a zacházení se signálem.

✎ *xDSL* (Digital Subscriber Line) je skupina širokopásmových technologií (broadband, používá se multiplexování), která v sobě sdružuje víc různých technologií (ADSL, SDSL, HDSL, HDSL-2, G.SHDL, IDSL, VDSL). Jedná se o vyhrazenou službu, přenos typu point-to-point.

Je to služba se spojením, ale tarifikace není závislá na čase.

8.4.1 ADSL

ADSL (Asymmetric Digital Subscriber Line) je asymetrická služba. Asymetrická proto, že downstream (stahování dat do DTE) je rychlejší než upstream (odesílání dat do sítě).

Teoreticky lze dosáhnout rychlostí až 24 Mb/s (v novějším standardu dokonce až 48 Mb/s). U nás nabízená rychlost je 8 Mb/s nebo 16 Mb/s *pro downstream*, ale reálná rychlost může být nižší. Upstream bývá na rychlosti 1 Mb/s.

Rychlost je ovlivněna více různými kritérii. Je to nejen použité zařízení a přenosové protokoly, ale také délka spoje – čím větší vzdálenost k místní ústředně, tím pomalejší spojení.

Rozsah	Šířka	Účel
0 kHz – 4 kHz	4 kHz	přenos hlasu (telefon)
4 kHz – 26 kHz	22 kHz	nárazníkové pásmo
26 kHz – 138 kHz	112 kHz	upstream
138 kHz – 1100 kHz	962 kHz	downstream

Tabulka 8.2: Využití přenosového pásma v ADSL

V technologii ADSL (a taky v dalších xDSL technologiích) se navýšení rychlosti dosahuje kromě jiného i odkloněním datového toku z telefonních linek na datovou síť poskytovatele (obvykle některou WAN síť na optických kabelech).

✎ **Agregace** = sdílení přípojky ADSL. Každý ISP rozdělí mezi své zákazníky kapacitu linky, kterou má k dispozici, formou časového multiplexu. Vzorec pro agregační poměr je následující:

$$\textit{agregační poměr} = \frac{\textit{rychlost}(ISP)}{\sum_i \textit{rychlost}(i)}$$

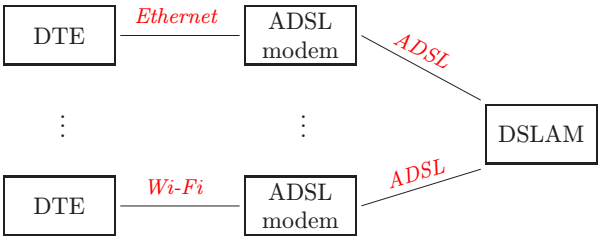
Typické hodnoty agregace mohou být například $1:50,\ 1:20$.

✎ **Zařízení v síti ADSL.** Když nepočítáme koncová zařízení v síti zákazníka (která ve skutečnosti do sítě ADSL vůbec nepatří), v ADSL se používají tato zařízení:

- *ADSL modem* (MODulator/DEModulator) – moduluje digitální signál na analogový (šířka přibližně 1,1 MHz) a zpětně demoduluje, připojují se k němu datová koncová zařízení,
- *splitter* – sloučení analogového hlasového přenosu a modulovaných dat, připojuje se k němu modem a hlasová koncová zařízení (analogová),
- *DSLAM* (DSL Access Multiplexer) – na straně ISP, sdružuje spojení z ISP splitterů pro jednotlivá připojení.

V praxi je u účastnických přípojek modem a splitter v jednom zařízení (interní, integrovaný), na straně ISP bývají odděleny. Pokud nepoužíváme žádná analogová zařízení (analogový telefon), což je dnes běžný stav, měli bychom se splitteru zbavit, protože může být zdrojem mírných posunů signálu a tím i poruch. Digitální telefony samozřejmě ke splitteru nepřipojujeme!

✎ *ADSL modem* je v SOHO (Small Office, Home Office) často jen modulem v složitějším zařízení, například jsou běžné ADSL Wi-fi routery (tj. ADSL modem s Wi-fi routerem plus další funkcionalita: hardwarový firewall, DHCP server, atd.). Připojuje se k účastnické přípojce přes RJ-11 (telefonní rozhraní).



Obrázek 8.7: ADSL je technologie první míle

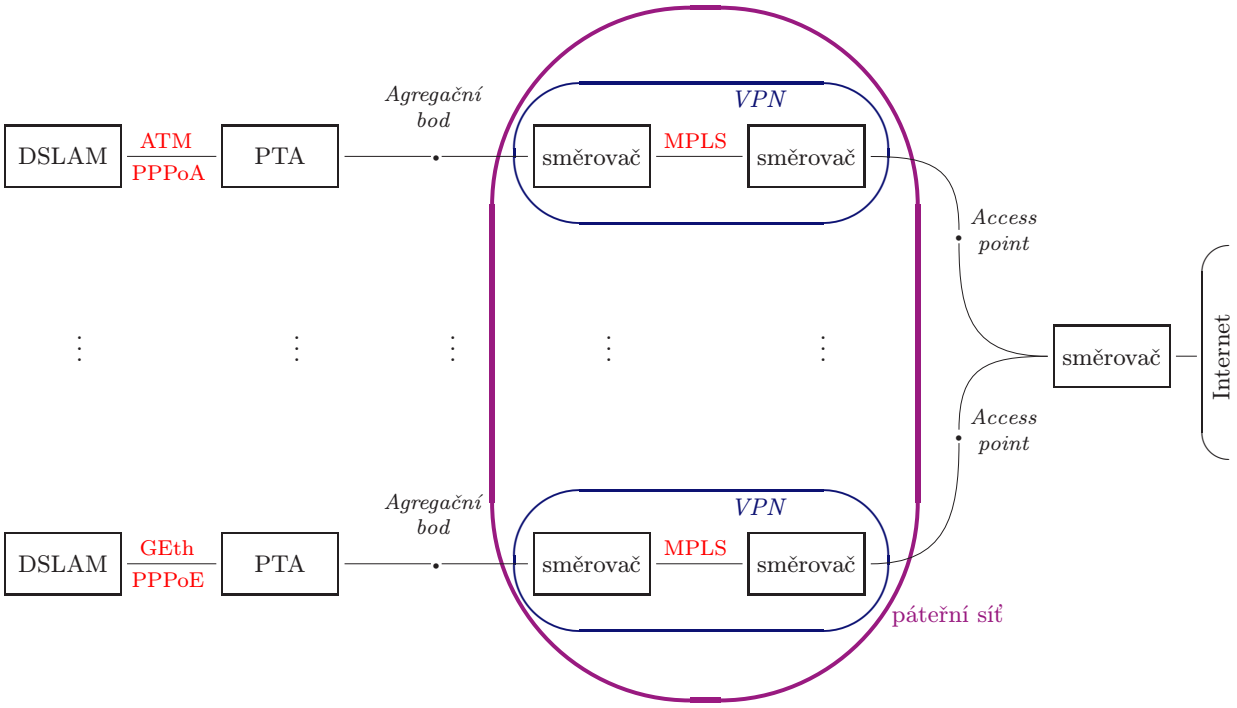
Jeho funkce jsou jak navazování spojení (obvykle při zapnutí nebo resetu přístroje), tak i zajišťování samotné komunikace. Pakety, které je třeba odeslat z lokální sítě přes přístupovou síť, zapouzdřuje do PDU protokolu PPPoE, PPPoA nebo IPoA (ten první je dnes nejběžnější).

DSLAM je brána do datové sítě poskytovatele. Existují menší DSLAMy pro připojení 24 nebo 48 DSL linek, a pak větší pro tisíce linek. Obvykle podporuje protokoly PPPoE i PPPoA.

