Počítačové sítě

J. Fesl, V. Černý, J. Janeček, V. Smotlacha, Y. Trofimova

Obsah

Předmluva

Toto skriptum je určené jako jako podpůrný studijní materiál pro studenty předmětu Počítačové sítě, vyučovaného na Fakultě informačních technologií Českého vysokého učení technického v Praze.

Počítačové sítě jsou perspektivní oblastí, která se aktivně rozvíjí již několik desetiletí. Ačkoli technologie v počítačových sítích se neustále inovují a mění, principy na kterých tyto technologie jsou založené, zůstávají stále stejné. Dobrá znalost principů počítačových sítí je důležitým předpokladem pro rychlé a efektivní řešení problemů v praxi.

Skriptum se skládá z 6ti kapitol týkajících se prvních čtyř vrstev modelu OSI/ISO. Čtenář se postupně seznámí se základními pojmy z oblasti počítačových sítí, pochopí principy technologií používaných pro doručování dat v lokálních i vzdálených sítích, získá detailnější znalosti rodiny protokolů IP, seznámí se s běžně používanými směrovacími algoritmy a protokoly, pochopí jak funguje síť Internet jako celek a bude rozumět tomu, jakým způsobem spolu komunikují síťové aplikace.

Věříme, že toto skritpum ocení nejen studenti výše zmíněného předmětu, ale kdokoli kdo má zájem porozumět problematice počítačových sítí.

V Praze, únor 2023 Kolektiv autorů

Kapitola 1

Úvod do počítačových sítí

Pojmem lokální síť zpravidla označujeme komunikační systém schopný propojit desítky až stovky počítačů na vzdálenost stovek metrů až jednotek kilometrů. Lokální sítě jsou využívány v administrativě, v inženýrských systémech a v technologickém řízení. Bez lokálních sítí si současné nasazení kvant osobních počítačů nelze představit. Zatímco rozsáhlé počítačové sítě jsou nejužitečnější v těch aplikacích, kde zajišťují přenosy dat (elektronická pošta, sběr dat), typickou aplikací pro lokální sítě je zajištění přístupu k systémovým prostředkům které jsou spravovány jen některými počítači sítě (označujeme je obvykle jako servery) a využívány počítači ostatními (označujeme je obvykle jako uživatelská nebo klientská pracoviště). Takovými systémovými prostředky jsou nejčastější drahá zařízení (rychlé a speciální tiskárny), velké a sdílené soubory a databáze.

Lokální počítačové sítě jsou vhodným prostředkem i tam, kde je třeba rozložit výpočetní kapacitu tak, aby poskytované služby byly snáze dostupné, aby bylo možné specializovat jednotlivé počítače na konkrétní funkce a abychom zvýšili spolehlivost výpočetního systému. Aplikacemi jsou měřící a sledovací systémy ve vědě a zdravotnictví, řízení technologických procesů v průmyslu a automatizace administrativy. Problematika lokálních sítí zahrnuje řadu oblastí. Patří sem vytvoření vlastního fyzického spoje mezi počítači, tedy technologie kabelových propojení a komunikačních (síťových) karet. Obvykle se rozhodujeme mezi několika řešeními která odpovídají zavedeným standardům.

S potřebou rozumět komunikačním protokolům se setká každý, kdo bude nucen implementovat aplikační program nebo službu, která komunikačních schopností lokální sítě využívá nad rámec funkcí souborového serveru. .

A konečně, i pro běžného uživatele počítačů má svůj význam přehled služeb, která mu síť poskytne. V nejjednodušší formě jde o rozšíření operačního systému jeho pracoviště o přístup ke sdíleným souborům a zařízením serverů. Modernější systémy pro lokální sítě podporují rozklad takových aplikací jako je přístup k databázím nebo elektronická pošta formou označovanou jako Klient-Server. Přehledu současných systémů podporujících provoz lokálních sítí a vývoj síťových aplikací je věnována závěrečná část textu.

Lokální sítě za krátkou dobu svého rozvoje prošly řadou proměn. Klasické sdílení jediného přenosového kanálu je u lokálních sítí opírajících se o kabeláž (elektrickou nebo optickou) stále více nahrazováno přepojováním. Jako přenosové médium jsou stále častěji využívána optická vlákna. S rozvojem přenosných počítačů (a o počítačovou techniku se opírajících přenosných zařízení) roste význam rádiových lokálních sítí. Mění se požadavky

kladené na vlastnosti lokálních sítí; nejen že roste množství vyměňovaných dat mezi zvyšujícím se počtem počítačů, ale zvyšují se i požadavky na kvalitu komunikačních služeb (isochronní provoz, rozumná degradace služeb při přetížení sítě). Klasické technologie jsou přizpůsobovány novým požadavkům tak že z nich často zbývá pouhé rozhraní koncových účastníků; jako příklad může sloužit rozhraní Ethernetu, využívané technologiemi širokopásmových sítí (CATV) nebo přístupovou technologií EFM (Ethernet for the First Mile). Takový přístup, spolu s využíváním formátů Ethernetu i na vysokorychlostních dálkových spojích gigabitového a desetigigabitového Ethernetu, usnadňuje integraci lokálních sítí a digitálních spojů sítí rozsáhlých a globálních. Moderní řešení lokálních sítí dovolují oddělit vlastní komunikační systém od uživatelské struktury sítě, nastupují řešení označovaná jako virtuální lokální sítě.

1.1 Architektura, topologie a model ISO/OSI

1.1.1 Architektura

Lokální sítě se od sítí přepojovacích liší hlavně tím, používají pro propojení stanic vícebodových kanálů. U těchto kanálů hraje, vzhledem k jejich sdílení vzdálenými stanicemi, podstatnou roli zpoždění signálu při průchodu médiem.

Přívlastek lokální vyjadřuje také skutečnost, že síť' pokrývá malé území. Rozměry sítě přitom nejsou omezené našimi potřebami, ale teoretickými vlastnostmi přístupových metod, které lokální sítě používají.

Budeme-li se snažit vyjádřit rozlehlost sítě numericky, můžeme ji definovat jako poměr a mezi zpožděním signálu T a střední dobou potřebnou pro vyslání jednoho paketu to při dané přenosové rychlosti a.

Pro sítě, které označujeme jako rozlehlé, platí a=1. Sítě, které budeme označovat jako lokální (nebo soustředěné) platí $a \neq 1$. Přenosové médium je využito v daném okamžiku pro přenos jediného paketu, v rozlehlých sítích může být média využito pro přenos více paketů současně.

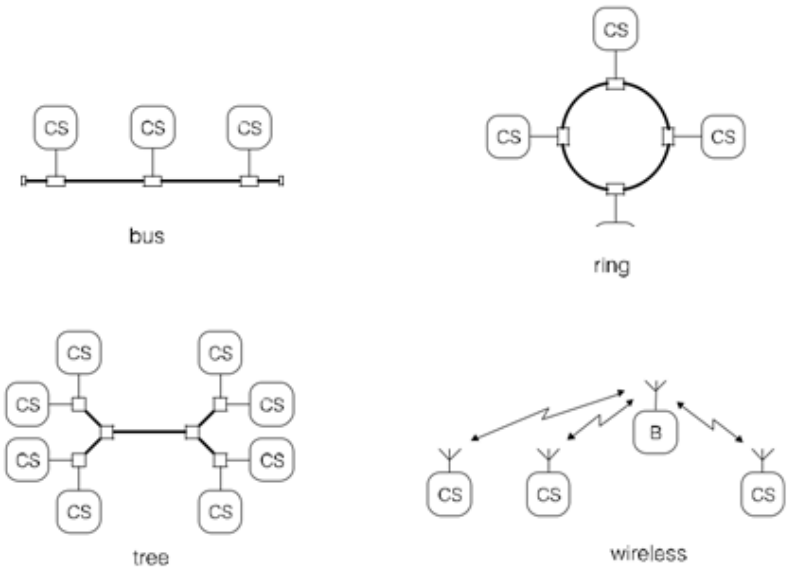
1.1.2 Topologie

Toplogií lokální sítě rouzíme způsob propojení jednotlivých stanic mezi ze sebou buď napřímo popřípadě prostřednictvím přepínačů (switchů). Topologií se klasické lokální sítě liší od rozsáhlých počítačových sítí. Ty se opírají o přepojování paketů nebo zpráv postupné předávání zpráv mezi uzly po dvoubodových spojích (technika store-and-forward) a jsou polygonální. Klasické lokální sítě využívají přímého propojení komunikačních stanic sdíleným kanálem, signál vyslaný jednou ze stanic je přijímán ostatními stanicemi sítě. Takové lokální sítě jsou někdy označovány jako broadcastové sítě. Volba topologie má vliv na řadu vlastností lokální sítě jako např:

- Rozšiřitelnost = možnost a snadnost doplňování stanic do existující sítě.
- Rekonfigurovatelnost = možnost modifikovat síť' při závadě komponenty nebo spoje.

- Spolehlivost = odolnost sítě proti výpadkům komponent nebo spojů složitost obsluhy a správ.
- Výkonnost = využití přenosové kapacity média, zpoždění zpráv.

V praxi se běžně setkáváme s topologií sběrnicovou, hvězdicovou, stromovou a kruhovou. Některé sítě jednotlivé topologie kombinují (stejně jako dnešní Ethernet).



Obrázek 1.1: Běžné topologie lokálních počítačových sítí

Sběrnicová

Základním prvkem sběrnicové sítě je úsek přenosového média — segment sběrnice, ke kterému jsou připojeny stanice sítě. Přenosovým médiem je nejčastěji koaxiální kabel nebo symetrické vedení (kroucený dvoudrát). U optických vláken je realizace odboček obtížná. Vlastnosti sběrnicové sítě lze shrnout do těchto bodů: pasivní médium, snadné připojování stanic, odolnost proti výpadkům stanic. Pro řízení sběrnicových sítí je využívána řada deterministických i nedeterministických metod které využívají faktu, že signál vysílaný jednou stanicí je přijímán ostatními stanicemi jen s velmi malým zpožděním.

Hvězdicová

Stanice sítě jsou připojeny k centrálnímu uzlu samostatnými linkami. Centrální uzel označovaný jako hub (v překladu "střed loukoťového kola") signál přicházející z jedné linky rozděluje do ostatních linek hvězdy. Rozlišujeme pasivní hub, ve kterém je signál pouze dělen (odporovým děličem), a aktivní hub (vícevstupový opakovač), ve kterém je přijatý signál upravován tak aby měl na výstupních linkách požadovanou úroveň a časování. Vlastnosti topologie hvězda lze shrnout takto: dvoubodové spoje mezi stanicemi a centrálním uzlem lze snadno realizovat, síť je odolná proti výpadku jednotlivých stanic a linek. síť je citlivá na poruchu centrálního uzlu.

Sítě s topologií hvězda, jak jsme si ji právě popsali, se tím, že signál jedné stanice mohou přijímat současně stanice ostatní, blíží sítím sběrnicovým a lze u nich použít i obdobné metody řízení. Topologie hvězda s pasivním centrálním uzlem často nacházíme u optických sítí.

Stromová

Stromová topologie je přirozeným rozšířením topologie typu hvězda. Setkáváme se s ní u širokopásmových sítí a u sítí využívajících pro přenos světlovody. Vlastnosti stromové topologie jsou podobné jako u sítí typu hvězda. Odolnost sítě proti výpadkům jednotlivých stanic a linek citlivost na výpadky uzlů, snadná rozšiřitelnost, dvoubodové spoje. Lokální stromové hvězdicové sítě používají podobných metod řízení jako sítě sběrnicové. U přístupových sítí převládá deterministická rezervace kanálu realizovaná centrálním prvkem.

Kruhová

U kruhových sítí jsou komunikační stanice propojeny spoji, které jsou využívány pouze jednosměrně. Signál vyslaný jednou stanicí je postupně předáván ostatními stanicemi kruhu (základní prvkem stanice je krátký posuvný registr) a po oběhu sítí se vrací ke stanici která jej odeslala.

1.2 Základní komunikační operace

Mezi stanicemi v síti mohou probíhat různé komunikační operace, které souvisí s používanými konkrétními síťovými službami resp. protokoly.

1.2.1 Unicast

Unicastová komunikace probíhá tak, že jedna stanice odesílá zprávu jiné konkrétní stanici. Unicastová komunikace je založena na použití unicastových (jednoznačných) adres.

1.2.2 Broadcast

Všesměrová (broadcastová) komunikace je specifická v tom, že jedna stanice odesílá jedinou zprávu, která je doručena všem stanicím patřícím do stejné skupiny. Pro broadcastovou komunikaci se v případě příjemce používá všesměrová (broadcastová) adresa.

1.2.3 Multicast

Skupinová (multicastová) komunikace probíhá tak, že jediná stanice odesílá zprávu, která je doručena určité skupině stanic. Rozdíl mezi všesměrovou (broadcastovou) a skupinovou

(multicastovou) komunikací je v tom, že v případě skupinové (multicastové) komunikace si stanice před začátkem doručování zpráv musí přihlásit jejich odběr.

1.2.4 Anycast

Anycastová komunikace probíhá tak, že jediná stanice posílá zprávu určité skupině stanic. Při posílání zprávy platí, že je zpráva doručena libovolné stanici spadající do dané cílové skupiny.

1.2.5 Manycast

Manycastová komunikace probíhá tak, že jediná stanice posílá zprávu určité skupině stanic. Při posílání zprávy platí, že je zpráva doručena více stanicím spadajícím do dané cílové skupiny.

1.3 Model ISO/OSI a jeho vrstvy

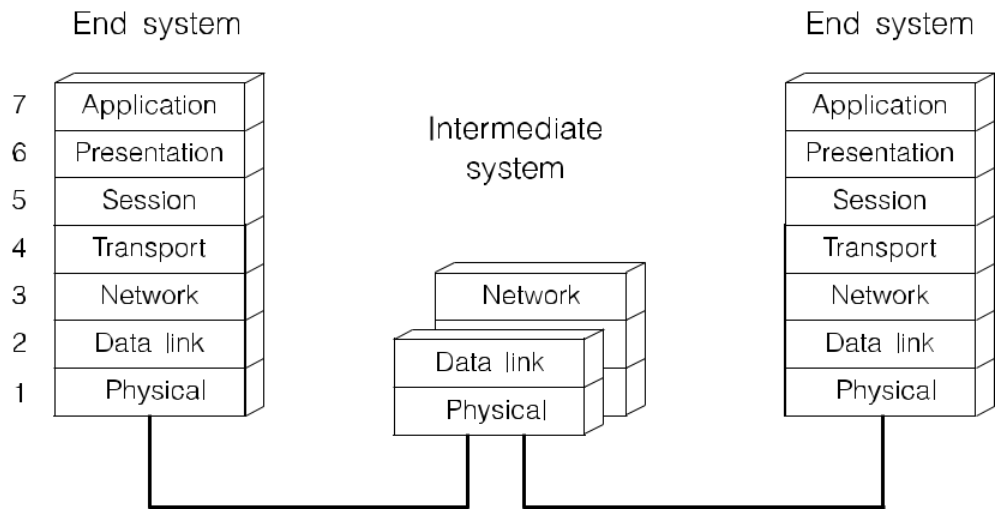
Současné lokální sítě se opírají o technologie, které jsou vesměs definovány standardy normalizačních organizací jako jsou IEEE (Institute of Electrical and Electronics Engineers), ETSI (Europian Telecommunications Standards Institute), ITU-T (International Telecommunication Union - Telecommunication Standardization Sector), ANSI (American National Standards Institute) a ISO (International Organization for Standardization). Patří sem např. varianty Ethernetu. Dobře definované a zavedené jsou standardy popisující použití sítí ATM (Asynchronous Transfer Mode) jako páteří lokálních sítí a standardy bezdrátových sítí.

Architektura ISO/OSI

Na síťové vybavení, technické a programové, jsme zvyklí se dívat jako na systém funkčních vrstev, ve kterém každá vyšší vrstva rozšiřuje možnosti vrstvy nižší. Důvodem takového rozkladu je složitost problémů, se kterými se v sítích setkáváme a které je třeba řešit pokud možno odděleně. Pro přepojovací počítačové sítě, ze kterých se na počátku osmdesátých let vyvinuly dnes provozované veřejné datové sítě, byl vytvořen standardní model síťové architektury označovaný jako ISO OSI (ISO Open Systems Interconnection). Architekturu vrstev modelu OSI ilustruje obr. 1.2.

Fyzická vrstva (Physical Layer) definuje fyzické propojení mezi prvky sítě, mechanické vlastnosti těchto propojení (konektory, typ média), elektrické vlastnosti (napěťové úrovně, způsob kódování a modulace) a u lokálních sítí i topologii propojení jednotlivých prvků a metodu přístupu k přenosovému médiu.

Linková vrstva (Data Link Layer) definuje pravidla pro předávání bloků dat. Zprávy jsou sítí přenášeny v pevně definovaných rámcích, rámce dovolují chránit předávaná data proti chybám při přenosu. U vícebodových spojů (a o ty se lokální sítě opírají) je nutné



Obrázek 1.2: Vrstvy síťové architektury ISO/OSI

zajistit linkovou adresaci stanic. Struktura rámce (ale spíš potřeba zajistit rozumné přidělování média) často limituje délku bloků dat.

Síťová vrstva (Network Layer) definuje způsob, jakým se sítí pohybují pakety, jak si je jednotlivé prvky sítě předávají na jejich cestě od odesílatele k adresátovi. Opírají se přitom o síťovou adresaci stanic, ta může být odlišná od adresace linkové. Mechanismy vrstvy se starají i o ochranu sítě proti nadměrné zátěži (Flow Control).

Transportní vrstva (Transport Layer) umožňuje současnou komunikaci více spuštěných programů na jednom počítači v síti, zajišťuje vytváření dočasných komunikačních spojení mezi aplikacemi a rozklad zpráv do paketů a skládání paketů do zpráv.

Relační vrstva (Session Layer) vytváří logické rozhraní pro aplikační programy, které používají služeb sítě. Definuje způsob komunikace programů a uživatelský pohled na komunikační kanál.

Prezentační vrstva (Presentation Layer) transformuje přenášená data zajišťuje převody kódů a formátů dat pro nekompatibilní počítače, kompresi a utajování přenášených dat.

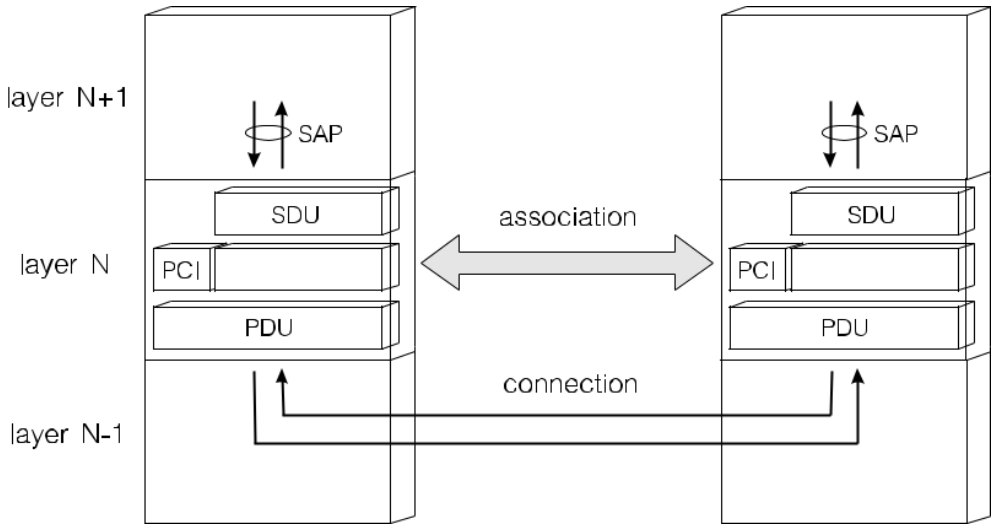
Aplikační vrstva (Application Layer) je vrstvou standardních aplikačních rozhraní a aplikačních programů, které síť využívají.

Model OSI se stal základem i pro lokální sítě, které používají jiných přenosových médií potvrzovacích technik a způsobů předávání zpráv, než starší sítě přepojovací.

1.3.1 Enkapsulace a dekapsulace dat

V rámci modelu ISO/OSI procházejí data postupně z vyšších vrstev do nižších. Průchod mezi vrstvami probíhá tak, že před data protokolu vyšší vrstvy je předřazena hlavička protokolu nižší vrstvy. Tento proces se opakuje rekurzivně až do doby, kdy data vyšších vrstev doputují k fyzické vrstvě. Tomuto procesu se říká enkapsulace dat.

Proces, který je inverzní enkaspulaci, se nazývá dekapsulace. Dekapsulace probíhá rekurzivně způsobem - data putují z nižších vrstev do vyšších tak, že jsou postupně zbavována hlaviček protokolů nižších vrstev dokud se nedostanou k vrstvě, která je vygenerovala.



Obrázek 1.3: Enkapsulace a dekapsulace dat v modelu ISO/OSI

Datové bloky předávané vyšší vrstvou Service Data Unit (SDU) jsou doplněny o řídící informace Protocol Control Information (PCI) a předávány ve formě datových bloků Protocol Data Unit (PDU) nižší vrstvě. Podobně v opačném směru, z doručovaných bloků PDU je jsou vybírána data SDU, řídicí informace PCI je využita procedurou řízení, která implementuje vrstvový protokol. Jako Service Access Point (SAP) označuje typicky entitu, která vyváří síťový provoz (např. běžící aplikace).

Rozdělením funkcí zajištovaných v síti (ochrana proti chybám přenosu, řízení toku dat, směrování) do jednotlivých vrstev má za cíl oddělit jejich řešení.

Kapitola 2

Fyzická vrstva

2.1 Přenosové médium

Jedním z důležitých prvků, který charakterizuje konkrétní lokální síť, je použité přenosové médium. Kromě malého počtu historických sítí, které používaly paralelní přenos po vícevodičových kabelech, jde u naprosté většiny dnešních sítí o přenos sériový. U některých technologií ještě najdeme nesymetrické vedení (koaxiální kabel), většina dnešních technologií se opírá o symetrické vedení (kroucený dvoudrát — twisted pair). Řada sítí se opírá o optická vlákna a ta jsou alternativním médiem i pro klasické (metalické) technologie. Významnou pozici získavají lokální sítě využívající vysokofrekvenčních rádiových spojů.

Koaxiální kabely

Nesymetrická vedení (koaxiální kabely) dovolují využití pásma 0 — 150 Mhz v základním pásmu (kódovaný datový signál) a pásma 50 — 750 MHz přeloženém pásmu (modulovaný signál). V základním pásmu lze dosáhnout přenosové rychlosti v rozmezí 1 50 Mb/s, v přeloženém pásmu lze vytvořit skupinu přenosových kanálů s přenosovou rychlostí až 40 Mb/s (pro kanál s televizní šířkou pásma 6 MHz). Při přenosu v základním pásmu omezují elektrické vlastnosti vedení překlenutou vzdálenost na stovky metrů, proto jsou často používány drahé speciální kabely (jako je tomu např. u sítě Ethernet 10BASE5). Přeložené pásmo lze využít pro přenos na kilometrové vzdálenosti, podstatnou výhodou je možnost použít kabely a další prvky určené pro kabelovou televizi. Koaxiální kabel byl po dlouhou dobu typickým médiem lokálních sítí má relativně dobrou odolnost proti rušení. Setkáme se s několika typy kabelů, které se liší charakteristickou impedancí (50 Q 75 Q a 93 Q), útlumem, ale i dalšími vlastnostmi, které ovlivňují jeho použitelnost.

Symetrická vedení - UTP STP

Symetrické vedení ve formě krouceného dvoudrátu (twisted pair), jak ho známe z telefonních kabelů, je nejlevnějším přenosovým médiem. Ve většině případů jde o stíněný (STP Shielded Twisted Pair) nebo nestíněný (UTP — Unshielded Twisted Pair), jednoduchý nebo dvojitý dvoudrát, který dovoluje bez problémů přenášet signály rychlých sítí, jako jsou sitě Ethernetu na vzdálenost 100 m, přenosové rychlosti jsou zde až 25 Gb/s.

Symetrické vedení je používáno pro přenos kódovaných signálů v základním pásmu. V průmyslových aplikacích se často setkáme s použitím napěťových úrovní odpovídajících standardním rozhraním RS-422 EIA a RS-485 EIA, varianty sítí Ethernet mají své vlastní standardy kódování, časování a úrovní datového signálu. Vlastnosti kabelů s kroucenými páry jsou definovány normami, nejpoužívanější standard EIA TIA 586 (z roku 1991) definuje vlastnosti kabelů UTP se čtyřmi dvoudrátovými vedeními. Dělí je podle mezního přenášeného kmitočtu (pro zvuk a obraz) nebo přenosové rychlosti do následujících kategorií (UTP Category):

- 3: Do 16 MHz nebo 10 Mb/s, je označován jako Voice Grade Cable, 4 - do 20 MHz nebo 20 Mb/s,
- 5: Do 100 MHz nebo 100 Mb/s, je označován jako Data Grade Cable.
- 5e (Cat 5e, Class D): Pro Fast Ethernet (100 Mbps, využívá dva páry) a Gigabit Ethernet (využívá všechny čtyři páry), používá stíněnou i nestíněnou kabeláž, šířka pásma 100 MHz, konektor RJ45.
- 6 (Cat 6, Class E): Podporuje 10 Gbps Ethernet, ale pouze do 55 metrů, používá stíněnou i nestíněnou kabeláž, šířka pásma 250 MHz, konektor RJ45. 6a (Cat 6a - augmented, Class EA): Podporuje 10 Gbps Ethernet na plnou vzdálenost 100 metrů, používá stíněnou i nestíněnou kabeláž, šířka pásma 500 MHz, konektor RJ45.
- 7 (Cat 7, Class F): Neujala se, pro podporu 10 Gbps Ethernet, vyžaduje stíněnou kabeláž (S/FTP) a konektory ARJ45, šířka pásma 600 MHz, nicméně výrobci raději zvolili Cat 6a a RJ45.
- 7a (Cat 7a, Class FA): Rozšíření Cat 7, mohlo by podporovat 40 Gbps Ethernet na vzdálenost 50 metrů, šířka pásma 1000 MHz.
- 8 (Cat 8): Aktuálně ve vývoji (ISO technické doporučení z roku 2013), podporuje 40 Gbps Ethernet, šířka pásma 1600 až 2000 MHz, dvě varianty, jedna používá stíněné kabely (U/FTP, F/UTP) a konektor RJ45, druhá kabely (F/FTP, S/FTP) a konektor ARJ45.

V současné době jsou používány téměř výlučně kabely odpovídající UTP Cat.5e, starší instalace používaly kabely UTP Cat.3 či 5. Moderní technologie dovolují vyrábět kabely, které překračují parametry vyžadované pro kategorii 5 (zadaný rozdíl mezi přeslechem na blízkém konci em NEXT a útlumem na mezní frekvenci). Pro vyšší kvalitativní třídy kabelů byly navrhovány další standardy - Cat.6 a Cat. 7. Frekvenční limity měly být podstatně vyšší - mezní frekvence 200 Mhz pro Cat.6 na kabelech UTP/ FTP a 600 Mhz pro Cat. 7 na kabelech STP. Vysoké nároky na řadu nových parametrů a citlivost na instalaci ukázaly, že v této oblasti již metalická vedení nejsou schopna konkurovat optice.

Strukturovaná kabeláž

V současnosti používané kabely UTP Cat.6 dovolují přenos signálu do kmitočtu 100 MHz. Kabely UTP se stávají i alternativou ke kabelům Shielded Twisted Pair (STP). Čtyřpárové kabely se společným stíněním označované jako Foiled Twisted Pair (FTP) fólií stíněné zkroucené páry nebo Screened Foiled Twisted Pair (SFTP) — FTP s ochranným opletením odolnější proti vlivu vnějšího rušení a omezující vyzařování přenášených signálů.

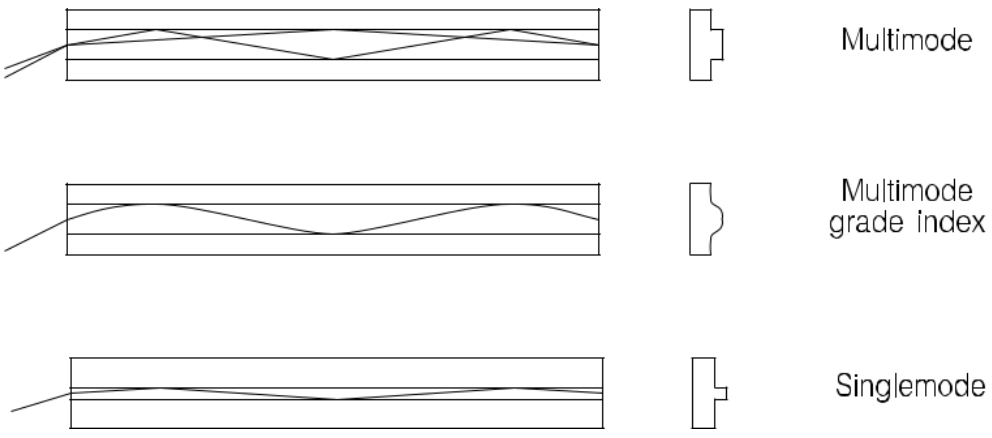
Kabely UTP (a jejich modifikace FTP a SFTP) jsou dnes považovány za univerzální materiál pro kabeláže, které kombinují přenos dat s přenosem telefonních signálů (analogových i digitálních) a videosignálů. Konkrétní lokální sít' lze vystavět poměrně jednoduše s využitím vedení takové univerzální strukturované kabeláže příslušným propojením na konektorových panelech (patch-panelech) v uzlech její většinou hvězdicové struktury.

Světlovodná vlákna

Světlovodná vlákna využívají infračervené a viditelné oblasti světelného spektra pro přenos dat rychlostmi do 400 Gb/s s na vzdálenost jednotek až desítek kilometrů. Výhodou optických vláken je vysoká přenosová kapacita při nízké ceně média a velká odolnost proti rušení nevýhodou je vysoká cena prvků rozhraní, konektorů a náročné spojování kabelů. S optickými vlákny se setkáváme v lokálních sítích s kruhovou nebo stromovou topologií.

Vícevidová optická vlákna jsou tvořena vnitřním jádrem (Core) o průměru do 100 μm a vnějším obalem (Cladding) z materiálu o nižším indexu lomu. Na rozhraní obou materiálů dochází k poměrně dokonalému odrazu přenášeného signálu. Materiálem jádra je převážně speciální sklo, obalem bývá sklo nebo plastická hmota. V technologických aplikacích jsou používána vlákna s plastovým jádrem i obalem. Vlákna jsou označována jako mnohavidová protože světelné paprsky se médiem šíří ve více videch charakterizovaných různými úhly odrazu. Takových diskrétních hodnot jsou u mnohavidových vláken tisíce. Důsledkem odlišných úhlů odrazu je rozdíl v absolvované délce cesty paprsku vláknem a z toho vyplývající rozptyl světelného výkonu v čase na výstupu z vlákna. Mluvíme o vidové disperzi, ta je hlavním limitem překlenutelné vzdálenosti. Limit vzdálenosti je uváděn jako součin délky vlákna a kmitočtu (MHz/km, GHz/km).

V praxi rozlišujeme historická vícevidová vlákna se skokovou změnou indexu lomu a modernější vlákna gradientní, u nichž je změna indexu lomu plynulá. Výhodou gradientních vláken je zvýšení podílu energie přenášené módy s většími úhly odrazu, zachování většího průměru jádra usnadňuje propojování vláken (ve srovnání s vlákny jednovidovými).



Obrázek 2.1: Optická vlákna

Jednovidová optická vlákna se vyznačují tím, že se při šíření světelného signálu uplatňuje jediný mód (nebo chceme-li být přesní, jde o dva módy lišící se polarizací).

Potřebného snížení počtu módů lze dosáhnout zvýšením vlnové délky světla (na 1300 nebo 1550 nrn), snížením poměru mezi indexy lomu jádra a obalu a snížením průměru jádra. Používaná jednovidová vlákna mají průměr vnitřního světlovodu kolem 10 μm (typicky používanými jsou vlákna 9/125 μm, horním limitem pro vlnové délky 1300 a 1550 nm a realizovatelné poměry indexu lomu je zhruba 15 μm). Jejich útlum bývá nižší než u mnohavidových vláken a pohybuje se kolem 0.55 dB/km na vlnové délce 1300 nm a až kolem 0.25 dB km na vlnové délce 1550 nm. Překlenutelná vzdálenost je až 100 km, šířka pásma až 100 GHz/km. Důležitým parametrem je zde chromatická disperze — závislost zpoždění signálu na vlnové délce signálu; ta se projeví více při použití světlo emitujících diod LED než při použití monochromatičtějších laserových diod TLD.

Optické kabely obsahují více vláken opatřených primární ochranou. Primární ochrana zvyšuje průměr vlákna typicky na 0.25 mm, je na vlákno nanášena bezprostředně po jeho vytažení a chrání materiál jádra před vlhkostí. Jako materiál primární ochrany je obvykle používán ultrafialovým světlem tvrditelný akrylát. Při potřebě práce ve větším teplotním rozsahu bývá akrylát nahrazen tenkou vrstvičkou polyamidu. Pro zvýšení odolnosti proti vlhkosti může být primární doplněna o tenoučkou uhlíkovou vrstvu nanesenou pod ní na vlákno.

Těsná sekundární ochrana vláken pro vnitřní použití má průměr typicky 0.9 mm a je tvořena vhodnou plastickou hmotou (polyamid, nylon). Kabely pro vnitřní použití pak ve své konstrukci ještě mají, obvykle kevlarové, prvky zachycující podélný tah, jako materiál vnějšího pláště vnitřních kabelů jsou používány materiály s nízkým obsahem halogenidů.

Kromě kabelů s těsným uložením vlákna v materiálu sekundární ochrany (většinou pro vnitřní použití) existují kabely s volným uložením vláken v konstrukci kabelu (většinou pro vnější použití). Vnější plást' kabelů pro vnější použití je obvykle polyetylénový, případně vyplněný gelem zabraňujícím přístupu vlhkosti.

Spojování vláken poněkud komplikuje instalaci optických spojů, přesně zakončená vlákna lze spojovat vzájemným přiložením konců, jejich slepením ve speciálních držácích nebo svařením. Je potřeba speciálních zařízení, realizované spoje je nutné proměřit (změřit útlum a případně odrazy ve spojích). Pro rozebíratelná spojení přesně zakončených vláken existuje škála různých konektorů, vedle starších typů ST a SC jsou dnes pro připojování koncových zařízení k dispozici rozměrově úsporné konektory LC MT-RJ a VF-45. Starší připojování již ve výrobě nakonektorovaných úseků vlákna (pigtails) je nahrazováno konektorováním při montáži. Potřebná úprava konce vlákna a montáž konektoru je však na technologii náročnější operací.

Jako zdroj světla pro světlovodné kabely jsou používány světloemitující diody Light Emitting Diode (LED) nebo rychlejší laserové diody Injection Laser Diode (ILD) — materiálem je GaAs nebo AIGaAs (850 mn) , InGaAs (1300 nm) a InGaAsP (1550 nm). Jako přijímače jsou používány fotodiody PIN nebo citlivější lavinové diody APD (Avalanche PhotoDiode) — materiálem je Si (850 nm), Ge a InGaAsP (1300 a 1550 nm).

Efektivitu napojení zdroje světla na vlákno ovlivňuje souhlas mezi průměrem zdroje světla a průměrem jádra. Do vlákna navíc mohou vstoupit pouze paprsky pod takovými úhly, které po průchodu rozhraním zdroj světla — jádro odpovídají rozsahu úhlů přenášených vláknem. Příslušné rozmezí úhlů definuje numerická apertura definovaná jako NA sine. Jak vysílače tak přijímače jsou dodávány bud' s úsekem připojeného vlákna (pigtail) nebo s připojeným optickým konektorem.

2.2 Datový kanál

Základním parametrem, který omezuje přenosovou rychlost, je šířka pásma použitého kanálu. Spojitý signál, který neobsahuje složky s vyšším kmitočtem než W, lze plně charakteri-zovat 2W vzorky za sekundu a z těchto vzorků signál opět rekonstruovat. Jinak řečeno, spojitým signálem s kmitočtovým spektrem omezeným kmitočtem W nemůžeme přenést více než 2W vzorků za sekundu. Může–li každý vzorek nabývat V diskrétních hodnot, pak pro přenosovou rychlost C platí Nyquistova věta.

C = 2W*log_2(V) [b/s; Hz]

Počet úrovní signálu v nelze s ohledem na poškození spojitého signálu při přenosu (obvykle toto poškození charakterizujeme přídavným signálem – šumem) libovolně zvyšovat; teoretický limit přenosové rychlosti C kanálu s pásmem o šířce W a odstupem signálu od šumu S/N udává Shannonova věta.