DSLAM má rozhraní RJ-11 („telefonní“) směrem k zákazníkovi a pak další rozhraní – většinou dvě RJ-45 pro Gigabit Ethernet nebo rychlejší, případně optiku. Administrace se provádí přes webové rozhraní, anebo přes konzoli, záleží na konkrétním výrobci.

Na obrázku 8.8 je zjednodušený nákres struktury sítě na straně poskytovatele Internetu. Pojmy k tomuto obrázku:

- *PTA* je širokopásmový server, provádí se zde konfigurace IP adres, autentifikace uživatelů, autorizace, účtování (RADIUS),
- *AP (Access Point)* je přístupový bod pro konkrétního ISP, přes který lze přistupovat z virtuální sítě tohoto ISP k směrovači na Internet.

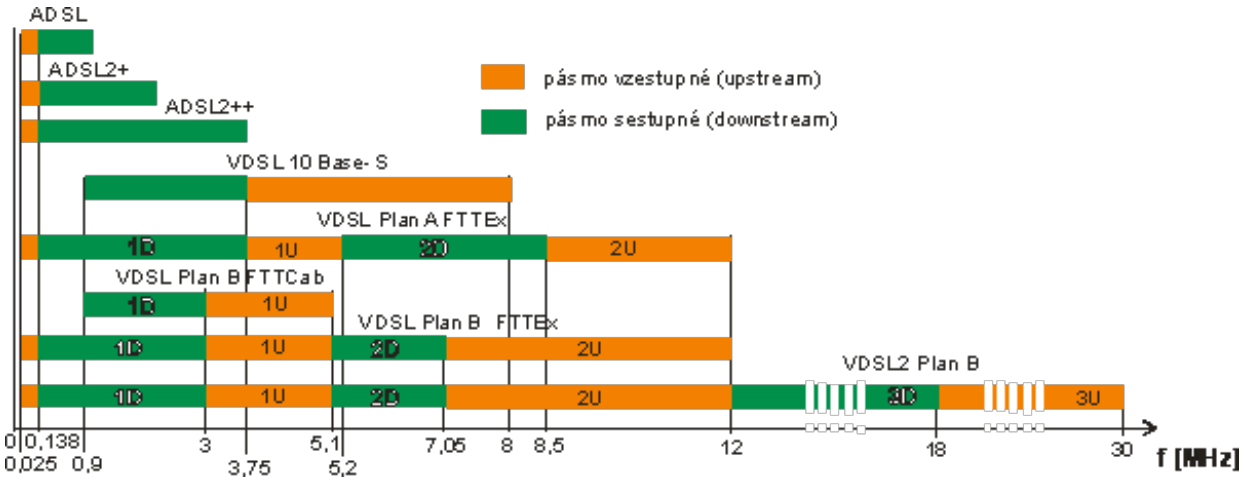


Obrázek 8.8: ADSL – na straně ISP

8.4.2 VDSL

VDSL (Very High Bit-Rate Digital Subscriber Line) je považována za následníka ADSL. Je také asymetrická (může být nastavena na symetrický přenos), používá širší pásmo než ADSL (protáhnutí na vyšší frekvence), a z toho důvodu je rychlejší. Rychlost na *kvalitní* telefonní UTP dosahuje až 52 Mb/s (asymetrická, downstream) nebo 36 Mb/s (symetrická varianta). Teoreticky až 200 Mb/s, reálně opravdu jen kolem 50 Mb/s.

Z obrázku 8.9 je patrné, že oproti ADSL technologie VDSL využívá širší frekvenční pásmo, tedy je k dispozici více komunikačních kanálů. Zatímco u ADSL je na nižších frekvencích upstream a na vyšších downstream, u VDSL jsou kanály pro oba směry rozloženy celkem rovnoměrně v celém spektru (důsledkem je, že zatímco u ADSL při přechodu na novou verzi využívající vyšší frekvence byl navýšen jen downstream, protože upstream z důvodů kompatibility měl pásmo pevně stanoveno, u VDSL mohou být navyšovány rychlosti v obou směrech).



Obrázek 8.9: Porovnání frekvencí ADSL a VDSL[1]

Zatímco ADSL provádí přenos na vzdálenost až 5 km (při vyšších rychlostech méně), VDSL pouze něco přes 1 km, což je daň za rozšíření pásma. Ovšem pokud chceme opravdu využít vysoké rychlosti nabízené VDSL, tak bychom měli být maximálně 500 m od DSLAMu. Upstream a downstream jsou odděleny frekvenčním multiplexem podobně jako u ADSL.

Rozšíření VDSL závisí na telekomunikačních organizacích, tato technologie musí být podporována v telekomunikačních ústřednách.

Poznámka:

Někteří telefonisté telekomunikačních operátorů zákazníkům tvrdí, že navýšení rychlosti na VDSL spoji (samozřejmě za příplatek) zrychlí taky lokální Wi-fi síť u zákazníka. To je hloupost – pokud máme z nějakého důvodu pomalou Wi-fi síť (ať už kvůli slabým anténám s nízkým výkonem, špatnému nasměrování antén, staršímu standardu, rušení nebo velkému množství klientů), navýšení rychlosti na VDSL spoji to nezmění, jen budeme více platit.

[1] Zdroj: http://access.feld.cvut.cz/view.php?cisloclanku=2004120302

Kapitola 9

Bezpečnost

Rychlý náhled: Tato kapitola obsahuje směs témat orientovaných na bezpečnost. Nejdřív se podíváme na několik základních typů útoků na počítačové sítě. Následuje vysvětlení k VPN, přehled běžných síťových analyzátorů a nakonec najdeme sekci o firewallu.

Klíčová slova: Mapping, network sniffing, SQL injection, spoofing, DoS, DDoS, Man-in-the-Middle, sociální inženýrství, VPN, remote access, site-to-site, tunel, šifrování, IPSec, GRE, L2TP, OpenVPN, SSH, IP-in-IP, síťový analyzátor, firewall, demilitarizovaná zóna (DMZ), paketový filtr, ACL na síťové vrstvě, SPI, IDS, IPS, DPI, proxy

Cíle studia: Po prostudování této kapitoly získáte základní přehled o bezpečnosti v oblasti počítačových sítí. Budete se orientovat v nejběžnějších typech útoků na sítě, budete vědět, co je to VPN, charakterizovat některé typy VPN, poznáte nejpoužívanější síťové analyzátory a budete umět popsat princip činnosti firewallu a jeho typy.

9.1 Bezpečnost v sítích

9.1.1 Typy útoků

V počítačové síti je třeba čelit různým typům útoků, z nichž některé jsou vzájemně provázány. Především nás zajímají tyto útoky:

Mapping (mapování) – jde vlastně o jakousi přípravu k následujícím útokům. Hacker získává co nejvíc informací (otevřené porty, použité rozsahy IP adres, operační systémy a jejich verze, atd.), aby vlastní útok na konkrétní zařízení mohl být co nejúspěšnější.

Network sniffing (naslouchání na síti) – packet sniffer (zachytávač nebo odposlouchávač paketů) odkloní provoz na síti, zachytává pakety a získává z nich informace (pak je posílá dál k cíli, nedoručené pakety by znamenaly jeho odhalení).

Účelem sniffingu je především získávat hesla přenášená v textovém tvaru a také další citlivé informace. Také je možné zachytávat a pozměňovat obsah paketů anebo dokonce číst a pozměňovat informace na uzlech v síti (včetně konfiguračních souborů nebo hesel).

Účinnou obranou je především spolehlivé šifrování (včetně autentizace po síti), a také fyzické zabezpečení sítě, protože tento typ útoku vyžaduje fyzický přístup k síti. To může být problém u bezdrátových typů připojení.

Packet sniffer (třeba Wireshark) však může být zcela legálně využíván administrátorem ke sledování provozu na síti.