C = W*log_2(1 + S/N) [b/s; Hz]

2.3 Kódování a modulace

Neupravený datový signál (např. signál s úrovněmi TTL) není vhodný pro přímý přenos datovým kanálem. Obsahuje stejnosměrnou složku, jejíž přenos je obtížné zajistit, ať už pro elektrické vlastnosti kanálu nebo pro nutnost galvanického oddělení. Další nepříjemnou vlastností původního signálu je nezaručený výskyt napěťových změn (hran signálu), o které se lze opřít při vzorkování na straně přijímače.

Zajistění vzájemné synchronizace vysílače a přijímače mají za úkol metody bitové synchronizace. Tu lze zajistit několika způsoby. Mohli bychom například vedle vlastního datového signálu přenášet signál hodinový, který označuje místa, ve kterých máme vzorkovat. Rozumnější je však vybavit přijímač samostatným generátorem hodin a tento generátor fázově synchronizovat s přijímaným signálem. Podmínkou správné funkce fázového závěsu je dostatečný výskyt změn v přenášeném signálu, což zajistí vhodné kódování (např. kódy Manchester používané u starších lokálních sítí, nebo kódy 4B5B a 5B6B používané u moderních rychlých sítí).

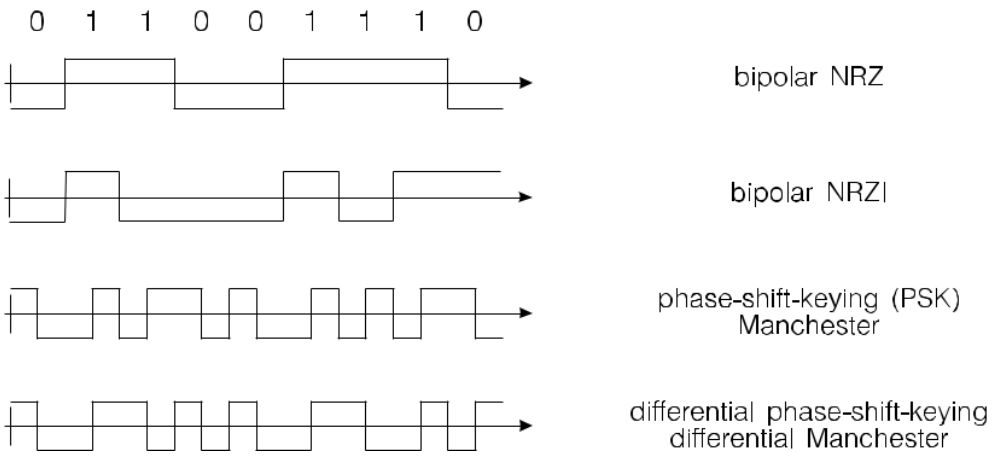
Datový signál můžeme zbavit stejnosměrné složky a doplnit o změny usnadňující jeho příjem vhodným kódováním. Starší metody kódování přitom berou při vytváření přenášeného signálu v úvahu jednotlivé bity .

Nejjednodušší kódování: bipolární NRZ, u kterého jsou bity zobrazeny jako dvě napěťové úrovně s opačnou polaritou, odstraňuje problémy spojené s přenosem stejnosměrné složky pouze částečně. Kódování NRZI, u kterého jsou jedničky (nebo naopak nuly) zobrazeny jako změny napěťové úrovně, odstraňuje závislost stejnosměrné složky na poměru počtu nul a jedniček v datovém signálu; množství synchronizační informace ale závisí na počtu jedniček (nebo naopak nul).

Fázovou modulaci NRZ (označovanou jako PSK nebo kód Manchester) používá lokální síť Ethernet. Diferenciální fázová modulace NRZ (označovaná také jako DPSK nebo diferenciální kód Manchester) byla použita v lokální síti IBM Token Ring. Tato kódování obsahují dostatek synchronizační informace (bity je kódovány jako hrany), vyžadují však

velkou šířku přenosového pásma.

Konečně, v případech, kdy chceme co nejlépe využít dostupnou šířku přenosového pásma (např. u rychlých lokálních sítí), saháme po kódováních posloupností přenášených bitů. Bitové posloupnosti o dané délce převádíme na delší bitové posloupnosti, případně na posloupnosti symbolů vícehodnotových (např. ternárních), tak, aby signál získaný jednoduchým kódováním NRZ nebo NRZI těchto posloupností měl nulovou stejnosměrnou složku a obsahoval dostatek hran (obr. 2.2).



Obrázek 2.2: Jednoduchá kódování datového signálu

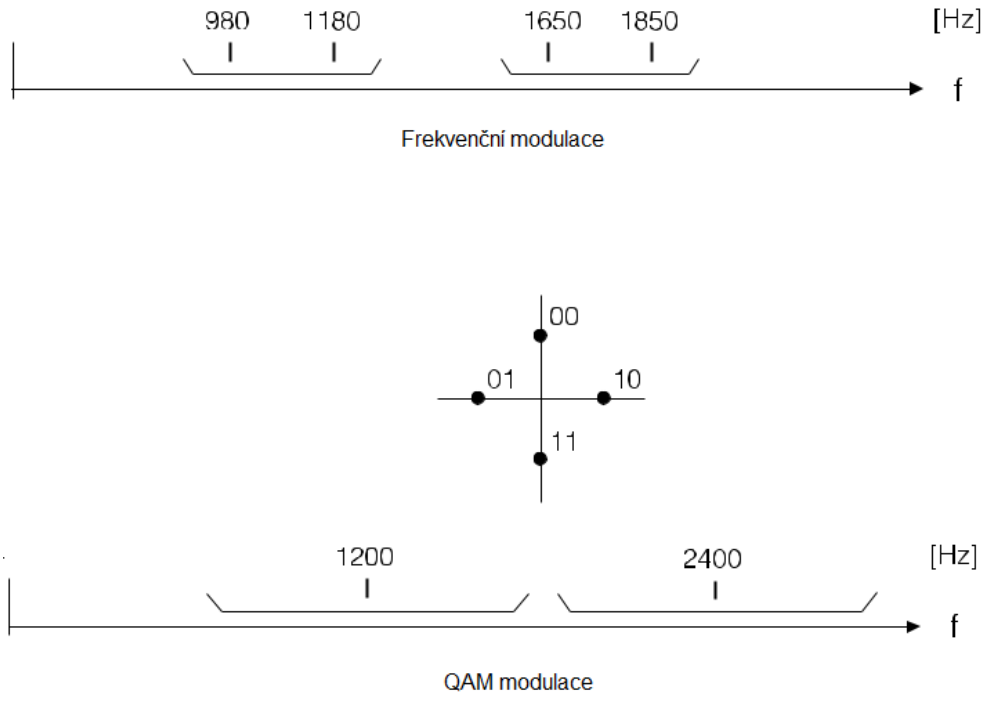
Náš příklad uvádí kódování 4B5B používané u rychlého Ethernetu (100BASE-Tx), u spojů gigabitového Ethernetu najdeme kódování 8B10B. Víceúrovňové kó-dování 8B6T používá varianta rychlého Ethernetu pro méně kvalitní kabely (100BASE-T4 pro kabely UTP Cat.3).

Přenos kódovaného datového signálu označujeme jako přenos v základním pásmu. Pokud chceme pro přenos využít kmitočtového pásma, které neobsahuje základní harmonické přenášeného datového signálu, musíme sáhnout k modulaci. Je-li nosným signálem harmonický signál

u(t) = U*sin(ω.t + ϕ);

můžeme modulací ovlivnit jeho amplitudu U, kmitočet ω, nebo fázi ϕ. Vzhledem k malé efektivitě kmitočtové modulace se s ní setkáváme u systémů, kde nám nezáleží na efek-tivním využití šířky pásma. Samotná amplitudová modulace je používána téměř výlučně v optických systémech. Tam, kde se snažíme o maximální využití kapacity přenosového kanálu používáme kombinaci fázové a amplitudové modulace (obvykle ji označujeme jako QAM) Příklady jednoduchých způsobů modulace použitých u starších modemových spojů (frekvenční modulace, fázová modulace) uvádí a příklady kombinované modulace amplitudové a fázové uvádí obrázek 2.3.

Frekvenční spektrum modulovaného harmonického signálu leží v jiné kmitočtové oblasti než spektrum signálu modulačního – mluvíme o přenosu v přeloženém pásmu.



Obrázek 2.3: Používané modulace v technologiích lokální sítí

Kapitola 3

Linková vrstva

3.1 Význam a části linkové vrstvy

Hlavním úkolem linkové vrstvy je doručování dat v lokální síti. K tomuto je samozřejmě nutné umět přistupovat k přenosovému médu, definovat logickou adresaci a vypořádat se s chybami, které mohou během přenosu dat nastat. Linková vrstva se dělí na dvě podvrstvy a sice:

- Medium Acces Control (MAC): Podvrstva, která zajišťuje přístup k přenosovému médiu a definuje adresaci.
- Logical Link Control (LLC): Podvrstva, která se stará o logické řízení toku a zabezpečení dat oproti chybám.

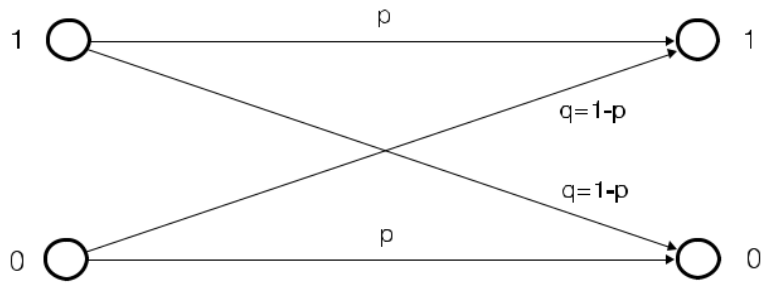
3.2 Chyby v přenosovém kanále

Chybovost přenosového kanálu je dána chybami ukončujících zařízení, ale hlavně chybovostí vlastního přenosu. Chybovost přenosu je důsledkem poškození signálu na médiu a má řadu příčin: frekvenční a fázovou charakteristiku přenosového kanálu, nelineární zkreslení, vliv tepelného šumu, impulsní rušení, přeslechy z jiných přenosových kanálů, odrazy na nepřizpůsobeném vedení či úniky signálu na rádiových spojích, atd.. Vliv jednotlivých zdrojů chyb se značně liší, a při vážnější práci je musíme respektovat.

Vliv bílého šumu na signál přenášený kanálem lze dostatečně realisticky popsat jednoduchým modelem - symetrickým binárním kanálem bez paměti (obr. 3.1).

Symetrický binární kanál přenáší vstupující bit signálu (nulu nebo jedničku) bezchybně s pravděpodobností p. Doplňková pravděpodobnost q = 1 - p je pravděpodobností chyby.

Předpokládejme, že bílý šum je hlavním zdrojem chybovosti. Ta se např. u telefonních linek použitých pro přenos dat pohybuje v mezích 10^{-5} (jakostní pevné spoje) až 10^{-3} (velmi nekvalitní komutovaná linka). Za předpokladu statistické nezávislosti jednotlivých bitových chyb bude pravděpodobnost správného přenosu rámce o délce N bitů dána výrazem



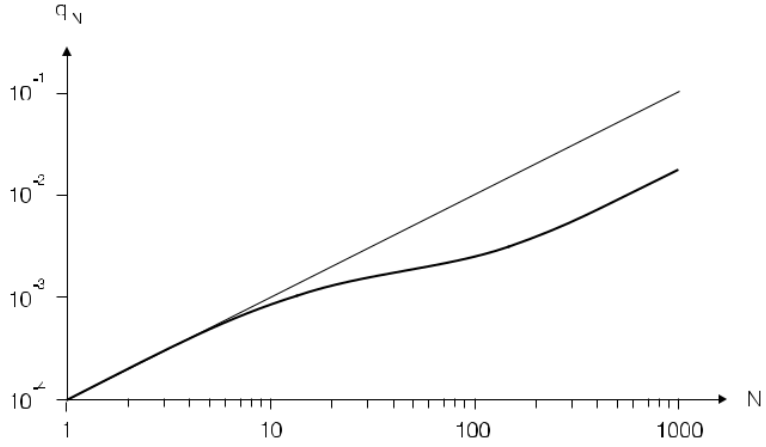
Obrázek 3.1: Symetrický binární kanál bez paměti

$$\mathrm{N} = p^N = (1-q)^N$$

Použijeme-li například nekvalitní telefonní linku (q = 10^{-3}) (256 bitů), bude pravděpodobnost jeho přenosu bez chyby pro přenos rámce o délce 32 znaků pouze

$$\mathrm{p256} = (1-10^{-3})^{256} = 0.774$$

Pro reálné přenosové kanály naštěstí náš předpoklad, že chybovost má na svědomí bílý šum neplatí. Převažující vliv mají přeslechy, impulsní rušení a přenosové charakteristiky, vliv těchto faktorů je podstatně odlišný.



Obrázek 3.2: Závislost četnosti chyb na délce rámce

Ani sebelepší zabezpečovací kódy nejsou schopné neindikovanou chybu zcela vyloučit. Pro rozumné přenosové kanály (s chybovostí q < 10^{-4}) je však pravděpodobnost neindikované chyby prakticky zanedbatelná (10^{-8}).

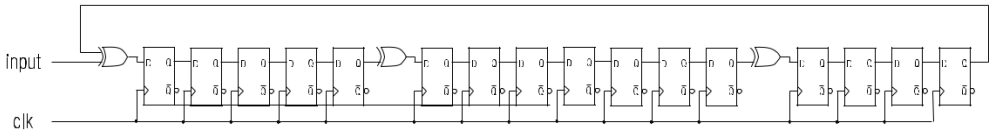
3.2.1 Bezpečnostní a samopravné kódy

Zatímco bílý šum poškozuje jednotlivé bity, např. impulsní rušení poškodí celé posloupnosti bitů. Díky tomu, že se chyby vyskytují ve shlucích, je pravděpodobnost bezchybového přenosu rámců vyšší.

Ochrana proti chybám se opírá o detekční (výjimečně samoopravné) kódy. Byly a jsou využívány paritní kódy, iterační kódy a kódy cyklické. Volba kódu by měla odpovídat charakteru chyb převažujících v daném kanále, např. zajištění pouhou paritou je u přenosů málo účinné. Pro přenosový kanál z obr. 3.1 je navíc vysoká pravděpodobnost toho, že se po jedné chybě objeví další chyba v následujících bitech.

Použijeme-li proto pro zajištění osmibitových znaků přenášených takovým kanálem paritní kontrolu (neindikuje sudý počet chybných bitů) zjistíme, že paritní kontrola nezjistí chybu u 38% poškozených znaků. Použití samotné parity je pro přenosové kanály nepostačující, samotná parita není schopna detekovat shluky chyb, pro datové kanály typické. Lepší vlastnosti mají iterační kódy (např. kombinace podélné a příčné parity) a zvláště kódy cyklické (CRC - Cyclic Redundancy Code). Výhodou cyklických kódů je jejich snadná technická implementace, obr. 3.3 uvádí schéma generátoru pro polynom CRC-CCITT definovaný generujícím polynomem

$$g(x) = x^{16} + x^{12} + x^5 + 1$$



Obrázek 3.3: Generátor CRC-CCITT

3.3 Sdílení přenosového média

Pokud přenosové médium poskytuje větší šíři pásma (větší přenosovou rychlost) než je potřebné pro realizaci jediného přenosového kanálu, lze médium sdílet více přenosovými kanály. V lokálních sítích se používají 4 základní typy sdíleného přístupu (multiplexu) k médiu. Jmenovitě se jedná o multiplex: časový - Time Division Multiple Access (TDMA), frekvenční - Frequency Divison Multiple Access (FDMA), kódový - Code Division Multiple Access (CDMA) a prostorový - Space Division Multiple Access (SDMA).

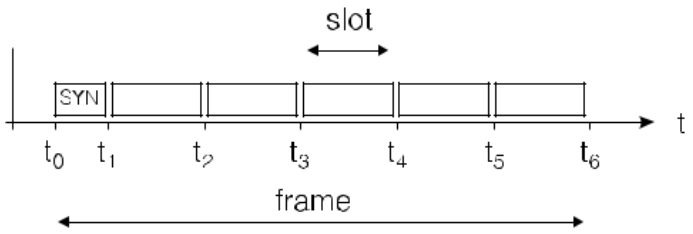
3.3.1 Časový multiplex (TDMA)

U TDMA přidělujeme přenosový kanál postupně jednotlivým stanicím. Každé stanici je vyhrazen časový úsek (slot), ve kterém může vyslat paket určité délky. časové úseky jednotlivých stanic se pravidelně střídají s periodou kterou obvykle označujeme jako rámec (frame).

Pro přenos dat zřejmě nelze plně využít kapacitu kanálu, v každém časovém slotu je nutné věnovat čas na sfázování přijímače a rámec je nutné doplnit synchronizačním slotem. Metoda je použitelná pro lokální sítě s malou rozlehlostí.

Nevýhodou pevného rozdělení kapacity sdíleného kanálu je neschopnost přizpůsobit využití kanálu nárazovému charakteru požadavků jednotlivých stanic. Optimálního využití kapacity bychom dosáhli v případě, že bychom měli k dispozici algoritmus, který

by evidoval požadavky jednotlivých stanic a přiděloval podle nich stanicím médium. V ideálním případě bychom dosáhli chování obslužného systému M/M/ 1 (označujeme ho tak v případě náhodně přicházejících požadavků na přenos náhodně dlouhých bloků dat po jednom kanálu). Tomu se mužeme vhodnými metodami řízení do určité míry přiblížit — mluvíme o asynchronním časovém multiplexu (ATDMA — Asynchronous TDMA, Adaptive TDMA). Porovnání středního zpoždění, ke kterému dojde při přenosu sítí s frekvenčním multiplexem, sítí se synchronním časovým multiplexem a sítí s ideálním přidělováním typu M/M/1 uvádí obr. 3.4.

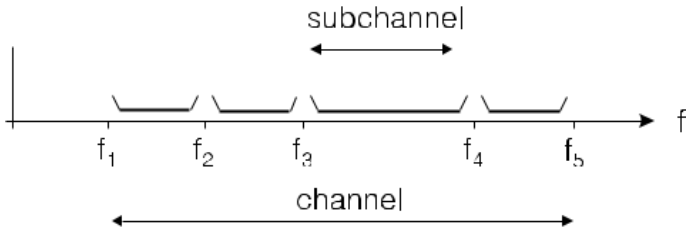


Obrázek 3.4: Časový multiplex

TDMA je dnes snadněji realizovatelný než FDMA, a jeho adaptivní formy (sdílení datového kanálu takovým způsobem, aby bylo maximálně využito jeho kapacity) jsou principem převážné většiny lokálních sítí.

3.3.2 Frekvenční mutiplex (FDMA)

FDMA se využívá skutečnosti, že pro přenos dat s danou přenosovou rychlostí vystačíme s určitou šíří frekvenčního pásma. Jeli šíře pásma, kterou nám poskytuje přenosový kanál, větší, lze kanál rozdělit na více podkanálů a každý z nich použít nezávisle. Pro převod datového signálu do daného frekvenčního pásma a zpátky používáme modemů vybavených selektivními filtry. Frekvenční multiplex je základem širokopásmových lokálních sítí.



Obrázek 3.5: Frekvenční multiplex

3.3.3 Kódový multiplex (CDMA)

Základem CDMA je to, že médium využívá v konkrétním časovém okamžiku vícero účastníků. Účastníci, kteří využívají stejný kanál, však zpracovávají pouze to, čemu rozumějí (svému kódu).

3.3.4 Prostorový multiplex (SDMA)

SDMA je založeno na tom, že médium využívá současně několik různých účastníků, díky tomu, že komunikační kanály prochází prostorově různými vzájemně se nepřekrývajícími směry.

3.3.5 Metody s příposlechem nosné (CSMA)

U lokálních sítí které se vyznačují malým zpožděním signálu a dokonalou slyšitelností stanic, lze podstatně omezit pravděpodobnost kolize tím, že stanice nezahájí vysílání, dokud přenosové médium využívá některá jiná stanice. Metody, které znalost obsazení kanálu využívají, nazýváme metodami náhodného přístupu s příposlechem nosné, zkráceně metodami Carier Sense Multiple Access (CSMA). Základní princip je pro všechny varianty společný a spočívá v tom, že stanice před vysíláním dat krátce naslouchá přenosovému médiu a podle jeho využití se rozhodne, zda skutečně vysílat začne.

CSMA/CD

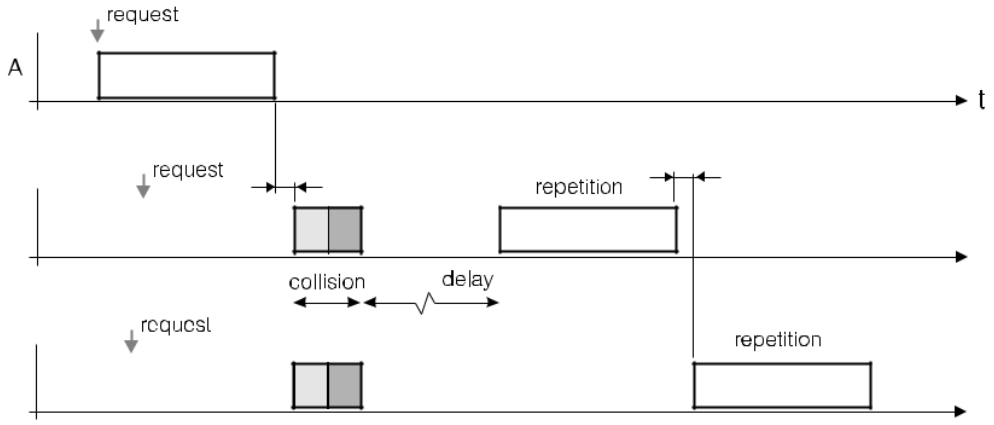
CSMA není schopné zabránit kolizi, je-li časový interval mezi zahájením vysílání dvou stanic menší než jistá mez, daná konečnou rychlostí šíření signálu v kanále, vzdáleností stanic a rychlostí reakce detekčních obvodů. U naléhající CSMA je navíc při větší zátěži velice nepříjemné, že dojde-li během vysílání rámce více než jeden další požadavek, je výsledkem kolize (bezprostředně po uvolnění kanálu). Kolize, které u dlouhých rámců blokují po dlouhou dobu přenosový kanál, snižují dosažitelnou průchodnost. Zlepšení lze dosáhnout, dokážemeli je detekovat a předčasně zastavit vysílání. Příslušné metody označujeme jako Carrier-Sense Multiple Access with Collision Detection (CSMA/CD).

Použití metod CSMA/CD vyžaduje použít kanálu, na kterém lze kolizi zjistit. Nejjednodušším kanálem, který detekci kolize umožňuje, je sběrnice typu otevřený kolektor. V praxi však obvykle kolizi detekujeme jinak, například sledováním napětí na médiu, které je buzeno proudovými zdroji vysílačů (Ethernet 10BASE5) nebo sledováním signálu na krouceném páru přijímače (10BASE-T).

Stanice, která má připravený rámec k vyslání a detekuje klid na sdíleném kanále po definovanou dobu označovanou jako kolizní slot, zahájí vysílání synchronizační posloupnosti.

CSMA/CA

U dosud popisovaných metod jsme neuvažovali potřebu potvrzování přijatých rámců (přesněji řečeno, neuvažovali jsme, že potvrzení budou muset soupeřit o přidělení kanálu). Na potvrzení se můžeme dívat jako na nutnou přídavnou zátěž, která pouze v určitém poměru sníží čistou průchodnost sítě. Chceme-li eliminovat nepříjemný vliv této přídavné zátěže na soupeření stanic o kanál, můžeme pro potvrzení rezervovat časový interval bezprostředně navazující na vyslání datového rámce a zajistit, že žádná ze stanic nesmí v tomto intervalu zahájit vysílání rámce nového. Taková modifikace bývá označována jako CSMA/CA (Collission Avoidance).



Obrázek 3.6: Princip přístupové metody CSMA/CD

CSMA/CA nalezneme i u rádiových sítí podle IEEE 802.11. Zde je vyčleněna vedle potvrzování ještě prioritní komunikace s prodlevou kratší než pro běžný provoz. Protože se stanice nemusí navzájem slyšet, lze navíc pro vysílání delších rámců využít mechanismus označovaný jako RTS/CTS. Stanice před startem vlastního vysílání požádá o přidělení kanálu krátkým rámcem Request to Send (RTS) a dostane od základnové stanice souhlas Clear to Send (CTS).

3.4 Adresace na linkové vrstvě a rámec linkové vrstvy

Rámec linkové vrstvy je základní přenášitelnou jednotkou linkové vrstvy. Adresy používáné na úrovni linkové vrstvy se označují MAC adresy. Dvojice MAC adres (zdrojová a cílová) jednoznačně určují způsob komunikace.

Rámec linkové vrstvy začíná specifickou sekvencí bitů, která se označuje jako začátek rámce (Frame start) za níž následují dvě adresní pole se označující cílovou a zdrojovou stanici, za kterými dále následují vlastní přenášená data. Celý rámec je zakončen sekvencí bitů, která se označuje jako přívěsek (Trailer).

Rámce linkové vrstvy jsou zpracovávány zařízeními, které se označují jako mosty nebo přepínače.

3.5 Mosty, přepínače a přepínání rámců

Lokální sítí lze propojit pouze omezený počet stanic, častým limitem je i nejvyšší překlenutelná vzdálenost. Tu omezuje jednak délka jednoho úseku přenosového média a jednak maximální počet opakovačů mezi stanicemi. U sítě Ethernet je třeba dodržet nejvyšší vzdálenost mezi stanicemi 2.5 km a do sítě připojit nejvýše 1024 stanic. Při větších požadavcích na rozlehlost sítě, na počet stanic nebo na kombinaci různých síťových technologií nezbývá, než jednotlivé menší sítě mezi sebou propojit prvkem který převede komunikaci z jedné sítě do sítě druhé.

Důvody k rozdělení stanic do více sítí a k propojení těchto sítí mohou být i jiné, než

překročení uvedených limitů. Rozdělení stanic do více sítí, pokud možno tak, aby se co nejvíce přenosů uskutečnilo uvnitř sítí dovolí dosáhnout vyšší celkové průchodnosti (zvyšuje kapacitu sítě) a nižší doby odezvy. Poruchu v jedné lokální síti lze v propojovacím prvku rozpoznat, její vliv se ve zbytku soustavy neprojeví. Izolace sítě proti poruchám v jejích částech zvyšuje spolehlivost. Provoz mezi stanicemi jedné sítě není propojovacím prvkem zbytečně do druhé sítě přenášen, propojovací prvek tak zajišťuje ochranu komunikace stanic proti odposlechu zvyšuje bezpečnost.

Lokální sítě propojujeme pomocí prvků, připojených ke dvěma nebo více propojovaným sítím, soustavu více propojených lokálních sítí obvykle nazýváme internetwork. Prvky propojující lokální sítě označujeme jako mosty (Bridges), přepínače (Switches) a směrovače (Routers). Funkce mostů a směrovačů je podobná funkci uzlů přepojovací sítě, a obvykle ji charakterizujeme termínem store-and-forward. Rámce přijaté z připojených sítí jsou analyzovány a podle výsledku bud' likvidovány nebo následně vyslány do některé (některých) ze sítí. Přepínače (Switches), které dovolují zahájit vysílání bezprostředně po analýze hlavičky rámce, funkci charakterizujeme termínem cut-through.

Mosty, přepínače a směrovače se od sebe liší rozsahem informace, kterou pro další kroky využívají. Mosty a přepínače se opírají pouze o adresační pole rámce (MAC adresy), směrovače analyzují předávaná data a využívají informací spojených s konkrétním síťovým nebo transportním protokolem. Existují i kombinované prvky — broutery (Bridging Routers anebo L3 switches), které pro některý síťový nebo transportní protokol fungují jako směrovače a pro jiné protokoly jako mosty či přepínače.

3.5.1 Mosty a přepínače

Most přijímá všechny rámce z propojovaných sítí a u každého z nich se rozhoduje, zda ho do druhé sítě přenese (adresát je v této druhé síti nebo je neznámý), nebo zda ho bude ignorovat (adresát je v síti, z níž byl rámec přijat).

Při rozhodování se most řídí MAC adresou příjemce a přepínacími tabulkami, ve kterých má uloženy informace o rozmístění stanic v sítích připojených k mostu (u mostu se statickými tabulkami a u mostů transparentních), nebo údaji uloženými v MAC rámci (u zdrojového směrování). Adresu MAC (a pochopitelně ani v MAC rámci přenášená data) běžný most nemění. Lze ho tedy použít pro propojení sítí respektujících jeden formát rámců, a lišících se nejvýše médiem. Mosty mohou brát v úvahu při svém rozhodování o tom, zda rámec přenést, i další informace, například typ rámce Ethernetu, adresu odesílatele nebo adresáta. Pak mluvíme o selektivní filtraci, produkty jednotlivých výrobců se v této oblasti značně liší.

Most sleduje veškerý provoz v sítích, které propojuje. Vede si evidenci stanic, jejichž adresy jsou uvedené jako adresy odesílatele. Tato evidence má formu směrovací tabulky (Forwarding Database). Pro každou adresu, která se objevila v poli odesílatele rámce, je ve směrovací tabulce uvedena síť, ze které zpráva s touto adresou přišla. Ukládaní do tabulky je označováno jako učení (Bridge Learning).

Na každou zprávu, která je přijata mostem z některé připojené sítě, most reaguje některým ze tří způsobů:

1. zpráva určená pro stanici, o níž most ví že leží ve směru odkud byla zpráva přijata,

je likvidována.

2. práva určená pro stanici, o níž most ví že leží v jiné síti, než ze které byla zpráva přijata, je mostem převedena do této sítě.

3. zpráva určená všem stanicím (broadcast) nebo zpráva určená stanici, kterou most dosud nezná, je rozeslána do všech směrů, kromě směru, ze kterého přišla.

Pro uložení směrovacích tabulek má most vyhrazenu oblast paměti; velikostí této paměti a způsobem jejího rozdělení na jednotlivé tabulky se mosty od sebe liší.

Typickou velikostí paměti je prostor pro 4096 až 16384 položek v jediné směrovací tabulce pro všechna rozhraní. Přístup je obvykle opřen o jednoduchou adresační funkci (např. hashing, výběr pole dvanácti až čtrnácti bitů z adresy MAC), důsledkem mohou být pochopitelně kolize — opakované přepisování záznamů ve směrovací tabulce a následné zbytečné rozesílání datových rámců do sítí, do kterých nepatří.

U přepínačů, které mají s běžnými mosty hodně společného, jsou běžné samostatné tabulky pro jednotlivá rozhraní, v krajním případě s kapacitou omezenou až na jedinou položku.

Transparentní most pracuje pouze v sítích se stromovou strukturou, v níž uzly reprezentují mosty a hrany reprezentují propojované lokální sítě. V propojených sítích nesmí vzniknout uzavřená cesta — cyklus. Pokud potřebujeme propojit sítě více mosty a zajistit tak odolnost proti jejich výpadkům, musí být tyto mosty schopné vypnout některá svá rozhraní a vytvořit tak stromovou strukturu (kostru propojovací sítě). Postup, kterého mosty při takovém omezování topologie využívají, je označován jako Spanning Tree algoritmus.

Blokované porty mostů zůstávají v záloze pro případ výpadku některého mostu nebo sítě. Algoritmus výběru kostry se opírá o jednoznačnou číselnou identifikaci mostů, distribuovaný výběr fungujícího mostu s nejnižší identifikací a o nalezení stromu nejkratších cest s vybraným uzlem jako kořenem. Je standardizován specifikací IEEE 802.1D.

Vlastní algoritmus výběru kostry se opírá o již uvedenou jednoznačnou identifikaci mostu, opřenou např. o výrobcem přidělené adresy řadičů Ethernetu a o cenu výstupu (ohodnocení výstupních portů). Služební rámce, které si mosty si mezi sebou vyměňují při konstrukci kostry, mají zvláštní formát a jsou označovány jako BPDU (Bridge Protocol Data Unit).

Prvním krokem algoritmu je výběr kořene. Každý z mostů může rozeslat rámec BPDU s vlastní identifikací do všech připojených sítí. Každý z mostů tak může zjistit, zda je jeho identifikace nejnižší a je tedy kořenem kostry. Most kořen kostry rozesílání rámců BPDU periodicky opakuje.

Kořen kostry v rozesílaném rámci uvádí jako cenu cesty cenu přiřazenou příslušnému výstupu. Rozhraní, na kterém most sousedící s kořenem přijímá jeho rámce BPDU, označujeme jako root port. K údaji o ceně cesty v rámci BPDU most přičte cenu svého výstupu a rámec vyšle dál. Jako výsledek opakování tohoto kroku může každý z mostů určit svůj root port.

Pro každou z propojovaných lokálních sítí je dále potřeba určit most s nejnižší cenou cesty ke kořeni kostry. To je snadné vzhledem k údaji o ceně cesty v rámcích BPDU.

Rozhraní tohoto mostu označujeme jako vyhrazené (Designated). Rozhraní R (root port) a D (designated port) vytvářejí kostru, ostatní rozhraní přecházejí do blokovaného stavu a neúčastní se přenosu datových rámců (rámce BPDU však přijímají a vysílají).

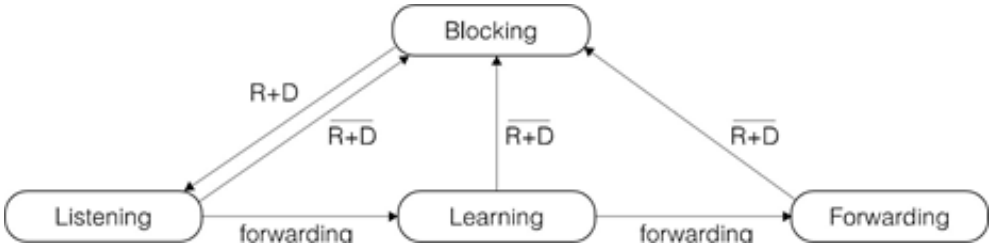
Kořen kostry v rozesílaném rámci uvádí jako cenu cesty cenu přiřazenou příslušnému výstupu. Rozhraní, na kterém most sousedící s kořenem přijímá jeho rámce BPDU, označujeme jako root port. K údaji o ceně cesty v rámci BPDU most přičte cenu svého výstupu a rámec vyšle dál. Jako výsledek opakování tohoto kroku může každý z mostů určit svůj root port.

Pro každou z propojovaných lokálních sítí je dále potřeba určit most s nejnižší cenou cesty ke kořeni kostry. To je snadné vzhledem k údaji o ceně cesty v rámcích BPDU. Rozhraní tohoto mostu označujeme jako vyhrazené (Designated).

Rozhraní R (root port) a D (designated port) vytvářejí kostru, ostatní rozhraní přecházejí do blokovaného stavu a neúčastní se přenosu datových rámců (rámce BPDU však přijímají a vysílají).

Přechod mezi blokováním portu a jeho běžnou činností je poněkud komplikován nutností zabránit nekorektnímu přenosu datových rámců při změnách topologie. Přechod z provozního stavu do blokování proběhne okamžitě, přechod z blokovaného stavu do provozního stavu je řízen časovačem Forwarding Timer a navíc procházíme stavem, ve kterém si most pouze aktualizuje přepínací tabulky.

Jak už jsme uvedli, rozesílání rámců BPDU kořenem stromu je periodické (perioda je označována jako Hello Time). Při běžném provozu mosty evidují, že je vše v pořádku; výpadek některého z mostů nebo portů může vyvolat změnu root portu a vyhrazeného rozhraní. Každou takovou změnu most hlásí kořeni stromu zvláštním rámcem BPDU a ten hlášení po určitou dobu potvrzuje zvláštním příznakem v rozesílaných rámcích BPDU. Příjem rámce BPDU s nastaveným příznakem zneplatňuje (po zadaném čase) údaje v tabulkách mostů.



Obrázek 3.7: Stavový diagram transparentního mostu

Víceportové mosty (připojené více než dvěma síťovými rozhraními do více než dvou lokálních sítí) jsou dnes častěji označovány jako přepínače Switches (jejich použití je především v přepínaném Ethernetu). Označení plně přísluší pouze těm mostům, které umožňují zahájit vysílání přenášeného rámce ještě před dokončením jeho příjmu. Metodu cut-throuqh použila jako první firma Kalpana. Výhodou metody je snížení zpoždění rámce proti klasickému mostu při malé zátěži, nevýhodou je, že jsou přenášeny i poškozené rámce. Při velké zátěži není přínos metody podstatný a modernější označení přepínač je (spíše z reklamních důvodů) používáno i pro klasické víceportové mosty pracující s technikou store-and-forward.