SQL Injection – hackerovi se podaří podstrčit (inject) SQL příkaz někam, kde nemá co dělat, případně vhodnou volbou řetězců „vyrobit“ a poslat serveru SQL příkaz. Většinou se jedná o zneužití formulářů na webu nebo řetězce předávaného přes HTTP GET, kdy útočník vyplní místo běžného řetězce „škodlivý“ řetězec obsahující symboly se speciálním významem. Typický zneužívaný symbol je třeba apostrof.

Proti tomuto typu útoku se celkem dá bránit, pokud programátor webové aplikace (tedy webového rozhraní k databázi) zajistí pečlivou analýzu vstupů od uživatele, a především je třeba mít správně nastavená oprávnění u databáze (například požadavky na destruktivní operace typu DROP, DELETE, ale také změny dat nebo vyžádání si chráněných informací nemají být prováděny běžným uživatelem z nechráněné části sítě, potažmo Internetu).

IP Spoofing – podvrhnutí IP adresy. Komunikující uzel předstírá, že je vlastníkem IP adresy, která ve skutečnosti patří jinému uzlu (nebo nepatří žádnému uzlu). Obvykle jde o případ, kdy se útočník snaží předstírat, že je řádným členem privátní sítě používající určitý rozsah IP adres, případně se pokouší vydávat za konkrétní uzel v síti.

Častým způsobem použití je zneužití některých protokolů, včetně protokolu SMTP pro odesílání e-mailů. Používá se také jako prostředek pro sofistikovanější útoky, například DoS (v paketech, které jsou zasílány na napadené zařízení, je zdrojová adresa podvržená).

Další typy spoofingu – podvržení identity se dá „spáchat“ i jinak, často se jedná o hacknutí některých tabulek s adresami, například:

- ARP Spoofing znamená podvržení záznamů v ARP tabulkách na zařízeních (je napaden mechanismus překladu IP adresy na MAC adresu),
- DNS Spoofing je podvržení záznamů v zónovém souboru na DNS serveru (je napaden mechanismus překladu jmenné adresy na IP adresu).

DoS (Denial of Service) – odmítnutí služby. Jde o vynucení odmítnutí služby legitimním uživatelům. Tento útok probíhá buď využitím chyby v kódu (některého protokolu nebo operačního systému), vyvolaným přetížením sítě nebo podvrženými zprávami – pakety o stavu sítě. Server je zahlcen žádostmi o spojení (TCP handshake) nebo žádostmi o data, které není schopen vyřídit, a proto i následný provoz je zadržen.

DDoS (Distributed DoS) – velmi nebezpečná varianta předchozího typu útoku, proti které se téměř nelze bránit. Častým nedobrovolným „účasníkem“ DDoS útoků jsou sítě botů. *Bot* je napadený počítač ovládaný na dálku hackerem (bez dovolení a většinou také bez vědomí právoplatného majitele). Sítě botů, a to i velmi rozsáhlé (resp. jejich služby), jsou „prodávány“ na černém trhu kromě jiného právě k DDoS útokům.

DDoS útok se dá provést i pomocí jinak celkem užitečných mechanismů. Například typ útoku *Ping Flood* (čte se [flad], znamená „záplava“) probíhá tak, že napadený stroj je zahlcen zprávami ICMP Echo Request (to jsou ty, které posílá příkaz `ping`). Tím se zahlcuje jak jeho vstupní kapacita

(příjem), tak i jeho výstupní kapacita (odesílání) – když se pokouší odpovídat. Aby pachatel skryl svou identitu, v IP paketech zapouzdřujících tyto zprávy falšuje zdrojovou IP adresu. DDoS varianta útoku znamená, že ICMP zprávy posílají zařízení zapojená do botnetu.

Smurf Attack („šmoulí“ útok) má ještě vyšší míru distribuovanosti. Spočívá v tom, že velké množství zařízení (typicky serverů a routerů) dostává zprávy ICMP Echo Request, jejichž zdrojová adresa je podvržená – nastavená na adresu oběti. Tato zařízení odpovídají zprávami ICMP Echo Reply na adresu oběti, čímž zahlcují její vstupní kapacitu.

Man-in-the-Middle – hacker se dostane mezi dva komunikující počítače a odposlouchává nebo dokonce pozměňuje komunikaci mezi nimi. Tento typ útoku souvisí s některými předchozími – únos spojení, zjišťování hesla apod.

Hijacking (únos spojení) – jde především o útoky související s vytáčeným spojením, ale také obecně s jakýmkoliv placeným spojením vyžadujícím autentizaci (například soukromé Wi-Fi sítě). Účinnou obranou je použití vhodného šifrovacího algoritmu při autentizaci.

Útoky na zjištění hesla – heslo lze zjistit například brute-force útokem (hrubá síla), použitím trojského koně a některými výše popsanými metodami. Další možností je sociální inženýrství, které je v současné době na vzestupu (uživatel defacto tento údaj prozradí sám a dobrovolně, je z něho podvodně vylákán).

Brute-force útok může znamenat zkoušení všech možných kombinací znaků v řetězci o „vhodné“ délce, ale může být veden jako *slovníkový*, tj. automaticky jsou zkoušeny všechny řetězce z vytvořeného slovníku (nemusí jít jen o anglická slova!). Proti tomu se lze bránit nastavením maximálního počtu neúspěšných přihlášení, po překročení tohoto limitu je přístup zcela zablokován. Dále je možné po uživatelích vyžadovat používání tzv. „silného“ hesla (dostatečně dlouhého, obsahujícího jak písmena, tak i číslice a další znaky – dvojtečka, procento, zavináč, apod.), s čímž ale bývají problémy (uživatelé raději volí takové heslo, které si dokážou zapamatovat).

Lidský faktor – „nepřítel“ může být i uvnitř sítě, a to dokonce i takový, který to o sobě netuší. Zvláště v poslední době jsou pro útoky často voleny *metody sociálního inženýrství*. Sociální inženýrství je metoda, jak z uživatele vytáhnout potřebné informace třeba i bez použití techniky, a to jeho uvedením v omyl (obelstěním). Uživatel je přesvědčen, že informace předává důvěryhodné osobě z naprosto nutných důvodů.

Útočník se vydává například za zaměstnance bezpečnostního oddělení, opraváře, zaměstnance telefonní společnosti, nového kolegu, apod. a nenápadně ze svých obětí vytáhne vše potřebné (hlavně hesla), případně si „vyzkouší“ nový skvělý počítač nebo se jinak dostane k technice na pracovišti. Stává se to především ve větších firmách, kde se nepočítá s tím, že by se všichni zaměstnanci znali, ale kontakt může probíhat i „neosobně“ přes telefon nebo mail. Právě do telefonu jsou zaměstnanci schopni říci o své firmě neuvěřitelné věci včetně těch, které jsou obchodním tajemstvím.

Popřípadě mnoho uživatelů nepochopitelně důvěřuje e-mailu: stačí například poslat e-mail s informací, že zřejmě došlo ke kompromitaci účtu, přičemž pro kontrolu je třeba zpět odeslat přihlašovací údaje, aby administrátor ověřil, zda je vše v pořádku (navíc e-mail musí být „správně“ graficky vyveden), a spousta uživatelů slepě poslechne.

Sociální inženýrství může mít i „materiální“ povahu – například volně „pohozená“ přenosná

zařízení, která jsou pro mnoho lidí neodolatelná. Podle DHS[1] je kolem 60 % běžných uživatelů schopno náhodně nalezený USB flash disk připojit ke svému počítači, třebaže netuší, zda se na něm nenachází škodlivý software. Podobně lze zneužít i jiná přenosná zařízení, zde je nejlepší ochranou poctivost, tedy odevzdat nalezený předmět do ztrát a nálezů.

Sociální inženýrství je v současné době jedna z nejpoužívanějších (a taky nejefektivnějších) metod získávání utajených informací. Na to by měli myslet zejména administrátoři firemních sítí a zajistit patřičné školení všech, kdo se do firemní sítě připojují.

Jak se bránit?

- Nevěřit všemu, co kdo povídá. Zvláště když jsem například administrátor firemní sítě, musím si vše ověřovat (totožnost žadatele o nové heslo po jeho zapomenutí, nutnost provedení požadovaného zásahu do systému, apod.).
- Heslo či jiné podobné údaje si nenecháváme na papírku přilepeném k monitoru (nebo na jiných „obvyklých“ místech). Volíme silná hesla. Je vhodné nepoužívat totéž heslo u více systémů: pokud útočník prolomí heslo na jednom systému, obvykle se totéž heslo pokouší použít i jinde, kde zjistí účet oběti.
- Každý systém zabezpečený heslem (včetně operačních systémů) lze nastavit tak, aby po stanoveném počtu chybných pokusů o přihlášení byl účet zablokován.
- Banky ani jiné instituce nechtějí po svých zaměstnancích zasílání hesel, certifikátů, TAN, čísla kreditní karty a podobných údajů mailem ani žádným jiným nezapezpečeným způsobem (jen přes zabezpečené stránky přímo při přihlašování). A rozhodně o ně nežádají mailem ani telefonem.
- Správce sítě by měl mít každou žádost o zásah do účtů zaměstnanců nebo systému vždy „papírově“ podloženou.
- Útočníci používající sociální inženýrství dělají „domácí úkoly“ – shánějí co nejvíc informací (včetně osobních), které by mohli využít. Firma by měla zvážit, co zveřejní (a totéž platí i o domácích uživatelích, také například na diskusních fórech a sociálních sítích). Skartovačka není zbytečné zařízení.

9.1.2 Zabezpečení na jednotlivých vrstvách ISO/OSI

Na kterékoliv vrstvě lze šifrovat data. V ISO/OSI modelu se při toku dat směrem dolů šifrují veškerá data vyšší vrstvy, tj. včetně záhlaví. Dále:

1. Na linkové vrstvě (L2) se provádí šifrování podle dané přenosové techniky (např. dle IEEE 802.11).
2. Na síťové vrstvě se implementuje zabezpečení spojení mezi dvěma uzly včetně vytvoření tunelu při VPN spojení pomocí protokolu IPSec nebo stejné funkcionality vestavěné v IPv6.
3. Na aplikační vrstvě modelu TCP/IP pracuje SSL/TLS (o zařazení v referenčním modelu ISO/OSI se vedou diskuse, svou funkcionalitou totiž zřejmě zasahuje na prezentační i relační vrstvu).
4. Na aplikační vrstvě se používají bezpečnostní řešení vycházející z PGP (GPG, OpenPGP apod.).

[1] DHS (Department of Homeland Security, http://www.dhs.gov)

✎ *Protokol IPSec* je dodatečným mechanismem zabezpečení komunikace IPv4 na vrstvě L3, v protokolu IPv6 je již přímo zahrnut. Používá se k zabezpečení spojení mezi dvěma sítěmi (tzv. typ *site-to-site*), typicky když zajišťujeme spojení mezi dvěma pobočkami firmy přes WAN síť.

✎ *SSL* (Secure Sockets Layer) a jeho následník *TLS* (Transport Layer Security) se typicky používají pro připojení mobilního či doma pracujícího zaměstnance, typ *remote access*. Pozor, protokol SSL do verze 2.0 včetně není považován za bezpečný, takže rozhodně používejte alespoň verzi 3.0!

TLS je na rozdíl od SSL standardizován (TLS verze 1.2 je popsán v RFC 5246), oba protokoly jsou sice funkčně velmi podobné, ale vzájemně nekompatibilní (obě strany musí používat buď SSL nebo TLS, nelze kombinovat).

Zatímco IPSec je síťové řešení (transparentní pro různé aplikace), SSL/TLS je aplikační řešení (tj. musí být podporováno konkrétní aplikací, jejíž komunikace má být šifrována). Ve webových prohlížečích je podpora SSL/TLS již dávno zabudována, takže s funkčností tohoto řešení nebývají problémy alespoň v případech, kdy se komunikuje přes protokol HTTP nebo podobné. Pokud aplikace nepodporuje SSL/TLS, je možné to řešit například pomocí *STunnel*.[2]

SSL sice podporuje oboustrannou autentizaci, ale často se setkáváme s řešeními využívajícími jednostrannou autentizaci (autentizuje se dotyčný externí zaměstnanec). Rovněž integrita dat a šifrování jsou tímto protokolem zajištěny, šifrují se pouze přenášená data. Používá se kombinace veřejného a soukromého klíče, je možné použít certifikáty.

SSL je považováno za sice méně bezpečné, ale zato pružnější řešení (například s mechanismem NAT má IPSec problémy, kdežto SSL si s ním poradí celkem bez problémů).

9.2 VPN

✎ VPN (Virtual Private Network) je zabezpečené spojení procházející nedůvěryhodným prostředím (obvykle Internetem). Jedná se o vytvoření tunelu, komunikačního kanálu mezi dvěma body (to mohou být konkrétní koncová zařízení nebo třeba routery, pak propojujeme sítě) obalením původního paketu či rámce do PDU se záhlavím, které bude srozumitelné síťovým zařízením mimo naše sítě.

Existence tunelu neznamená automaticky šifrování, záleží na konkrétním protokolu. U některých „tunelovacích“ protokolů je tedy třeba použít navíc i šifrovací protokol (třeba SSL nebo TLS).

✎ VPN se používá v těchto případech (tj. dělení podle typu ukončujících bodů):

- *Remote Access*: mobilní zaměstnanec potřebuje na svých cestách zabezpečený přístup do firemní (lokální) sítě, aby mohl přistupovat do firemního informačního systému a dalších zabezpečených zdrojů, nebo se jedná o zaměstnance pracujícího doma (Home Office).
- *Site-to-Site* (nebo také *Network-based*): je třeba propojit vzdálené lokální sítě (například pobočky téže firmy).
- Je třeba komunikovat s obchodním partnerem zabezpečeným komunikačním kanálem, ale zároveň ho nechceme pustit přímo do naší lokální sítě. Může se jednat o *site-to-site* nebo *remote access*, podle skutečné konfigurace VPN, tato možnost je vlastně speciálním případem obou předchozích.

[2] http://www.stunnel.org/

Ve všech těchto případech je třeba vybudovat zabezpečený šifrovaný *tunel*, přes který povede komunikace nedůvěryhodným prostředím – zajišťujeme předně vhodné zapouzdření s případným překladem adres (protože pakety půjdou přes cizí síť s jinými adresními rozsahy a případně jinými protokoly), a dále zajišťujeme šifrování. Ve třetím případě navíc musíme použít firewall, který bude propouštět pouze komunikaci povolenou pro tento účel.

VPN tunel do firemní sítě ústí většinou buď přímo na routeru s firewallem na hranici lokální sítě, který je viditelný na Internetu, anebo v demilitarizované zóně (tam se často umísťuje VPN koncentrátor, což je vlastně jakási VPN brána), záleží, kam v lokální síti je třeba přes tunel přistupovat.

VPN řešení by mělo zajistit následující:

- autentizace – je třeba ověřit totožnost obou komunikujících bodů (např. uživatel s notebookem na pracovní cestě a firewall s podporou VPN v LAN firmy), případně se zajišťuje autentizace přenášených PDU,
- autorizace – stanovení konkrétních přístupových oprávnění,
- zajištění důvěrnosti dat – přenos je vždy šifrován,
- zajištění integrity dat – detekce pozměnění či poškození paketu po cestě.