Vzdálené mosty

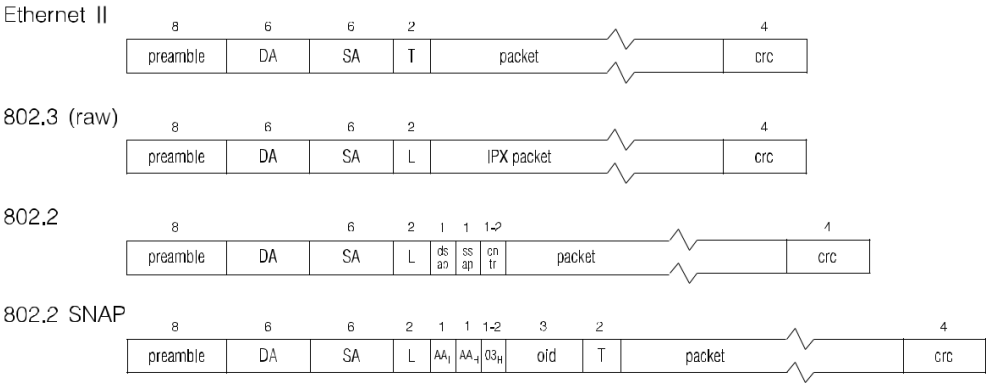
Někdy potřebujeme propojit lokální sítě na větší vzdálenost dvoubodovým spojem a pochopitelně chceme po tomto spoji přenášet pouze rámce určené vzdáleným stanicím. Řešením je umístění dvou mostů na konce dvoubodového spoje, jejich směrovací tabulky však budou identické a filtrace rámců přicházejících z dvoubodového spoje bude zbytečná. Redukce funkcí těchto dvou mostů vede na řešení, označované jako (Remote Bridge). Most se rozhoduje o převedení rámce lokální sítě do dvoubodového spoje, v opačném směru přenáší všechny rámce.

Podobnou redukci funkcí jako u vzdálených mostů nalezneme u mostů určených pro oddělení provozu malých skupin stanic od zbytku sítě. Takový most bývá označován jako Workgroup Bridge, jeho směrovací tabulka obsahuje pouze informace o adresách stanic skupiny, všechny stanice s adresami mimo skupinu leží implicitně na druhé straně mostu.

Možnost použití mostů k propojení sítí na větší vzdálenost je zajímavá, je však nutné vzít v úvahu limitovanou kapacitu dvoubodového spoje a fakt, že mosty přenášejí provoz typu broadcast a multicast. Efektivnější využití limitované kapacity proto často přináší použití směrovačů.

3.6 Technologie Ethernet

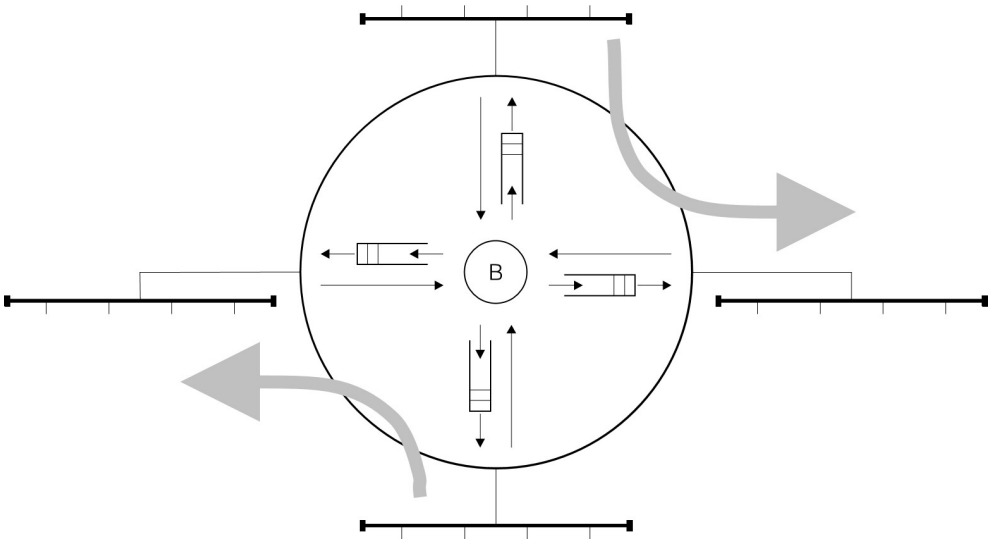
Základy technologie, známé jako Ethernet, byly položeny ve vývojových laboratořích Xerox Palo Alto Research Center začátkem 70. let. V roce 1980 byl Ethernet standardizován konsorciem firem DEC, Intel a Xerox, standard je známý pod zkratkou DIX a Ethernet II. Současně začaly práce na standardu IEEE, jehož prvá verze byla publikována v roce 1985 pod označením IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specification. Standard byl později podstatně rozšiřován o další média a nové způsoby provozu. Dnes je Ethernet standardizován i normou ISO 8802 3.



Obrázek 3.8: Formát rámce Ethernetu

3.6.1 Přepínání v Ethernetu

Mosty Ethernetu dovolují rozdělit rozsáhlejší síť na kolizní domény, provoz v jedné části sítě nemá vliv na provoz v části druhé a součtový tok v síti může být vyšší než je limit v každé z kolizních domén. U více-portových mostů které propojují čtyři a více kolizních domén, se objevuje další zajímavý efekt. Přenos rámců mezi dvěma kolizními doménami přes takový most neblokuje jiný přenos mezi jinými dvěma kolizními doménami přes týž most. Při větším počtu portů a možnosti rozdělit síť na menší kolizní domény (mluvíme o segmentaci sítě, ale tomuto termínu se budeme snažit vzhledem ke kolizi s běžně používaným pojmem segment vyhýbat) je tento efekt silnější. Takové prvky běžně označujeme jako přepínače (Ethernet Switches). Technologii, využívající přepínačů ke zvýšení průchodnosti sítě označujeme jako přepojovaný Ethernet (dáváme tomuto termínu přednost před termínem přepínaný Ethernet).



Obrázek 3.9: Princip přepínání v Ethernetu

V krajním případě se můžeme dostat až k situaci, kdy na každý port přepínače je připojena jediná stanice a takto využívané přepínače jsou propojené dvoubodovými spoji (v síti nejsou více-portové opakovače ani sběrnicové segmenty s více než dvěma připojenými prvky), mluvíme o mikrosegmentaci. Taková síť funguje prakticky stejně jako každá jiná síť s přepojováním paketů. Pouze místo paketů (jako v X.25 nebo Internetu) jsou zde přepojovány rámce Ethernetu (a opíráme se o adresaci linkové vrstvy) a s ohledem na jednodušší topologii (pro provoz je využitelná pouze stromová podsíť získaná použitím Spanning-Tree algoritmu podle IEEE 802.1D) se zjednodušuje směrování. Rámce přijaté z jednotlivých vstupů jsou ukládány do pamětí přepínače, po rozhodnutí o způsobu odeslání a případné úpravě směrovací tabulky (přepínač se učí rozložení stanic v síti) převedeny do front na výstupech a odesílány do výstupních kanálů. Tento postup je označován jako Store-and-Forward.

Určitou nevýhodou techniky Store-and-Forward je zpoždění, způsobené tím, že rámec může být vysílán do výstupního kanálu až po jeho dokončeném převzetí. Zpoždění lze eliminovat, dovolíme-li přepínači zahájit vysílání do neobsazeného výstupního kanálu okamžitě jakmile přepínač přečte adresu příjemce (prvních šest slabik rámce za preambulí). Využití této myšlenky (dlouho známé v teorii přepojovacích sítí jako Virtual-Cut-Through

a využívané v paralelních počítačích) je známé jako technika Cut-Through a dovolí snížit zpoždění rámce při průchodu přepínačem až na 12 μs (proti 58-1220 μs u metody Store-and-Forward, kde záleží na délce rámce). Takové zlepšení může vypadat jako velký přínos a urychlilo rozšíření přepínaného Ethernetu, ale při zatížené síti, kdy v přepínačích vznikají fronty rámců, nemusí být rozdíl mezi oběma metodami podstatný.

Metoda Cut-Through má však i zápory. Patří mezi ně skutečnost, že odeslán je i rámec, u kterého bude při jeho příjmu zjištěna chyba CRC (v době, kdy přepínač zahajuje vysílání předávaného rámce, ještě nebyl zabezpečovací kód na konci rámce přijat). Další problém vyvolávají kolize na vstupech, přepínač zahájí vysílání rámce, který nebude díky zafungování detekce kolize přijat celý. Tento problém lze poměrně jednoduše řešit tak, že vysílání zahájíme až po převzetí dostatečného počtu znaků, tedy až budeme mít jistotu, že přijímaný rámec dojde celý (bylo přijato 64B a vysílání rámce již nepřeruší detekce kolize). Úprava metody Cut-Through, která brání předání krátkých fragmentů rámců na výstup (a jejich dalšímu šíření sítí) je označována jako Fragment-Free a typické minimální zpoždění přepínače je 58 μs.

Pokud jde o reálné prvky, označované jejich výrobci jako přepínače Ethernetu, je potřeba si uvědomit, že mezi nimi existují podstatné rozdíly, které omezují jejich nasazení:

Nejširší použití mají přepínače, na jejichž vstupy lze připojovat celé kolizní domény (tvořené víceportovými opakovači nebo sběrnicovými segmenty). Takové přepínače dovolují realizovat přepínání označované termínem Segment Switching a bývají někdy označovány jako Corporate Switches. Pokud potřebujeme mít v síti náhradní spoje pro zvýšení spolehlivosti, musíme mít jistotu, že přepínač splňuje požadavky IEEE 802.1D (umí Spanning Tree Algoritmus)

Přepínačům, které počítají s připojením jediné stanice na každý vstup a které budou připojeny jediným rozhraním na zbytek sítě, stačí jednodušší přepínací tabulky (jedna adresa pro každý vstup, implicitní adresace pro rozhraní zbytku sítě). Přepojování je označováno jako Link Switching, přepínače bývají označovány jako Workgroups Switches a jsou využitelné pro mikrosegmentaci.

3.6.2 Řízení toku v Ethernetu

Přepínače v síti Ethernet zvyšují sumární průchodnost sítě, je s nimi však spojen jeden problém: kapacity přepínačů jsou konečné a při přetížení některého z rozhraní může dojít ke ztrátám rámců, které není kam uložit. Výrazné zvýšení kapacity pamětí přitom není řešením, oddálí riziko ztrácení paketů, ale za cenu zvýšeného zpoždění rámců čekáním ve frontách rozhraní.

U přepínačů s rozhraním v poloduplexním režimu se ztrátám rámců můžeme bránit odmítáním rámců, které není kam uložit. Lze toho dosáhnout dvěma metodami:

- vyvoláním kolize na vstupním rozhraní přepínače, z něhož nechceme přebírat rámce.
- vysíláním výplňových rámců do rozhraní, z něhož nechceme přebírat rámce.

Nevýhodou první metody je skutečnost, že opakovaná kolize vede na ustupování, nelze proto rychle reagovat na zlepšení situace, navíc po překročení limitu kolizí v sekvenci může dojít k indikaci výpadku linky a jejímu případnému vyjmutí z topologie sítě.

Výhodou je pouze možnost rozlišit rámce, které mají být směrovány do zahlcených výstupů, od rámců, které problémy nevyvolávají.

Druhá metoda dovoluje sice rychlou reakci na zlepšení situace, protějšek může začít okamžitě po uvolnění média vysílat, v žádném případě však není možné diferencovat mezi rámci. Podstatným problémem obou metod, označovaných jako backpressure metody, je skutečnost, že zahlcení jednoho přepínače je vede k přenesení problému na jeho okolí (přepínače v rozsáhlejší síti). Důsledkem se může stát zahlcení rozsáhlejších částí sítě a v důsledku i omezení datových toků, které přes přepínač, který zahlcení vyvolal nevedly.

Backpressure metody jsou použitelné pouze u poloduplexních spojů, u duplexu jsou z principu nepoužitelné. Proto byla pro řízení toku navržena podstatně pružnější metoda opírající se o přenos řídících MAC rámců - rámců PAUSE.

Rámce PAUSE se od datových rámců liší polem Type, ve kterém najdeme typ protokolu 8808, a které označuje skutečnost, že se jedná o řídící rámce MAC vrstvy. Rámce PAUSE jsou jedním konkrétním typem řídících rámců MAC, pole Opcode je u nich nastaveno na hodnotu 0x0001. Rámce PAUSE být směrovány všem zdrojům ve směru konkrétního rozhraní, pro tento účel je vyhrazena multicast adresa 01:80:C2:00:00:01. Alternativně lze rámce vysílat s unicast adresou konkrétního zdroje dat a tak selektivně omezit tok o sítě.

Nejdůležitější údaj přenášený v PAUSE rámci je informace o době, po kterou chceme pozastavit vysílání do spoje. Tento čas se udává jako násobek doby potřebné pro vyslání 512 bitů, tato volba vychází z alternativního využití obvodů u poloduplexu používaných pro mechanismus exponenciálního ustupování.

Pominou-li důvody pro pozastavení vysílání, lze rámcem PAUSE s nulovou hodnotou doby pozastavení vysílání okamžitě uvolnit. U vysokorychlostních spojů (mechanismus je navržen i pro gigabitový Ethernet a technologie ještě rychlejší) je nutné brát v úvahu zpoždění linek a množství dat, které do nich mohlo být vysláno (například kapacita dvou kilometrů gigabitového spoje je 20000 bitů).

U řízení toku v lokálních sítích odlišujeme dva typické scénáře. U prvního jsou špičky přenosu krátkodobé a nemají svůj zdroj převážně na jedné straně spoje. V takovém případě je rozumné dovolit symetrické řízení toku tedy oba prvky na spoji si mohou v případě hrozícího zahlcení posílat rámce PAUSE.

Alternativní asymetrické řízení toku je vhodné u koncových zařízení, hraniční přepínač sítě se omezením datového toku ze stanice muže bránit přetížení sítě, stanice proti tomu nemůže ztěžovat práci zbytku sítě blokováním toku.

3.6.3 Ethernet 10Mb/s

Standard IEEE 802.3 definuje fyzické médium, algoritmus přístupu a formát přenášených rámců.

Nejnižší úroveň standardu je označována jako rozhraní Medium Dependent Interface (MDI) a definuje přenosové médium (tím dnes může být koaxiální kabel, kroucený dvoudrát nebo optické vlákno), signál na médiu a konektor. Přenosové médium podstatně ovlivňuje vlastnosti sítě. Jednotlivým technologiím lišícím se (hlavně) médiem jsou při-

dělena jména konstruovaná tak, že zahrnují informaci o rychlosti přenosu, signálech na médiu a dalších charakteristických vlastnostech. Jako příklady jmen technologií si můžeme uvést historickou technologii 10BASE5 (přenos rychlostí 10 Mb/ s v základním pásmu s délkou segmentu 500m) a 1OOBASE-FX (přenos rychlostí 100 Mb/s v základním pásmu po optickém vlákně). Aktivní prvek, který vysílá a přijímá signál přenosového média, běžně známý jako transceiver (TRANSmitter-reCEIVER) má v normě označení Medium Attachment Unit (MAU).

Jednotka MAU je připojena rozhraním Attachment Unit Interface (AUI) k vlastní stanici, počítači vybavenému řadičem Ethernetu. Rozhraní AUI definuje: speciální (nepříliš ohebný) kabel, se čtyřmi kroucenými dvoudráty o impedanci 78 Ω přenášejícími signál vysílaný, signál přijímaný, signál detektoru kolize a napájecí napětí, konektor, kterým je upravený 15-ti špičkový Canon DB-15 s bajonetovým zámkem na místě zajišťovacích šroubků a jeho zapojení a elektrické signály rozhraní a zajištění izolace do 500 V (10BASE2) nebo 2000 V (10BASE5).

Stanice, která má připravený rámec k vyslání a detekuje klid na sdíleném kanále po dobu alespoň 9.6 μs, zahájí vysílání synchronizační posloupnosti a potom odešle vlastní rámec rychlostí 10 Mb/s. Stanice, která chce vysílat, ale indikuje provoz na médiu, musí počkat na uvolnění média a uplynutí ochranného intervalu 9.6 s. Tento postup je označován jako naléhající CSMA. Stanice začíná vysílat po uvolnění média bez nějaké další podmínky, v případě sítě Ethernet je základní mechanismus ještě doplněn o detekci kolize (CSMA/CD). Ta dovoluje podstatně snížit ztráty způsobené kolizí stanic, které čekaly na uvolnění média a kolizi si tím "naprogramovaly"stanici, která vstoupila do kolize a tuto skutečnost rozpoznala, se pokusí o opakované vysílání po náhodně zvolené době se střední hodnotou rovnou délce kolizního intervalu (51.2 ps). Náhodná volba odmlky brání periodickému opakování kolize stanic. Pokud k opakované kolizi dojde, stanice prodlužuje střední dobu prodlevy na dvojnásobek. Po deseti neúspěšných pokusech přestane prodlevu prodlužovat a po šestnácti hlásí závadu vyšším vrstvám obsluhy (Může jít o odrazy na přerušeném nebo zkratovaném kabelu porouchanou některou ze stanic segmentu, apod.). Postup označovaný jako exponenciální ustupování (Exponential Back-off) je navržen tak, aby zajistil stabilitu sítě pro alespoň 1024 stanic. To je také limit, který stanovuje norma pro skupinu segmentů propojených opakovači kolizní doménu.

Signál přenášený po médiu je kódován tak, že jednotlivým bitům odpovídají hrany signálu kód známe pod jménem Manchester. Vysílače fungují jako zdroje proudu, na kabelu s pevně definovanou charakteristickou impedancí detektor kolize se pak opírá o měření střední hodnoty signálu na kabelu. Podle nastaveného limitu je schopen detekovat kolizi vysílající stanice s jinou stanicí na kabelu (Transmit Mode), nebo kolizi dvou jiných stanic na stanici v klidu (Receive Mode). Pro testování detektoru kolize muže transceiver vysílat po příslušném vedení AUI kabelu indikaci kolize po odvysílání rámce (1 us po ukončení po dobu 1 NS), funkce je označována jako SQE Test nebo Heartbeat. Další přídavnou funkcí stanice je Jabber Control, schopnost vypnout vysílač, pokud doba jeho vysílání překročí 20 ms, a to na dobu 500 ms. Tato funkce brání trvalému obsazení média při poruše transceiveru.

Přístupová metoda Ethernetu CSMA/CD se opírá o informace, které je stanice schopna získat pozorováním sítě. Vzhledem ke konečné době šíření signálu v přenosovém médiu a ke zpožděním v opakovačích se však jedná o informace nepřesné čímž efektivita metody CSMA/CD klesá s rostoucí vzdáleností stanic. Proto je standardem omezena jak

vzdálenost po médiu tak i počet opakovačů mezi každými dvěma stanicemi. Překročení limitů může být důvodem podstatného zvýšení počtu kolizí a počtu poškozených rámců a tím i výsledného snížení průchodnosti sítě.