Existuje řada protokolů, které dokážou vytvořit tunel. Liší se různými parametry, například typem tunelu (site-to-site nebo remote access), možností šifrování a dalšími. Podíváme se na pár nejběžnějších řešení (podrobněji na navazujícím magisterském studiu):

IPSec (Internet Protocol Security) zajišťuje jak vytvoření tunelu, tak i šifrování. Kombinuje se s IPv4, v IPv6 je již zahrnut (jako jedno z volitelných záhlaví).

Pracuje na síťové vrstvě, díky tomu je transparentní pro aplikační protokoly. Většinou zapouzdřuje IP pakety. Používá se jak pro řešení site-to-site, tak i pro remote access.

Nevýhodou IPsec jsou problémy s NAT, ale taky to, že nedokáže přenášet multicast vysílání.

Další informace:

- Podrobnosti především o Linuxu najdeme v odkazech na konci sekce o VPN, především v odkazu http://www.ipsec-howto.org/ipsec-howto.pdf
- Konkrétní postup konfigurace na Linuxu je v https://www.root.cz/clanky/tuneluji-tunelujes-tunelujeme-ipsec/, je to jeden z dílů seriálu o tunelování.

GRE (Generic Routing Encapsulation) pracuje na síťové vrstvě a používá se typicky pro site-to-site tunely.

Obvyklý postup je takový, že původní paket je opatřen GRE záhlavím (velice jednoduchým, pár polí) a následně je přidáno záhlaví nosného protokolu, obvykle IP (v něm je v poli Protocol/NextHeader číslo 47 určující protokol GRE). Všimněte si, že v postupu není ani slovo o šifrování, to totiž neumí.

Jeho výhodou je univerzálnost, dokáže zapouzdřit i jiné typy PDU síťové vrstvy ISO/OSI než jen IP. Dokonce může zapouzdřit i protokoly jiných vrstev, včetně rámců z vrstvy L2. Dokáže přes tunel transportovat i multicast a broadcast vysílání. Další výhodou je, že tento protokol je podporován prakticky všude (v Linuxu, ve Windows, na síťových zařízeních různých výrobců).

GRE dokáže přenášet přes tunel i multicast vysílání, což je důležité například při distribuování směrovacích informací mezi routery.

Na druhou stranu, velkou nevýhodou GRE je, že neprovádí šifrování. V praxi je GRE kombinovanán s IPSec, TLS nebo jiným řešením, které umí šifrovat.

Další informace:

- https://www.ietf.org/rfc/rfc2784.txt
- https://www.root.cz/clanky/tuneluji-tunelujes-tunelujeme-jak-a-k-cemu/

L2TP je potomkem starších protokolů PPTP (Microsoft) a L2F (Cisco). Jak název napovídá, tento protokol pracuje na vrstvě L2 (Layer 2 Tunneling Protocol). VPN protokoly linkové vrstvy jsou zajímavé tím, že obvykle dělají „smyčku“ na vyšší vrstvu – zapouzdřují provoz z nadřízené vrstvy, ale samy se zapouzdřují do TCP nebo UDP segmentu.

Hlavním účelem protokolu *L2TP* je tunelování (zapouzdřování) paketů protokolu PPP (ten se používá pro přenos dat přes telefonní síť, včetně ADSL/VDSL), je velmi oblíbený u různých ISP. Zapouzdřuje se do UDP segmentů, nepoužívá TCP (tunel tedy nemá oporu na vrstvě L4). Stejně jako GRE, ani L2TP neprovádí šifrování, tedy je obvykle kombinován s IPSec

L2TP je podporován v Linuxu i ve Windows. V Linuxu se dnes doporučuje použití OpenL2TP (s využitím IPSec), ve Windows se nachází klient L2TP/IPSec.

Poznámka:

Ve Windows se také můžeme setkat s protokolem SSTP (Secure Socket Tunneling Protocol), který přenáší PPP nebo L2TP šifrované s využitím protokolu SSL.

OpenVPN je projekt, který běží na většině známých UNIXových systémů včetně Linuxu, existuje i varianta pro Windows. Je to otevřené řešení pro remote access, které zvládá jak vytvoření tunelu, tak i šifrování, šifrují se pouze přenášená data.

Toto řešení pracuje na vyšších vrstvách (nad L3), což znamená, že například není třeba řešit problémy s NAT či jinými překlady IP adres. Pro šifrování, autentizaci a správu klíčů se používá protokol SSL nebo TLS, na transportní vrstvě se používá většinou UDP, ale je možné použít i TCP. Protokol TLS je sice standardizován, ale celé řešení OpenVPN nikoliv.

SSH je protokolem vyšších vrstev a komunikuje přes TCP. Na rozdíl od TLS a některých dalších podobných protokolů SSH-2 nabízí správu více navázaných relací najednou (TLS pouze jedno spojení). Jedná se o tunely typu remote access, point-to-point, na vyšších vrstvách.

Existuje více různých implementací SSH. K nejoblíbenějším patří nástroj *OpenSSH*, pro který existuje klientská i serverová varianta (spíše pro UNIXové systémy). Pro Windows existují mezi volně šiřitelným softwarem spíše klienty, například *PuTTY*.

IP-in-IP je jednoduše zapouzdření IP paketu do IP paketu. Může jít o stejné verze nebo různé verze, často se například tímto způsobem řeší transport IPv6 paketů přes úsek sítě, který „nerozumí“ IPv6 (takže IPv6 paket zapouzdříme do IPv4 paketu).

Důležitým prvkem Ip-in-IP je změna IP adres v záhlaví, což je vlastně jedna z vlastností tunelování: ve vnějším záhlaví se jako zdrojová použije IP adresa zařízení (routeru) na začátku

tunelu, jako cílová se dosadí adresa konce tunelu. Na rozdíl od NAT (což je vlastně také překlad adres) se původní záhlaví zachovává (NAT zasahuje do původního záhlaví, nové nevytváří).

V případě Ipv6 je přímo v tomto mechanismu vestavěna možnost šifrování (ve vnějším IP záhlaví by pak bylo volitelné bezpečnostní záhlaví), u IPv4 bychom šifrování museli zajistit jinak (třeba pomocí protokolu IPSec).

MPLS VPN. MPLS sítě se používají spíše pro implementaci propojení firemních poboček tunelem (příp. firmy a obchodního partnera), tedy VPN typu site-to-site, spoje jsou typu point-to-point. S MPLS sítěmi jsme se již seznámili a víme, že se používá systém záhlaví, z nichž jedno je určeno právě pro virtuální sítě. Řešení pracuje na rozhraní vrstev L2 a L3.

Pro MPLS VPN existují dvě základní řešení – MPLS Layer-3 VPN, MPLS Layer-2 VPN. Zapouzdřovat se mohou jak IP pakety, tak i ethernetové rámce (řešení Ethernet over MPLS – EoMPLS), případně PDU jiných protokolů na těchto vrstvách.

Nevýhodou MPLS VPN je, že potřebujeme externího poskytovatele této služby (obvykle ISP, přes jehož WAN síť náš tunel povede), tyto tunely si nemůžeme vytvořit sami. Naopak výhodou je, že pakety přenášené tunelem neprocházejí přes nezabezpečený Internet, konfigurace je na straně poskytovatele služby a součástí služby je i QoS, takže i přes poměrně vysoké finanční náklady si tato služba své zákazníky nachází.

9.3 Síťový analyzátor

Síťový analyzátor je zařízení, které se připojuje mezi některý prvek sítě (PC, server, most, router, atd.) a LAN, také může být součástí jiného zařízení. Samostatný síťový analyzátor (přenosný) může být i velmi malý, může připomínat mobilní telefon. Už jsme se setkali s analyzátorem pro Ethernet.

Pracuje na fyzické nebo vyšších vrstvách, na tom záleží, které charakteristiky dokáže sledovat. Na fyzické vrstvě především stav média, provoz (zatížení), dokáže generovat vlastní provoz, na vyšších vrstvách může podporovat také virtuální sítě, sledování stavu dostupných uzlů sítě, sledování rámců, příp. paketů, atd.