Formát rámce jsme si již popsali, za upozornění pouze stojí, že formát rámce podle normy DIX se poněkud liší od formátu rámce podle IEEE 802.3. Zatímco IEEE Ethernet uvádí v hlavičce délku LLC bloku, DIX Ethernet zde identifikuje síťový protokol (např. IP, IPX). Odlišení obou typů rámců je možné díky tomu, že délka datového pole je omezena na 1500B (tato hodnota se označuje jako MTU - Maximum Transmission Unit) a údaj o délce tak může být nejvýše 5DCH, zatímco označení typu využívá hodnot od 800H (kromě některých historických identifikací protokolů, jimž se lze v praxi vyhnout).

3.6.4 Ethernet 10BASE-T

Koaxiální kabely jako médium pro výstavbu sítí Ethernet se stávají historií. Důvodem je přechod k 'I levnějšímu a univerzálnějšímu kabelu UTP (Unshielded Twisted Pair) a k odlišnému způsobu vytvoření sdíleného kanálu. úseky UTP kabelu o délce do 100 m (přesněji do 90 m pevného rozvodu a dvakrát 5 m pohyblivý kabel pro připojení zařízení) propojují jednotlivé stanice s vícevstupovým opakovačem (Multiport Repeater, koncentrátor). Ten je středem hvězdice tvořené skupinou až osmi, dvanácti, šestnácti nebo i více stanic. Technologie dostala název 10BASE-T (T jako Twisted Pair) a je specifikována doporučením IEEE 802.3i z roku 1990.

Ze čtyř párů kabelu UTP jsou využity dva, jeden pár přenáší signál od stanice k opakovači druhý přenáší signál ve směru opačném. Kabel UTP musí splňovat podmínky na šířku pásma, charakteristickou impedanci a přeslech. (Přeslech signálu z vysílacího vedení do přijímacího může být považován za kolizi.) Podmínky splňují kabely UTP Cat.3 (Voice Grade) a s rezervou dnes běžnější kabely UTP Cat.5 (Data Grade), ty lze při správné montáži použít i pro sít 1OOBASE-TX. Jako konektor (zásuvky karet, zásuvky pro pevný rozvod, zástrčky na kabel) slouží plochý konektor EIA RJ45 (podobný telefonnímu konektoru podle americké normy EIA RJ-II).

3.6.5 Ethernet 100BASE-T

Výrazně technologickou modifikací hvězdicového Ethernetu 10BASE-T se stal standard označovaný jako 1OOBASE-T zvyšující přenosovou rychlost na 100 Mb/ s na kabelovém rozvodu UTP/FTP Cat.5 (modifikace 1OOBASE-T4 vystačí dokonce i s UTP Cat.3) a na vícevidových optických vláknech (62.5/125 um a 50/125 um). Specifikace rychlého Ethernetu pod označením IEEE 802.3u byla schválena v červnu 1995. Rychlý Ethernetu je založen na efektivnějším využití přenosového média. Kódování Manchester je u technologií IOOBASE-TX/FX nahrazeno efektivnějším kódováním 4B5B, se kterým jsme se již setkali u sítě FDDI doplněným o víceúrovňové kódování pro přenos po metalických vedeních (MLT-3 Multi-Level Transmit). Ještě výraznějšího zvýšení efektivity dosahují technologie 100BASE-T4 a 100BASE-T2.

Vzdálenost mezi stanicí a koncentrátorem je, pokud použijeme metalický kabel, stejně jako u sítě 10BASE-T, do 100 m. Optické vlákno dovolí jít až na 412 m (mezi dvěma stanicemi nebo mezi stanicí a přepínačem) při poloduplexním a na 2000 m při

duplexním provozu. U poloduplexního přenosu je omezením doba šíření signálu médiem: signál musí proběhnout médiem do nejvzdálenějšího místa sítě a zpět (včetně časů potřebných pro elektronice koncových prvků a opakovačů) za dobu potřebnou k odeslání 512 bitů. Při návrhu sítě rychlého Ethernetu se používá jednotka označovaná jako bittime. U optického vlákna odpovídá jeden metr vlákna jednomu bittime, u metalických kabelů s menší rychlostí šíření signálu jeden metr kabelu odpovídá 1.1 bittime.

Rychlý Ethernet definuje tři rozdílné realizace fyzického kanálu. Základem jsou kanály 1OOBASE-TX - dva páry kabelu UTP/FTP a 1OOBASE-FX - dvojice optických vícevidových vláken. Zajímavým doplňkem normy je kanál 100BASE-T4, který využívá tří párů kabelu UTP Cat.3 k přenosu dat a čtvrtého páru k detekci kolize. Později byl doplněn standard 100BASE-T2 který vystačí i u kabelů Cat.3 se dvěma páry.

S ohledem na různá řešení fyzického rozhraní Physical Medium Dependent (PMD) je pro rychlý Ethernet definováno rozhraní mezi fyzickou vrstvou a vrstvou MAC. To je označováno Medium Independent Interface (MII) a má šířku čtyř datových bitů. Pro toto rozhraní je sice definován čtyřicetipinový konektor rozhraní MII, rozhraní je však, na rozdíl od AUI využíváno pouze jako standard rozhraní obvodů na desce síťového rozhraní (největší vzdálenost 0.5 m). Často je zcela skryté uvnitř obvodu.

3.6.6 Ethernet 100BASE-FX

Metalická vedení jsou dodnes levnější variantou kabeláže lokálních sítí technologie rychlého Ethernetu však již předpokládá použití optických vláken, konkrétně vícevidových optických vláken datových (62.5/125 μm) nebo telekomunikačních (50/125 μm). Aby bylo možné překlenout vzdálenosti shodné s technologiemi 10BASE-FL/FB, používá IOOBASE-FX světlo o vlnové délce 1300 mn.

Při poloduplexním přenosu je vzhledem k vidové disperzi a použití mnohavidových vláken nejvyšší vzdálenost omezena na 412 m. Na větší vzdálenost, až do 2 km, je nutné pracovat v duplexu, ten je však v moderních přepojovaných sítích standardně podporován.

Dvoukilometrový limit dovoluje i ve velkých budovách realizovat přepojovanou síť s architekturou jednoúrovňové hvězdy, koncová zařízení a servery sítě jsou samostatnými vlákny připojeny k centrálně umístěných přepínačům. Takové řešení často snižuje náklady, protože nevyžaduje aktivní prvky, přepínače nebo opakovače na patrech. Jeho praktické využití usnadňuje zjednodušení technologií přímého připojování maloformátových optických konektorů (LC, MT-RJ, VF-45) na optické kabely.

Datový signál je pro přenos IOOBASE-FX kódován obdobně jako u IOOBASE-TX, tedy nejdříve přeložen kódérem 4B5B, vlastní signál optického vlákna je z výstupu kódéru 4B5B získán překódováním NRZI (jednička je reprezentována změnou, použitý kód 4B5B zaručuje nejvýše tři nuly za sebou).

Na rozdíl od jiných technologií rychlého Ethernetu nemusí zařízení IOOBASE-FX podporovat automatickou volbu přenosové rychlosti, kompatibilita s rychlejšími technologiemi Ethernetu pracujícími na vlnové délce 1310 nrn je zajišťována na straně gigabitového Ethernetu.

3.6.7 Ethernet 100BASE-SX

Ethernet IOOBASE-FX pracuje s vlnovou délkou 1300 nm a dovoluje překlenout na vícevidových vláknech v duplexním provozu vzdálenost až 2 km. Nepříjemnou vlastností této technologie je nekompatibilita se staršími technologiemi FOIRL a 10BASE-FL/FB.

Technologie IOOBASE-SX je modifikací rychlého Ethernetu, pracuje na 850 nm, tedy na vlnové délce shodné se staršími technologiemi. Koncová zařízení dovolují automatickou volbu přenosové rychlosti, i když poněkud odlišnou od systému využívaného u technologie 1OOBASE-TX. Zařízení mohou, pro omezení vidovou disperzí a tedy i při duplexu, komunikovat rychlostí 100 Mb s na vzdálenost nejvýše 300 m.

Výhodou 1OOBASE-SX proti technologii 1OOBASE-FX jsou také poněkud levnější vysílací a přijímací diody. Technologie je proto považována za možnou alternativu metalických spojů pro připojování koncových zařízení v klasické strukturované kabeláži.

3.6.8 Gigabitový Ethernet

Přenosová rychlost 100 Mb s nezůstala nadlouho limitem. Z iniciativy skupiny výrobců známé jako GEA (Gigabit Ethernet Alliance) byl vytvořen standard sítě založené na principech Ethernetu s přenosovou rychlostí 1 Gb/s - IEEE 803.z. Technologicky se gigabitový Ethernet opírá o ověřené technologie vyvinuté původně pro spoje Fiber Channel.

Podobně jako standard rychlého Ethernetu předpokládá gigabitový Ethernet více typů přenosového média. Základním médiem je vícevidové vlákno (62.5 um, 50 um) pracující na vlnové délce 780 nm (IOOOBASE-SX). Alternativní vlnovou délkou pro vícevidové vlákno je 1300 nm (IOOOBASE-LX), na této vlnové délce lze využívat i jednovidová vlákna a překlenout vzdálenosti i více než 3 km. Pro propojování zařízení na vzdálenost do 25 m lze využít kabel typu Twinax, dvoudrátové vedení s dobře definovanou impedancí proti vnějšímu plášti (1000BASECX).

Dodatečně byl standard gigabitové Ethernetu rozšířen i na typické přenosové médium pomalejších sítí, na kabely UTP/FTP (IOOOBASE-T) - IEEE 802.3ab.

3.6.9 10ti Gbitový Ethernet

Modifikace technologie Ethernet na vyšší přenosové rychlosti mnohem častěji využívají plně duplexního provozu a přepojování. Jejich přenosová rychlost výrazně překračuje hodnoty potřebné většinou koncových zařízení, a spolu se schopností překlenout větší vzdálenosti (jednotky až desítky kilometrů) jsou chápány jako technologie metropolitních sítí, ale i sítí rozsáhlejších. Typickým příkladem takového posunu je zvýšení přenosové rychlosti na 10 Gb/s, tyto sítě už jsou často využívány i jako alternativa k typickým WAN technologiím.

Standard desetigigabitového Ethernetu IEEE 802.3ae zahrnuje řadu variant využívajících různých přenosových kanálů. Na rozdíl od technologií pomalejších jde výlučně o optická vlákna a je podporován výlučně plně duplexní provoz.

3.7 Pasivní optické sítě

Technologie EFM dovoluje dosáhnout přenosové rychlosti vyšší než 10 Mb/ s, pro budoucí služby by však i tato rychlost mohla být nepříjemným omezením. Již i proto jsou v současné době připravovány standardy pro takzvané optické pasivní sítě. Výhodou optických pasivních sítí (PON - Passive Optical Network) je to, že nepotřebují relativně náročný (a napájený) aktivní prvek na rozhraní optického vlákna a metalického vedení. Pochopitelným kritériem návrhu je minimalizace počtu vláken, kterými je propojen provozovatel připojení k telekomunikačním sítím s velkým množstvím koncových účastníků.

Z hlediska koncového účastníka výhodnější variantou je samostatné připojení. Jsou využívána jednovláknová připojení. Vlákno je využíváno v režimu širokopásmového vlnového multiplexu (vlnové délky 1300 nm a 1500 nm). Takové připojení jednovidovým optickým vláknem dovoluje be problémů dosáhnout přenosové rychlosti 100 Mb/s nebo 1 Gb/s.

I když je varianta samostatného připojení pro koncového účastníka zajímavější, její nevýhodou je vedení jednovidového vlákna ke každému zakončeni. Výhodnější možností je požití pasivního optického rozbočovače pro skupinu účastníků v malé lokalitě. Zakončení pak mohou být realizována vícevidovými vlákny, nevýhodou řešení je samozřejmě sdílení kanálu více účastníky.

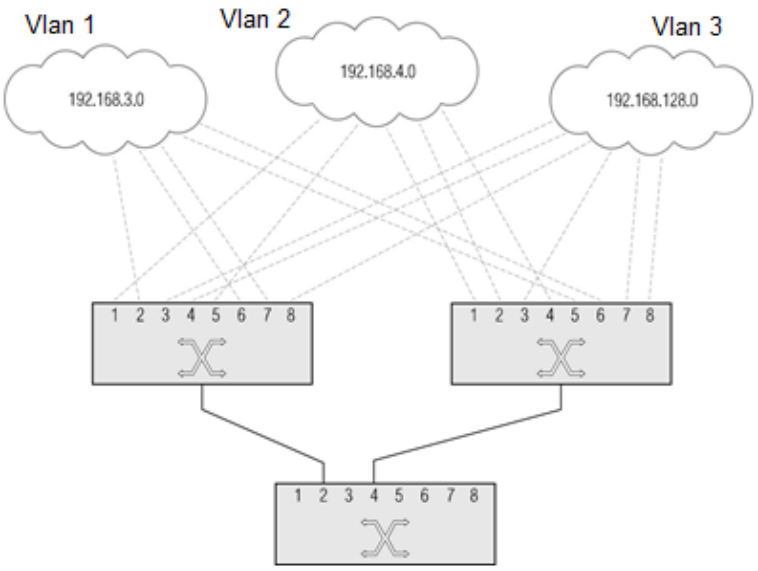
Poměrně zajímavou oblastí je způsob využití pásma poskytovaného pasivním optickým kanálem. Situace je složitější u sdílených kanálů, u kterých je potřebné zajistit řízení přístupu k dostřednému kanálu. Využívána je zde metoda obdobná technice používané u sítí využívajících distribučních sítí CATV. Jedno optické vlákno přivedené do objektu je pasivním rozbočovačem napojeno na vlákna vedoucí ke koncovým zařízením.

Pokud jde o základní formátování dat na optickém vedení, v současné době spolu soupeří tři přístupy - APON, EPON, GPON. Technologoie APON (ATM over PON) se opírá 0 buňky ATM, technologie EPON (Ethernet over PON) o rámce Ethernetu a konečně telekomunikační unií standardizovaná technologie GP ON kombinuje synchronní přenos s přenosem dat.

3.8 Virtuální lokální sítě (VLAN)

Princip virtuální lokální sítě Virtual LAN (VLAN) je poměrně jednoduchý. Vychází z předpokladu mikrosegmentované LAN u které jsou jednotlivá zařízení připojována přímo k přepínačům. Virtuální sít' je tvořena skupinou stanic, mezi kterými je zajištěna komunikace, mechanismus virtuální sítě zajišťuje, že data příslušející komunikaci stanic určité skupiny se nedostanou ke stanicím, které do skupiny nepatří.

Technické řešení virtuální sítě je velice jednoduché: rámce vyslané stanicí příslušející k určité skupině jsou ve vstupním přepínači označeny identifikátorem skupiny a přenášeny přepínanou sítí podobně, jako rámce neoznačené. Výstupní přepínače sítě před předáním rámce koncovému zařízení zkontrolují, zda toto zařízení přísluší ke skupině určené identifikátorem skupiny přenášeným v označeném rámci. Pokud zjistí shodu, jednoduše označení z rámce odstraní a rámec předá adresátovi. V opačném případě není rámec koncovému zařízení doručen a přepínač ho zlikviduje.



Obrázek 3.10: Virtuální lokální síť (VLAN)

Mechanismus funguje jak pro dvoubodovou, tak pro vícebodovou komunikaci a broadcast. Jediným problémem, se kterým se muselo zavedení technologie virtuálních sítí vyrovnat bylo doplnění identifikátoru VLAN do struktury rámce.

Předchůdcem v současnosti standardizované technologie virtuálních sítí, tedy technologie IEEE 802.1q, byla technologie virtuálních lokálních sítí navržená firmou Cisco pro Ethernet. Tato technologie, v případě Ethernetu označovaná jako ISL (Inter Switch Link), se opírá o nevyužívaný standard IEEE 802.10 pro kryptografickou ochranu dat v rozsáhlých lokálních a metropolitních sítích. Formát rámce Ethernetu s identifikací virtuální sítě je uveden níže.

Technologii IEEE 802.10 je vyhrazen identifikátor SAP 0x0A. Čtyřznakové pole SAID původně určené pro identifikaci kryptovaného spojení, je využíváno jako identifikátor virtuální sítě. Implementace ISL v přepínačích a směrovačích firmy Cisco omezuje počet virtuálních sítí tohoto typu na 1024. Pole SID (Source Station ID) a FF (Fragmentation Flag) jsou rezervována pro funkce správy.

Důležitým krokem v rozvoji virtuálních lokálních sítí bylo vytvoření standardů IEEE 802.1q a 802.1p. Oba využívají společný formát rámce, rozšíření rámce zahrnuje jednak identifikátor virtuální lokální sítě, jednak údaj o prioritě datového toku.

Na rozdíl od ISL doporučení IEEE 802.1q vkládá doplněné pole před původní pole L/ T. Pro identifikaci skutečnosti, že rámec je vybaven údajem podporujícím VLAN (tag) slouží šestnáctibitový identifikátor protokolu VLAN - TID (Tag Protocol Identifier) s hodnotou 0x8100.

Následující šestnáctibitové pole obsahuje tříbitový údaj o prioritě P, který dovoluje rozlišit osm úrovní priority. Přepínač podporující doporučení IEEE 802.lp pak upřednostňuje při zařazování do front portů rámce s vyšší prioritou, určitým druhům provozu (například hovorové služby, přenos videosignálu) tak lze zajistit potřebnou kvalitu provozu (doručení do časového limitu).

Virtuální lokální sítě jsou rozlišeny dvanáctibitovým identifikátorem VI, což dovoluje vytvořit na jedné fyzické LAN až 4096 LAN virtuálních.

Rozdělení stanic do virtuálních sítí se může opírat o číslo portu, fyzickou (MAC) adresu koncové stanice, případně příslušnost koncové stanice k logické podsíti internetu.

Prvá z metod je nejčastější, a pokud nevyžadujeme přiřazení jednoho portu do více lokálních sítí i nejjednodušší. Je vhodná v případech, kdy potřebujeme virtuální sítě oddělit i prostorově například při vytváření komunikačního prostředí pro více společností v jedné budově.

Rozdělení stanic do virtuálních sítí podle fyzické adresy může být proti tomu užitečné pro sítě podporující mobilní koncová zařízení. Ta se pak mohou pohybovat v dosahu celé fyzické LAN, přiřazení do příslušné VLAN se však musí opírat o tabulku.

Využití informací o síťovém protokolu a informací z hlavičky paketu dovoluje například oddělit provoz pod protokoly IP a IPX nebo vytvořit na jedné fyzické LAN více logických podsítí internetu bez nutnosti definovat umístění zařízení jednotlivých podsítí.

Rozdělení jedné fyzické LAN na více lokálních sítí virtuálních dává, vedle administrativních důvodů, šanci efektivněji využívat spoje lokální sítě. Cestou k vyšší efektivitě je podpora nezávislé funkce algoritmu Spanning Tree pro každou z lokálních sítí. Takový přístup dovolí jednak využít všechny spoje fyzické sítě, jednak výpadek jednoho konkrétního spoje nemusí znamenat dočasný výpadek komunikace ve všech virtuálních LAN.

Určitým problémem je zajištění komunikace mezi virtuálními LAN. Směrovač propojující logické podsítě postavené na VLAN by bez vestavěné podpory IEEE 802.1q musel být připojen ke dvěma nebo více portům sítě. Podpora standardu IEEE 802.1q přímo ve směrovači dovoluje pracovat přímo s rámci VLAN, směrovač je pak do fyzické sítě připojen jediným rozhraním. Moderní směrovače takovou možnost podporují, častá je i kombinace přepínače se směrovačem (Layer 3 Switching).

Kapitola 4

Síťová vrstva

Pokud bychom vyhlásily soutěž o nejdůležitější vrstvu ISO-OSI modelu, vítězem by se pravděpodobne stala třetí tedy síťová vrstva. Zjednodušeně řečeno je zodpovědná za doručení informace mezi koncovými stanicemi kdekoliv v síti (nejen v rámci segmentu, jak to zajišťuje linková vrstva).

Poznámka: V této kapitole budeme kreslit síťové diagramy, který používají notaci viz obrázek 4.1.

4.1 Funkce síťové vrstvy

Základní funkcí síťové vrstvy je doručit zprávu mezi účastníky počítačové sítě. Připomeňme si, že nižší vrstva (tedy linková) zodpovídá za doručení zprávy mezi účastníky v síťovém segmentu. Zjednodušeně lze říci, že zajišťuje komunikaci mezi „sousedy“. Síťová vrstva přidává možnost komunikovat mezi síťovými segmenty. Pro komunikaci v rámci stejného segmentu se využívá stále služeb protokolu, na kterém daný segment běží. Aby bylo možné komunikovat mezi segmenty je potřeba definovat způsob logické adresace a způsob předání dat mezi jednotlivými segmenty (směrování).

Obrázek 4.1: Notace pro síťové diagramy

4.1.1 Síťové rozhraní

Než se pustíme do funkcí síťové vrstvy, musíme definovat pojem síťové rozhraní. Jedná se v podstatě o entitu v operačním systému, která zajišťuje výměnu dat mezi operačním systémem a počítačovou sítí. Pro správnou funkčnost musí mít přiřazený logický identifikátor (IPv4 nebo IPv6 adresu, viz dále). Síťové rozhraní je typicky spojené se síťovou kartou (např. typu Ethernet nebo WiFi), ale může být i virtuální. V takovém případě zajišťuje komunikaci například s virtuální sítí pro virtuální počítače nebo třeba posílá data do síťového tunelu (VPN).

4.1.2 Logická adresace

Protokoly na síťové vrstvě obvykle zajišťují systém logických adres, které lze libovolně nebo systematicky přidělovat jednotlivým účastníkům v rámci celé sítě. V závislosti na protokolu se adresy přidělují jednotlivým účastníkům nebo i celým segmentům. Díky logické adresaci je možné doručit zprávu i účastníkům, kteří se nachazí v jiných segmentech. Aby měla logická adresace smysl, z logické adresy musí být jasné do jakého segmentu patří. Dále potřebujeme zajistit, aby měl i každý síťový segment unikátní logický identifikátor. Pokud máme toto zajištěno, můžeme začít směrovat.

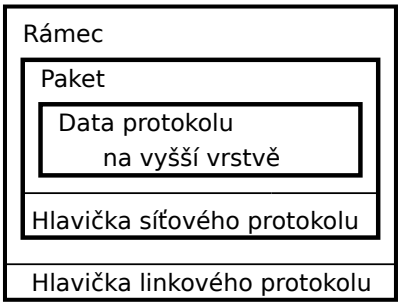
4.1.3 Směrování

Směrování je způsob hledání cesty pro doručování dat mezi zdrojovou a cílovou stanicí. Cesta se hledá na základě tzv. směrovací tabulce, která je přítomna na každém zařízení podporujícím protokol IPv4. Na hranici síťových segmentů pak stojí zařízení, kterému říkáme směrovač (router). Směrovač má typicky dvě a více síťových karet. Každá síťová karta může být připojena do jednoho segmentu (nebo více segmentů například použijeme-li techniku VLAN). Jednotlivé segmenty mohou být různého typu. Směrovač pak přeposílá zprávy dalším směrovačům nebo přímo cílovým účastníkům na základě informací ze směrovací tabulky. Samotná směrovací tabulka obsahuje informace o známých klientech a segmentech a zároveň zná, jak se k nim lze dostat. Samotný obsah tabulky a konkrétní způsob rozhodování záleží na konkrétním protokolu. Více o procesu směrování v samostatne kapitole 5.

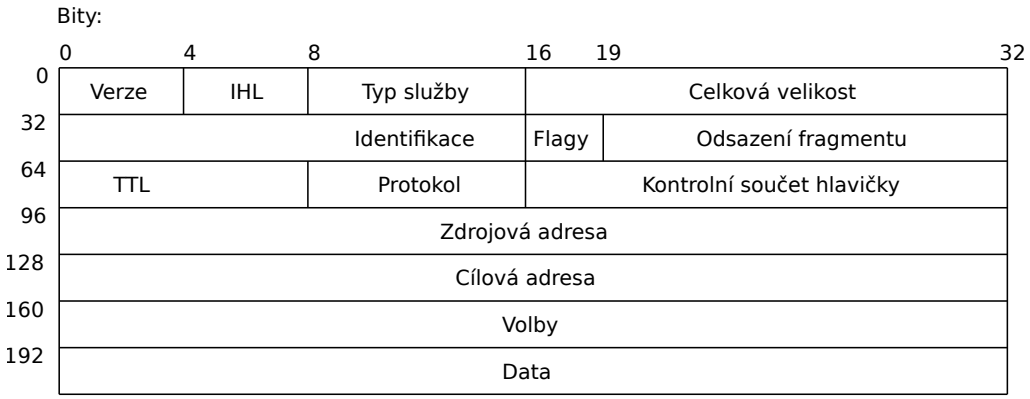
4.2 Protokol IPv4

4.2.1 Stručná historie

Návrh protokolu IPv4 byl poprvé zveřejněn v roce 1981 v RFC 791 pod názvem Internet Protocol. Poprvé byl nasazen v první komplexní počítačové síti nazývané ARPANET, která byla v podstatě přímým predchůdcem internetu. Jednalo se v podstatě o síťové propojení výzkumných pracovišť a univerzit za účelem efektivnější výměny dat. Hlavním investorem bylo ministerstvo obrany Spojených států amerických. V osumdesátých a devadesátých letech měl protokol několik konkurentů (např. protokoly SPX/IPX nebo Ap-



Obrázek 4.2: Zapouzdření na síťové vrstvě



Obrázek 4.3: Hlavička IPv4

pleTalk), ale nakonec se stal vítězem, který vládne počítačovým sítím prakticky dodnes, i když je za něj připravena náhrada v podobě protokolu IPv6. Ta se však prosazuje velmi pomalu.

4.2.2 Základní datová jednotka

Zprávám, které se posílají mezi protokoly síťové vrstvy, se říká pakety (na rozdíl do linkové vrstvy, kde mluvíme o rámcích). Každý paket obsahuje dvě části – hlavičku a data. V datové části se obvykle zapouzdřují data protokolů na vyšších vrstvách. Viz obrázek 4.2.

4.2.3 Hlavička

Q: K čemu mi znalost hlavičky bude?

A: Upřímně řečeno, znalost hlavičky není pro práci se sítí vůbec potřeba. Nicméně pokud chceme pochopit, jak protokol IPv4 funguje, náhled do hlavičky poskytne docela slušný přehled schopností, které protokol IPv4 poskytuje.

Strukturu hlavičky protokolu IPv4 vidíme na obrázku 4.3.

Návrh hlavičky je už hodně starý (1981) a čas ukázal, že některé položky jsou nepraktické a v následném protokolu Ipv6 už nebyly použity (viz kapitola 4.5.2). Dále se podíváme na jednotlivé položky a stručně si vysvětlíme jejich účel. Detailní popis jednotlivých položek by byl příliš vyčerpávající, takže se při popisu zaměříme na položky, které jsou relevatní pro základní práci s počítačovou sítí. Kompletní popis položek můžeme nalézt v původním RFC 791 [1].

Verze

Obsahuje identifákor verze protokolu. V protokolu IPv4 obsahuje samozřejmě konstantu označující protokol verze 4.

IHL (Internet Header Length)

IHL udává velikost hlavičky jako počet 32-bitových slov. Na obrázku 4.3 je vidět každé 32-bitové slovo jako jedna řádka. Z jiného pohledu můžeme říct, že IHL je vlastně adresa začátku datové části paketu. Důvodem, proč tuto hodnotu vůbec potřebujeme, je fakt, že položka *Volby* v hlavičce má proměnlivou velikost. Bez této hodnoty by nebylo možné jednoduše určit, kde začíná datová část paketu.

Typ služby

Tato položka specifikuje typ služby, jejíž data jsou přenášena v aktuálním paketu. Účelem bylo rozlišit datové toky, které potřebují nízkou latenci (např. telefonní hovor) od toků, kterým nevadí mírná prodleva (např. načítání webové stránky, kde tolik nevadí, že se načte o 300ms později). Díky tomu lze při provozu sítě upřednostňovat pakety určitých služeb.

Zajímavost: Proč některé specifikace hovoří o oktetech místo o bajtech? Z dnešního pohledu to vypadá zvláštně, protože oktet je definován jako osmice bitů a zároveň byte má také osm bitů. Podle definice bajtu se jedná o skupinu bitů typicky o velikosti osm, tedy bajt nemusí mít vždy osm bitů. V době úsvitu moderních počítačů existovalo mnoho různých vzájemně nekompatibilních architektur, které měly různou velikost základní datové jednotky, tedy bajtu. Lze vysledovat, že se velikost bajtu v podstatě sjednotila až někdy během osmdesátých let. Vágní definice bajtu však zůstala, proto se stále můžeme setkávat se slovem oktet v takovém kontextu, ve kterém potřebujeme pracovat přesně s osmi bity.

Celková délka

Tato 16-bitová hodnota udává velikost celého paketu v bytech (oktetech). Z její velikosti je jasně patrná maximální velikost paketu - 65535B, což se ovšem v praxi moc nevyužilo. Například maximální velikost rámce jednoho z nejpoužívanějších linkových protokolů Ethernet je jen 1518B, což jasně limituje maximální velikost paketu na 1500B (od 1518B odečteme velikost Ethernet hlavičky).

Položky fragmentace (Identifikace, Flagy a Odsazení fragmentu)

Následující položky jsou využívány při procesu fragmentace, jemuž jsme věnovali samostatnou kapitolu 4.4.2.

TTL (Time to live)

Tato položka odpočitává počet okamžiků jak dlouho bude datový paket naživu. Jedná se o jednoduché 8-bitové pozitivní číslo, které se při vytvoření hlavičky nastaví na určitou hodnotu. Když pak paket prochází přes směrovač, hodnota TTL je směrovačem snížena o jedničku. Pokud dojde ke snížení až na 0, je takový paket zahozen a tedy virtuálně umírá.

Protokol

Zde je specifikováno, jaký protokol se nachází ve vyšší vrstvě. Každý protokol má svůj unikátní 8-biotvé internetové číslo protokolu. Seznam známých protokolů byl udržován RFC dokumenty, ale kvůli relativně častým změnám byl seznam přemístěn do online databáze. Aktuální verzi listu lze shlédnout na stránkách organizace IANA [2].

Kontrolní součet hlavičky

Kontrolní součet pománá detekovat, zda byl paket cestou poškozen. Hodnota kontrolního součtu se počítá přes celou hlavičku a datovou část paketu. Pokud změníme data v hlavičce nebo datové části paketu, je nutné kontrolní součet spočítat znovu.

Source address, Destination Address

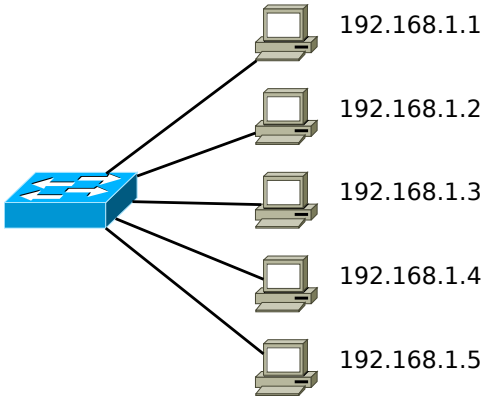
Položky obsahují zdrojovou a cílovou IP adresu. Na její podobu se podíváme podrobně v kapitole 4.2.4.

Volby

Tato část hlavičky není povinná a lze ji úplně vynechat. Velikost této položky je proměnlivá, protože záleží na obsahu. Obsahem mohou být dodatečné informace o bezpečnosti, nebo například směrování. Použití options je lehce kontroverzní téma, protože proměnná délka položky v podstatě znamená, že i délka IPv4 hlavičky je proměnlivá, což vede k náročnějšímu zpracování ze strany síťových prvků. V dnešní době se tato položka prakticky nepoužívá.

Zarovnání

Poslední položka hlavičky vlastně žádnou položkou není. Jedná se pouze o zarovnání předchozí položky tak, aby datová část paketu začínala vždy na násobku 32-bitového slova (viz. kapitola IHL).



Obrázek 4.4: Ukázka rozložení adres v jednom síťovém segmentu

4.2.4 Adresace

Protokol IPv4 používá pro identifikaci jednotlivých klientů identifikátor nazývaný IP adresa, která má délku čtyři bajty. Každý jednotlivý bajt se v IP adrese zapisuje v desítkové soustavě, přičemž jednotlivé bajty jsou odděleny vždy jednou tečkou. Následuje několik ukázek:

- 147.32.232.212
- 192.168.1.123
- 8.8.8.8

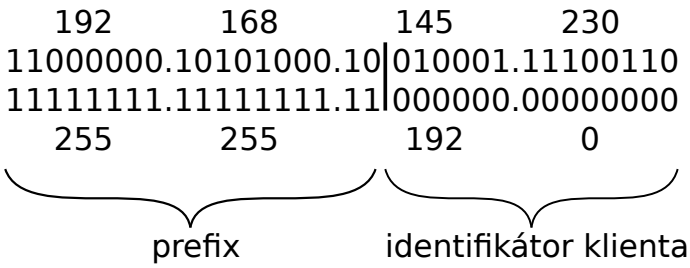
Limity IP adresy jsou zřejmé – každá číselná komponenta může nabývat hodnot 0-255 a celá IP adresa nabízí 2^{32} unikátních identifikátorů (což z dnešního pohledu rozhodně není mnoho). Aby IP adresy splnily požadavky na logickou adresaci pro síťovou vrstvu, je nutné zavést mechanismus, který umožní z adresy určit do jakého síťového segmentu patří. Tento problém se řeší rozdělením IP adresy na dvě části – adresa sítě a adresa klienta v síti. Část IP adresy s adresou sítě je pak totožná pro všechny IP adresy v rámci stejného segmentu. Situace je naznačena na obrázku 4.4. Všimněme si, že první tři komponenty IP adresy jsou shodné pro všechny koncové stanice. Poslední komponenta musí být unikátní pro kařdou stanici. Historicky vznikly dvě různé metody, které daný problém řeší – třídní a beztřídní adresace.

Třídní adresace (classfull addressing)

Tento způsob je už zastaralý a dnes se nepoužívá, proto se na něj podíváme velmi stručně. Jediný důvod proč si o něm povídáme je, že terminologie, která se zavedla s touto metodou adresace se používá dodnes, i když samotná metoda už zanikla. Třídní adresace je součástí původního návrhu protokolu IPv4 [1]. Hlavní myšlenkou třídní adresace je rozdelení adresního prostoru na několik tříd, které se mezi sebou liší poměrem délky adresy sítě vůči délce adresy klienta. Jednotlivé skupiny se pak rozlišují na základě hodnoty v bitovém prefixu. Vše je vidět v tabulce 4.2.4. Z tabulky je patrné, že se zavedly třídy A, B, C, D a E (původní návrh nepočítal s třidou E).

Tabulka 4.1: Třídní adresace

Třída	Bitový prefix	Délka prefixu / délka klientské části adresy	Celkový počet podsítí	Celkový počet adres uvnitř podsítě	Celkový počet adress pro třídu	Počáteční adresa
A	0	8/24	2^7	2^{24}	2^{31}	0.0.0.0
B	10	16/16	2^{44}	2^{16}	2^{30}	128.0.0.0
C	110	24/8	2^{21}	2^{28}	2^{29}	192.0.0.0
D (multicast)	1110	-	-	-	2^{28}	224.0.0.0
E	1111	-	-	-	2^{28}	240.0.0.0



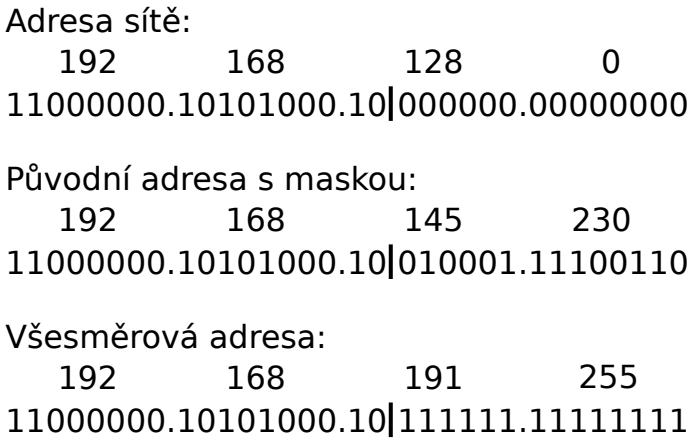
Obrázek 4.5: Ukázka vztahu mezi IPv4 adresou a její masky

Beztřídní adresace (classless addressing)

Již jsme si ukázali, jaká byla hlavní nevýhoda třídní adresace. Tuto nevýhodu řeší zavedení techniky CIDR (Classless Inter-Domain Routing [3]), která od roku 1995 zcela nahradila třídní adresaci [4]. Metoda beztřídní adresace zavádí dodatečnou informaci k IPv4 adrese: síťovou masku. Tento dodatečný atribut dokáže rozdělit IP adresu na dvě části v libovolném poměru, takže architekt sítě již není omezen třídami a může adresní prostor rozdělit výrazně efektivněji s menším plýtváním adres. Samotná maska má délku čtyři bajty a vždy se jedná o sekvenci jedniček následovanou sekvencí nul. Právě předěl mezi jedničkami a nulami určuje, kde se příslušná IP adresa dělí na prefix a unikátní identifikátor klienta. Pro příklad si vezmeme IP adresu 192.168.145.230 s maskou 255.255.192.0. Vztah této adresy a masky je naznačen na obrázku 4.5. Adresa i maska jsou zde pro větší názornost vyjádřeny v binární podobě.