Hardware of firmy Fluke je v této oblasti všeobecně znám. Nejpoužívanější přístroje:

- *LinkRunner* – pracuje na fyzické vrstvě, identifikace vlastností Ethernetového spojení – negociace (10/100/1000 Mb, duplex apod.), detekce poruch kabelů včetně vzdálenosti k poruše, zásuvek, provoz na segmentu, atd.,
- *NetTool* – síťová vrstva, testy na fyzické vrstvě (kabeláž, negociace, využívání segmentu), linkové (detekce kolizí a chybových rámců, VLAN, vyhledávání MAC adres v síti, apod.), síťové (vyhledávání IP adres v síti, podsítí, routerů a dalších zdrojů, ping, atd.), měření PoE, monitoring parametrů VoIP, atd.,
- *AirCheck* – analýza Wi-fi sítě.

Hardware of firmy Embedded Technologies obsahuje obvykle vlastní variantu embedded Linuxu (tj. Linuxu upraveného pro speciální účely, zde pro síťová zařízení) zároveň s dalším potřebným softwarem. Například:

- *Síťový analyzátor* – vestavěný sniffer Wireshark, NAT, firewall, atd.,
- *Diagnostický switch* pro Ethernet.

Něco podobného si lze vytvořit vlastnoručně ze staršího (nadbytečného) počítače s potřebným množstvím hardwarových rozhraní, distribuce embedded Linuxu jsou dostupné na Internetu.

✎ **Softwarové síťové analyzátory:** Síťové analyzátory mohou být také softwarové – sledují a případně ovlivňují síťový provoz související s počítačem, na kterém jsou nainstalovány (ideálně server, ale může to být kterýkoliv počítač v síti). Asi nejznámější softwarový síťový analyzátor je *Wireshark*.

9.4 Firewall

9.4.1 Princip firewallu

✎ Firewall je síťový prvek (hardwarový nebo softwarový), který slouží k řízení provozu mezi sítěmi s různou úrovní důvěryhodnosti či zabezpečení. Ve firewallu jsou definována *pravidla* pro komunikaci mezi sítěmi, podle kterých reaguje buď povolením komunikace, jejím zakázáním a nebo žádostí o vyjádření uživatele. Pravidla se obvykle týkají těchto údajů:

- zdroj a cíl komunikace, může být zadán IP adresou,
- číslo portu, přes který se komunikuje (tj. můžeme zablokovat používání některého portu), může jít o zdrojový i cílový port,
- používaný protokol,
- stav spojení, atd.

✎ Firewall může být buď jen *jednosměrný* (filtruje pouze příchozí pakety) nebo *obousměrný* (filtruje příchozí i odchozí pakety). Lepší je samozřejmě obousměrný, dokáže efektivněji zachytit případné vynášení citlivých informací.

✎ Kromě softwarových firewallů existují také *hardwarové firewally* fungující jako mezilehlé prvky sítě. Dnes jsou většinou součástí routerů nebo jiných běžných síťových prvků (na typu zařízení závisí, k jakým informacím se firewall dostane). Hardwarový firewall má velkou výhodu v nižším riziku napadení (není závislý na operačním systému klientského počítače). Další výhodou je nezávislost na výkonu počítače – pokud je procesor počítače hodně zatížen, má to vliv i na činnost (především rychlost) softwarového firewallu na tomto počítači nainstalovaného. Hardwarový firewall (třeba vestavěný v jiném síťovém zařízení) takto ovlivněn není.

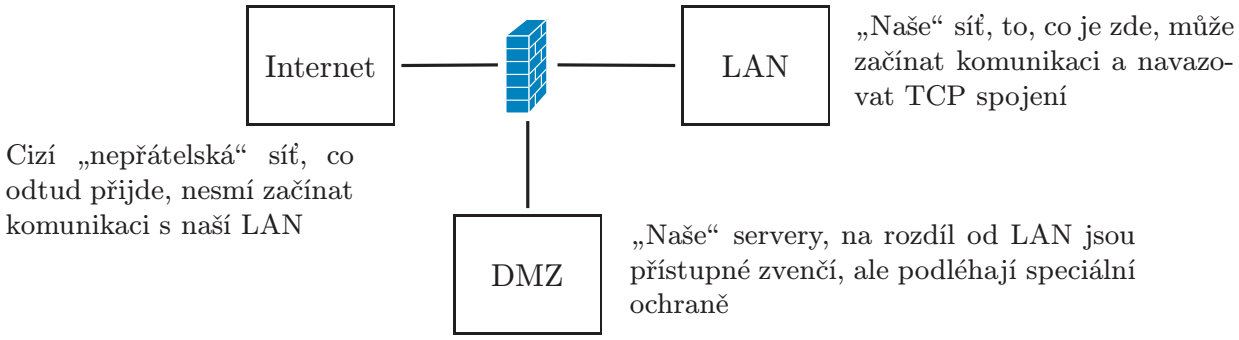
Z hlediska bezpečnosti je v domácnostech a menších firmách za ideální považována kombinace jednoduchého hardwarového firewallu ve směrovači (například ADSL router) a softwarového routeru na počítači, každý z nich jiného typu.

Softwarové firewally slouží buď k ochraně koncových uzlů, a nebo mohou běžet v operačním systému nainstalovaném na (téměř) jakémkoliv síťovém prvku.

✎ **Oblasti podle firewallu.** Z pohledu firewallu členíme síť do několika oblastí:

- *vnitřní síť* – do této oblasti patří vše, co je „naše“, čemu můžeme důvěřovat, uzel z vnitřní sítě může (s ohledem na přístupová oprávnění) iniciovat spojení k vnitřní i vnější síti,
- *vnější síť* – typicky Internet,
- *demilitarizovaná zóna* (DMZ) – zóna „nikoho“, z bezpečnostního hlediska někde mezi vnitřní a vnější sítí.

Do DMZ lze umístit například to, co je sice „naše“, ale má být přístupné z vnější sítě (například web server, mail server, DNS server). Servery v DMZ by měly být zabezpečeny tak, jako by byly opravdu přímo na Internetu, třebaže jsou od Internetu odděleny firewallem.



Obrázek 9.1: Oblasti z pohledu firewallu

Další možnost využití DMZ je propojení k třetí straně, které víceméně důvěřujeme, typicky dodavateli služby, kterou si nemůžeme zajistit vlastními silami. Obě možnosti můžeme zkombinovat a používat dvě demilitarizované zóny (nebo více).

Na síťovém zařízení podporujícím DMZ máme obvykle jeden nebo více (hardwarových!) portů takto označených, případně můžeme některé porty sami nakonfigurovat tak, aby se s nimi zacházelo jako s DMZ (například tehdy, když chceme starší počítač využít jako hardwarový firewall, ukázky najdeme v příloze).

✎ **TCP/UDP porty.** Běžný uživatel si většinou s nastavením (softwarových) portů neví rady. Na internetu najdeme stránky se seznamy známých a registrovaných portů používaných různými protokoly.[3] Tyto seznamy jsou však použitelné pro sledování odchozího provozu nebo při nastavování portů na serveru. Ve skutečnosti aplikace mohou používat i jiné porty, než které jsou běžné pro protokoly, se kterými pracují.

9.4.2 Typy filtrování

Rozlišujeme různé typy filtrování podle toho, na kterou vrstvu ISO/OSI lze danou metodu zařadit (a tedy na které informace v záhlavích „dosáhneme“). Obvykle se jedná především o filtrování na síťové vrstvě (L3), protože tam lze pracovat s IP adresami.

✎ **Paketový filtr.** Jedná se o nejjednodušší filtrování na L3 a částečně L4, v pravidlech se uvádějí jen IP adresy a čísla portů. Je to jednoduché a rychlé řešení (provoz je zdržován jen minimálně), které se dříve běžně uplatňovalo především na mezilehlých síťových prvcích (například starší verze operačního systému IOS pro směrovače), dnes je najdeme v některých nejlevnějších směrovačích. Nevýhodou je neschopnost nahlížet do komunikace probíhající v složitějších protokolech.

✎ **ACL na síťové vrstvě.** ACL (Access Control List – seznam řízení přístupu, přístupový seznam) je metoda široce používaná jak v desktopových a serverových operačních systémech,

[3] Seznam portů a případně služeb najdeme například na http://www.iana.org/assignments/port-numbers, http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers, http://www.ports-services.com/. Na české Wikipedii není seznam úplný.

tak i v síťových zařízeních. Účelem je určovat pravidla přístupu pro různé uživatele/komunikace. Vpodstatě se jedná o funkční nástavbu metody paketového filtru.

Přístupový seznam je vlastně seznam položek označených kategorií:

- *deny* (nepustit) – to, co nemá projít,
- *permit* (pustit) – to, co může projít,
- *deny all else* (nepustit nic kromě...) – co nebylo zmíněno, nesmí projít.

ACL je většinou implementován ve formě „co není dovoleno, je zakázáno“, takže najdeme spíše jen položky *permit* (a co není v těchto položkách, to neprojde). Případně se můžeme setkat s celou strukturou navzájem provázaných ACL. Položky se procházejí sekvenčně, jedna po druhé, a hledá se shoda. Také pořadí položek je důležité.

Filtruje se obvykle podle cílové IP adresy, podle zdrojové IP adresy, podle jejich kombinace, a nebo případně podle dalších kritérií (například podle položek v PDU síťové vrstvy – protokolu IP, ICMP, podle TCP/UDP portu, se kterým se navazuje spojení ze síťové vrstvy, apod.). Adresa obvykle bývá adresou podsítě, tedy je uložen prefix a jeho délka (resp. maska, aby bylo zřejmé, u kolika bitů adresy se má hledat shoda).

✎ **Stavová inspekce paketů.** Přesuneme se o vrstvu výše – na transportní vrstvě (L4, konkrétněji teď jsme na L3 a L4) se provádí filtrování *SPI* (Stateful Packet Inspection, stavová inspekce paketů, také stavový paketový filtr). Zatímco na L3 se pakety berou jako „jedináčci“ bez vzájemné vazby, na L4 je možné brát při filtrování v úvahu jejich vzájemné vztahy. například můžeme odlišit pakety, které navazují spojení, od paketů, které patří do již existujícího spojení.

Paket navazující spojení je důkladně prověřen (údaje z vrstev L3 a L4, například zdrojová a cílová adresa, protokoly, zdrojové a cílové porty, příznaky nastavené podle jednotlivých protokolů, cokoliv, co je v záhlavích PDU) a pokud projde, vytvoří se v *stavové tabulce* záznam povolující dané spojení. Pokud paket kontrolou neprojde úspěšně, záznam se nevytvoří.

Pakety, které patří do již navázaného spojení (nebo se za takové vydávají), se filtrují podle toho, zda jejich spojení je zaznamenáno ve stavové tabulce.

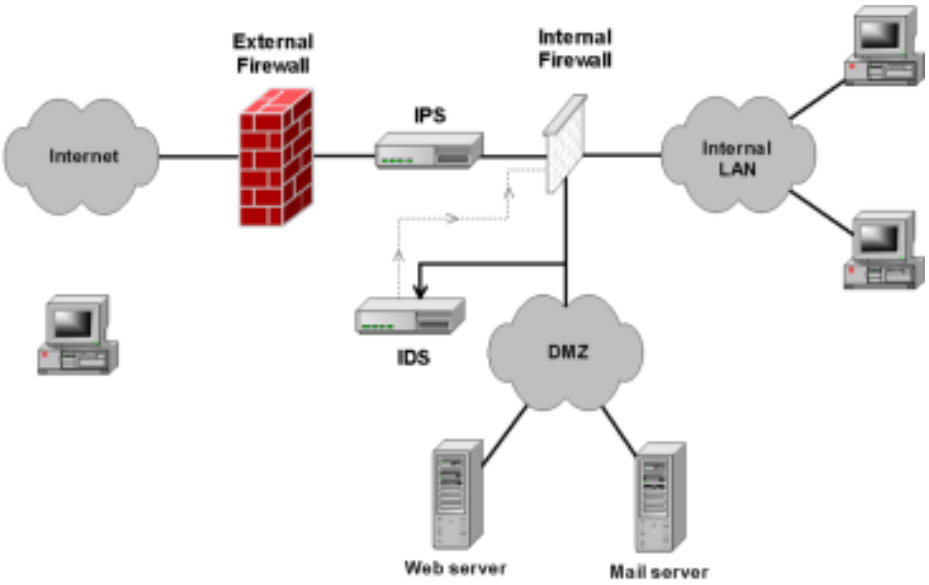
V současné době je SPI vpodstatě standardem kvalitních firewallů. Oproti běžnému paketovému filtru nabízí větší bezpečnost při relativně malém zpomalení provozu při filtrování.

✎ **IDS/IPS, DPI.** SPI můžeme brát jako základ pro IDS/IPS systémy:

1. *IDS* (Intrusion *Detection* System, systém pro detekci útoků) – pracuje podobně jako antivirus, tedy používá
   - signatury útoků pro rozpoznání známých typů útoků (vede si jejich databázi, kterou je nutné aktualizovat nebo „mít k dispozici v cloudu“),
   - heuristiky (využívá statistické metody vyhodnocující provoz na síti) – hledá v provozu na síti podezřelé pakety,
   - detekci neobvyklého chování sítě (zjišťují se odchylky od běžného provozu na síti).

   Detekuje pokusy o průnik do systému a podá informaci zařízení, které dokáže na útok reagovat.
2. *IPS* (Intrusion *Prevention* System) – podobně jako IDS provádí detekci útoků, ale navíc aktivně reaguje. Reakce má útoku zabránit, proto také konkrétní místo IPS sondy je třeba naplánovat tak, aby mohla případně také zasahovat do konfigurace jiných síťových zařízení (například přenastavit pravidla ve firewallu).

Firewall může mít integrovanou funkci (modul) IDS/IPS, ale obecně je bezpečnější (zvláště u větších sítí) nasadit IDS/IPS odděleně od firewallu. Setkáváme se také s názvem *Stavový paketový filtr s hloubkovou kontrolou* (Deep Packet Inspection), to je právě firewall s integrovaným modulem IDS/IPS. Tento typ firewallu přidává další možnosti definování pravidel související s obvyklými vlastnostmi komunikace s danou (známou) aplikací či protokolem. Pokud například zjistí, že protokol HTTP je používán pro jiný typ komunikace než s WWW serverem, tento požadavek zablokuje jako podezřelý.



Obrázek 9.2: Schéma možného zapojení IDS/IPS[4]

Na obrázku 9.2 vidíme možné schéma zapojení firewallů a IDS/IPS ve větší síti. Celou síť chrání externí firewall, za kterým je IPS – přes IPS jde veškerá komunikace, tedy může reagovat na cokoliv podezřelého v kterémkoliv paketu. Následuje vnitřní firewall, který rozděluje síť na oblasti tak, jak bylo výše naznačeno – odděluje vnější „nedůvěryhodnou“ síť (Internet), vnitřní (relativně důvěryhodnou) lokální síť a demilitarizovanou zónu pro servery. DMZ má dodatečnou ochranu ve formě IDS zařízení.

Poznámka:

Všimněte si, že IDS není „přímo na cestě“ k DMZ, ale vede k němu „odbočka“. Provoz směřující do DMZ (nebo opačně) je zrcadlen (kopírován) na port vedoucí k IDS zařízení. Proč tomu tak je? Protože nechceme, aby zařízení IDS „bylo vidět“. Pokud se útočníkovi podaří vloudit se do sítě a začne skenovat naši síť (zjišťovat, kde co máme, jakou co má adresu apod.), IDS pro něj zůstane neviditelným ze dvou důvodů – pouze detekuje (nedává o sobě vědět žádnými aktivními reakcemi) a zároveň není na cestě k žádnému síťovému prvku. Takže máme v síti skrytý prostředek, který můžeme následně použít proti útočníkovi, resp. sledovat jeho činnost bez toho, aby to útočník zpozoroval.

Proč máme IDS právě u DMZ? Protože DMZ je zranitelnější než vlastní lokální síť. Zatímco směrem do LAN nesmí být z Internetu navazována žádná spojení (cokoliv takového na firewallu

[4] Zdroj: http://www.actinet.cz

hned „zařízneme“, u DMZ to udělat nesmíme, protože potřebujeme, aby na naše servery přistupovali klienti z vnějšku. Takže ochrana DMZ je složitější a každý ochranný prvek navíc se hodí.

Další informace:

- http://www.actinet.cz/bezpecnost_informacnich_technologii/l19/cl25/st1/j1/Uvod_do_IDS/IPS.html
- http://www.systemonline.cz/clanky/systemy-pro-detekci-neopravneneho-pruniku.htm

Proxy na aplikační vrstvě. Posuneme se ještě o vrstvu výše – na aplikační vrstvě TCP/IP pracují proxy firewally (také aplikační brány). Pracují s aplikačními protokoly, například rozumí protokolu HTTP, FTP, IMAP, dokážou pracovat s pakety obsahujícími prvky pro ActiveX, apod.

Tento typ firewallu odděluje sítě až do té míry, že počítač (server) ve vnější síti nezná IP adresu počítače ve vnitřní síti, se kterým komunikuje. Veškerá komunikace je zpracovávána pomocí tzv. *proxy* – softwarových bran buď naprogramovaných pro konkrétní typ komunikace (protokol) s poměrně vysokým stupněm zabezpečení (například pro protokol FTP nebo HTTP), nebo pomocí generické proxy použitelné obecně pro různé protokoly.

Proxy defacto zcela odděluje vnitřní a vnější síť (v podobném smyslu jako NAT) a při filtrování využívá informace prakticky ze všech vrstev TCP/IP, protože při „prokopávání“ k záhlaví protokolů vrstvy L7 prochází přes záhlaví předchozích vrstev.

Rozlišujeme dva typy proxy:

1. *Běžný (standardní) proxy* – pro něj platí vše, co bylo dříve o proxy napsáno. Filtruje všechny pakety podle údajů z PDU protokolů aplikační vrstvy a nižších vrstev.
2. *Dynamický proxy* – chová se odlišně k různým paketům; k zahajujícím spojení se chová stejně jako první typ proxy, ale k paketům patřícím do již vytvořeného spojení se chová jako SPI (tj. na nižší vrstvě TCP/IP). Důsledkem je zrychlení odbavování příchozího i odchozího provozu.

Další informace:

http://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=15131

Seznam doporučené literatury

[1] *CESNET2* [online]. Dostupné na: http://www.cesnet.cz/ [cit. 20. 5. 2017]

[2] Donahue, Gary A. *Kompletní průvodce síťového experta*. Brno: Computer Press, 2009, 528 s. ISBN 978-802-5122-471.

[3] Doseděl, Tomáš. *Počítačová bezpečnost a ochrana dat*. Brno: Computer Press, 2004, 190 s. ISBN 80-251-0106-1.

[4] Elenkov, Nikolay. *Android Security Internals*. San Francisco, USA: No Starch Press, 2014. ISBN 1-593-227581-1.

[5] Empson, Scott. *CCNA: Kompletní přehled příkazů. Autorizovaný výukový průvodce.* Brno: Computer Press, 2009. ISBN 978-80-251-2286-0.

[6] Gubbi, Jayavardhana, et al. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, Elsevier, 2013, Volume 29, Issue 7. Pages 1645–1660, ISSN 0167-739X.

[7] Hubert, Bert et al. *Linux Advanced routing & Traffic Control* [online]. Dostupné na: http://lartc.org/howto/, další na http://lartc.org/ [cit. 1. 7. 2017]

[8] Hura, Gurdeep S. a Mukesh Singhal. *Data and computer communications: networking and internetworking*. Boca Raton, FL: CRC Press, 2001. ISBN 08-493-0928-X. Dostupné také z: https://books.google.cz/books?id=BViV0PoH_voC&printsec=frontcover

[9] Jirovský, Václav. *Kybernetická kriminalita: nejen o hackingu, crackingu, virech a trojských koních bez tajemství*. Praha: Grada, 2007, 284 s. ISBN 978-80-247-1561-2.

[10] Lammle, Todd. *CCNA: výukový průvodce přípravou na zkoušku 640-802*. Brno: Computer Press, 2010, 928 s. ISBN 978-802-5123-591.

[11] *Linux Home Networking* [online]. Knihy o správě počítačových sítí v Linuxu. Dostupné na: http://www.linuxhomenetworking.com/wiki/index.php [cit. 1. 7. 2011]

[12] Lockhart, Andrew. *Bezpečnost sítí na maximum*. Překlad Jiří Veselský. Brno: CP Books, 2005, 276 s. ISBN 80-251-0805-8.

[13] Peterka, Jiří. *eArchiv: Co je čím v počítačových sítích* [online]. Dostupné na: http://www.earchiv.cz/i_coje.php3 [cit. 20. 5. 2010]

[14] Peterka, Jiří. *eArchiv: Principy počítačových sítí* [online]. Dostupné na: http://www.earchiv.cz/i_pri.php3 [cit. 20. 5. 2010]

[15] Price, Brad. *Active Directory: Optimální postupy a řešení problémů*. Brno: Computer Press, 2005, 381 s. ISBN 80-251-0602-0.

[16] Pužmanová, Rita. *Bezpečnost bezdrátové komunikace: jak zabezpečit wi-fi, bluetooth, GPRS či 3G*. Brno: Computer Press, 2005, 179 s. ISBN 80-251-0791-4.

[17] Pužmanová, Rita. *TCP/IP v kostce*. 2., upr. a rozš. vyd. České Budějovice: Kopp, 2009, 619 s. ISBN 978-80-7232-388-3.

[18] Ruest, Danielle a Nelson Ruest. *Virtualizace: Podrobný průvodce*. Brno: Computer Press, 2010. ISBN 978-802-5126-769.

[19] Sanders, Chris. *Analýza sítí a řešení problémů v programu Wireshark*. Brno: Computer Press, 2012, 288 s. ISBN 978-80-251-3718-5. Dostupné také z: https://www.nostarch.com/packet2.htm

[20] Satrapa, Pavel. *IPv6: internetový protokol verze 6*. Třetí, aktualiz. a dopl. vyd. Praha: CZ.NIC, 2011. 407 s. Kniha je dostupná také v elektronické formě na https://knihy.nic.cz/

[21] Schroder, Carla. *Linux: Kuchařka administrátora sítě*. Brno: Computer Press, 2009, 596 s. Z originálu Linux Networking Cookbook vydaného nakladatelstvím O'Reilly Media. ISBN 978-802-5124-079.

[22] Thomas, Thomas M. *Zabezpečení počítačových sítí. Autorizovaný průvodce*. Brno: CP Books, 2005, 338 s. ISBN 80-251-0417-6.

[23] Trulove, James. *Sítě LAN: hardware, instalace a zapojení*. Praha: Grada, 2009, 384 s. ISBN 978-80-247-2098-2.

[24] Verma, Dinesh C. *Content distribution networks: an engineering approach*. New York, N.Y.: Wiley, c2002, xv, 182 p. ISBN 04-714-4341-7.