Z obrázku 4.5 je patrné, že počet bitů s hodnotou 1 v masce odpovídá délce prefixu. Délka prefixu se vyjadřuje buď maskou sítě, nebo přímo uvedením délky prefixu za IP adresou. Například IP adresa z obrázku 4.5 a její maska lze vyjádřit jako:

```
192.168.145.230/18
```



Obrázek 4.6: Ukázka vztahu mezi IPv4 adresou a její masky

Délka prefixu tedy vyjadřuje stejnou informaci jako maska sítě jen zkrácenou formou. Za IP adresu se uvede jednoduše „/N“, kde N je počet binárních jedniček v odpovídající masce. Počet nul v masce nám pak říká kolik unikátních adres máme uvnitř tzv. podsítě (neboli subnetu). V příkladu na obrázku 4.5 máme celkem čtrnáct nul, takže počet unikátních adres je 2^{14}.

Z příkladu je patrné, že délka prefixu (nebo maska) jednoznačně určuje rozsah unikátních adres, které máme k dispozici. Nejnižší a nejvyšší adresa z každého rozsahu má unikátní význam. Situace je naznačena na obrázku 4.6. Ta nejnižší adresa z rozsahu nám určuje tzv. adresu sítě, která se používá jako jednoznačný identifikátor síťového segmentu (adresa na obrázku nahoře). V našem příkladu se jedná o adresu 192.168.128.0. Ta nejvyšší adresa z rozsahu se nazývá všesměrová adresa (neboli directed broadcast, viz 4.8.4) a slouží pro hromadnou komunikaci se všemi IP adresami ve stejném síťovém segmentu (adresa na obrázku dole). V našem příkladu vypadá všesměrová adresa jako 192.168.191.255. Více se dozvíme v kapitole 4.3. Adresa sítě a všesměrová adresa se obvykle nepřidělují síťovým rozhraním.

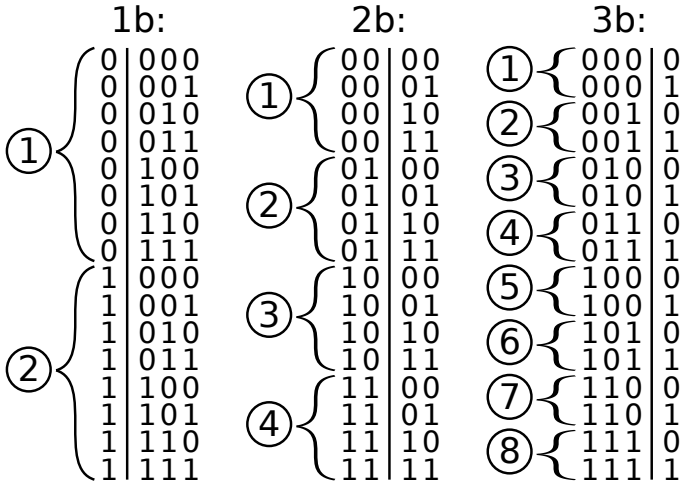
Zajímavost: V síťařském slangu se dodnes používá pro některé adresy označení "áčková", "béčková"a "céčková". Název je odvozen z původních adresních tříd. Například specifikace třídy C víceméně odpovídá IP adresám s maskou 255.255.255.0, proto se adresy s touto maskou označují jako "céčkové". Podobná analogie platí i pro ostatní skupiny - třída A odpovídá masce 255.0.0.0 a třída B masce 255.255.0.0.

4.3 Dělení adresního prostoru v IPv4

Z pohledu síťové vrstvy lze říci, že počítačová síť se skládá z jednotlivých síťových segmentů (linek), které jsou spojeny směrovači. Pro funkční komunikaci musíme zajistit, aby každé síťové rozhraní připojené do segmentu, mělo unikátní adresu. Zároveň potřebujeme, aby měl každý segment svou jednoznačnou identifikaci. Adresní prostor, který nabízí protokol IPv4, obsahuje 2^{32} unikátních adres.

Počet bitů v masce	Počet podsítí	Počet adres v podsíti
1	2	8
2	4	4
3	8	2

Tabulka 4.2: Poměry počtu podsítí a jejich velikostí pro adresní prostor o velikosti 4 bity

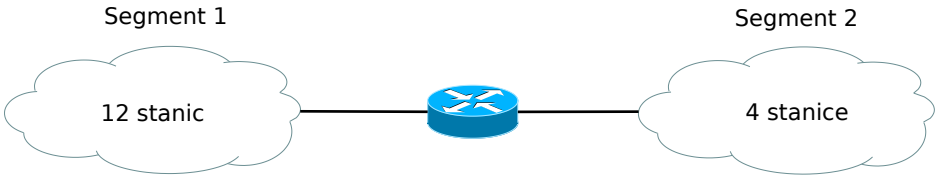


Obrázek 4.7: Ukázka dělení adresního prostoru s velikostí 2^4 podle masky s velikostí 1, 2 a 3 bity

4.3.1 Jak to funguje?

Pro snadnější demonstraci začneme s trochu menším adresním prostorem. Předpokládejme, že máme k dispozici adresní prostor o velikosti 2^4, který nabízí přesně 16 unikátních adres od 0 do 15. Jaké máme možnosti, pokud ho chceme rozdělit mezi dva síťové segmenty? Obzvláště, pokud chceme z adresy určit do jakého segmentu patří. Pokud použijeme notaci zavedenou ve specifikaci protokolu IPv4, tak k adresám můžeme přiřadit masku, která jasně určí, která část adresy určuje příslušnost k určitému síťovému segmentu. V případě dvou segmentů, nám k jejich rozlišení stačí pouze jeden bit. Zvolme tedy masku o velikosti jeden bit. Situaci si můžeme prohlédnout na obrázku 4.7 vlevo.

Z obrázku je patrné, že jedno-bitová maska nám rozdělí adresy do dvou skupin, které se liší v hodnotě prvního bitu. Tyto skupiny adres nazýváme podsítě (neboli subnets), protože vznikly rozdělení větší sítě. Pokud masku rozšíříme na dva bity, můžeme adresní prostor rozdělit až na čtyři podsítě (2^4) a když přidáme masce ještě jeden bit, lze získat až osm (2^3) podsítí. Každá podsíť je jednoznačně identifikována unikátní kombinací jednoho, dvou nebo tří bitů. Závislost mezi počtem jednotlivých skupin a jejich velikostí pro adresní prostor o velikosti čtyři bity je vidět v tabulce 4.3.1.



Obrázek 4.8: Ukázka nemožného rozdělení 4-bitového adresního prostoru

4.3.2 Limity při dělení adresního prostoru

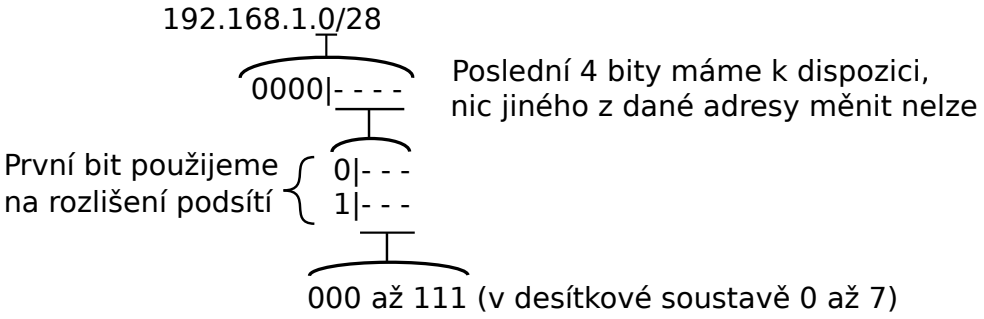
Zůstaňme u 4-bitového adresního prostoru a podívejme se na obrázek 4.8. Vidíme zde zjednodušené schéma počítačové sítě, která skládá ze dvou segmentů s uvedeným počtem potřebných adres - 12 a 4. Na rozhraní segmentů se pochopitelně nachází směrovač. Na první pohled vypadá vše v pořádku, protože máme k dispozici celých 16 adres, což je přesně tolik, kolik je potřeba. Naším cílem je rozdělit tyto adresy do dvou skupin za dodržení definovaných pravidel. Abychom mohli vytvořit dvě skupiny adres, musíme zavést masku o velikosti alespoň jeden bit. Podle obrázku 4.7 a tabulky 4.3.1 je jasné, že maximální počet adres pro skupinu je 8. Z toho vyplývá, že adresní prostor o velikosti 4 bity je nedostatečný, i když počet adres je teoreticky dostatečný. Adresy v tomto případě nelze rozdělit tak, aby byla každá skupina jednoznačně identifikovatelná pomocí unikátního prefixu.

Dalším limitem metody dělení adresního prostoru za pomoci masky je, že každý rozsah může začínat pouze na určitých hodnotách. Problém je dobře patrný z obrázku 4.7. Pokud máme k dispozici tři bity (1-bitová maska), tak daný rozsah může začínat pouze na adresách, které mají v posledních třech bitech samé nuly. Obdobnou situaci můžeme pozorovat pro 2-bitovou masku, kde každý rozsah začíná vždy na adrese, kde jsou poslední dva bity nulové.

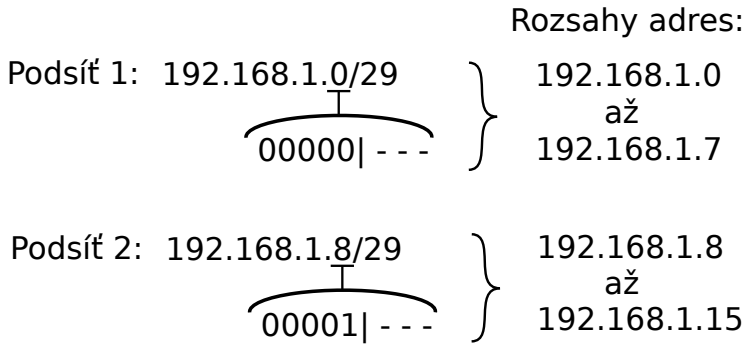
4.3.3 Jak to funguje v IPv4?

Nyní zkusíme stejnou metodu aplikovat na reálnějším příkladu. Předpokládejme, že jsme dostali přidělený adresní rozsah 192.168.1.0/28 a my jej máme rozdělit mezi dva síťové segmenty, tedy na dvě podsítě. Situace je naznačena na obrázku 4.9, kde jsou důležité části IP adresy pro větší názornost převedeny do binární podoby. Vhledem k tomu, že IP adresa má 32 bitů, tak máme k dispozici poslední čtyři bity pro naše potřeby. Pro snadnější pochopení je lepší představit si IP adresu jako sekvenci 32 jedniček a nul. Rozdělení provedeme stejně jako v předchozí sekci - jeden bit (konkrétně ten nejvíce vlevo) z našeho rozsahu použijeme na identifikaci jednotlivých podsítí. Poslední tři bity pak určují unikátní adresy v rámci jednotlivých podsítích.

Nyní můžeme sestavit IP adresy jednotlivých podsítí tak, že zkombinujeme přidělený rozsah a s posledními čtyřmi bity, které jsme navrhli (viz obrázek 4.10) Pro první podsíť nám vyjde rozsah 192.168.1.0 - 192.168.1.7 a pro druhý 192.168.1.8 - 192.168.1.15. Protože první bit z naší čtveřice slouží pro odlišení jednotlivých podsítí (ne pro rozlišení jednolivých adres v rámci segmentu neboli uvnitř jedné podsítě), tak jej připojíme již k existujícímu prefixu. Segmentům pak přidělíme následující adresy sítě: 192.168.1.0/29 a



Obrázek 4.9: Ukázka rozdělení 4-bitového adresního prostoru na dvě podsítě



Obrázek 4.10: Ukázka rozsahů adres po rozdělení na dvě podsítě

192.168.1.8/29.

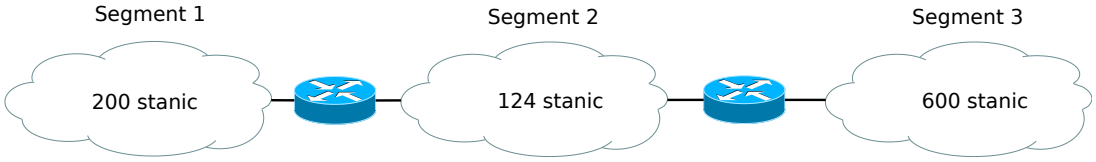
Pro správné dělení adresního prostoru v protokolu IPv4, musí platit následující pravidla:

P-1 Všechny IP adresy ve stejné podsíti mají stejnou adresu sítě, neboli stejný prefix.

P-2 Adresa sítě je pro každý síťový segment unikátní.

P-3 Žádná adresa sítě není obsažena v jiné adrese sítě.

Na aplikaci těchto pravidel se podíváme v následujícím příkladu. Na obrázku 4.11 vídíme jednoduchou lokální síť skládající se ze tří segmentů a dvou směrovačů. Dále máme přidělený adresní rozsah 192.168.0.0/21, který potřebujeme rozdělit mezi naše tři segmenty, tedy potřebujeme vytvořit tři podsítě. Síťová maska nám jasně udává kolik adres máme vlastně k dispozici - 2^11 v rozsahu od 192.168.0.0 do 192.168.7.255. Prvních 21 bitů adresy jsou prefixem celé naší lokální sítě a protože nám byl prefix přidělen, nebudeme do něj zasahovat.



Obrázek 4.11: Izolovaná lokální síť se třemi segmenty s různým počtem zařízení

Dělení adresního prostoru lze rozdělit do dvou kroků:

- Určení délky síťového prefixu (neboli určení velikosti masky)
- Určení adresy sítě pro každou podsíť

V našem příkladu z obrázku 4.11 máme pro každý segment uveden počet koncových zařízení. V prvním kroku tedy určíme délky prefixů pro každou podsíť tak, že najdeme první mocninu dvojky, která je vetší nebo rovno potřebnému počtu adres pro daný segment:

- **Segment 1:** $200 + 1^* + 2 <= 2^8$
- **Segment 2:** $124 + 2^* + 2 <= 2^7$
- **Segment 3:** $600 + 1^* + 2 <= 2^{10}$

Poznámka 1: Hodnoty s hvězdičkou jsou síťová rozhraní směrovačů, které potřebují mít také přidělenu IP adresu.

Poznámka 2: Do každého výpočtu započítáváme konstantně hodnotu 2, která zahrnuje dvě rezervované adresy - adresu sítě a všesměr.

Poznámka 3: Segmentu 2 bychom v praxi přidělili větší rozsah, protože takto už není možné přidat do něj nové síťové rozhraní.

Mocnina dvojky pak jasně udává délku prefixu, kterou spočítáme tak, že exponent z mocniny odečteme od celkové ho počtu bitů v IP adrese:

- **Segment 1:** $32 - 8 = 24$
- **Segment 2:** $32 - 7 = 25$
- **Segment 3:** $32 - 10 = 22$

V dalším kroku přidělíme adresy sítě jednotlivým segmentům. Vzhledem k tomu, že tento krok má několik možných řešení, uvedeme na ukázku dvě vybraná řešení, které pomohou pochopit problematiku do větší hloubky.

Segment	Počet adres	Prefix	Všesměr
3	601	192.168.0.0/22	192.168.3.255
1	201	192.168.4.0/24	192.168.4.255
2	126	192.168.5.0/25	192.168.5.255

Tabulka 4.3: Řešení 1 - od největšího segmentu po nejmenší

Řešení 1 - od největšího segmentu po nejmenší

První způsob řešení je nejjednodušší a nevyžaduje hlubší pochopení problematiky. Adresy budeme přidělovat segmentům od největšího po nejmenší.

Pro největší segment 3 zvolíme jednoduše první dostupnou adresu sítě a nastavíme délku prefixu, kterou jsme spočítali v předchozím kroku: 192.168.0.0/22. Maska je tedy 255.255.252.0. Všesměrová adresa je 192.168.3.255 a je zároveň nejvyšší adresou z celého rozsahu.

Nyní potřebujeme určit adresu sítě pro druhý nejvetší segment, tedy segment 1. V tento okamžik se projeví výhoda, že jsou segmenty seřazené podle velikosti, protože můžeme přiřadit hned první následující adresu, která je k dispozici: 192.168.4.0/24. Všesměrová adresa bude pak 192.168.4.255.

Poslednímu segmentu přiřadíme opět následující volnou adresu: 192.168.5.0/25. Všesměrová adresa bude v tomto případě 192.168.5.127.

Celkové řešení je v tabulce 4.3.

Řešení 2 - od nejmenšího segmentu po největší

Předchozí řešení se hodí především v případech, kdy adresní prostor dělíme při tvorbě návrhu sítě. V praxi je však někdy nutné pracovat s adresním prostorem, kde jsou již nějaké rozsahy přiděleny a další dělení vyžaduje hlubší porozumnění. V tomto a v následujícím příkladu ukážeme řešení některých specifických situací.

Rozdělme tedy adresní prostor přesně podle opačného pořadí oproti předchozímu řešení. Tímto si uměle nasimulujeme situaci, kdy není jednoduše možné si sítě na začátku seřadit. Začneme se segmentem 1, se kterým je nejméně komplikací. Přidělíme mu adresu přímo ze začátku našeho rozsahu: 192.168.0.0/25. Všesměrová adresa bude pochopitelně 192.168.0.127.

Přidělení adresy segmentu 2 bude již trochu komplikovanější. Intuitivně lze segmentu přiřadit hned další volnou adresu, která je na řade, jako tomu bylo v předchozím řešení: 192:168.1.128/24. Problémem však je, že tato adresa není adresou sítě podle definice, protože se nejedná o první adresu z rozsahu s daným prefixem. V adrese za prefixem by měly být jen binární nuly. Musíme tedy najít adresu sítě jiným způsobem. Na správné určení adresy sítě existuje několik pomůcek. My budeme vycházet z matematické podstaty masky, jež nám prostor dělí podle pravidel binárních čísel. Pro nalezení adresy sítě si stačí uvědomit, že vždy musí začínat na násobku své vlastní velikosti. Pokud tedy povazujeme

Segment	Počet adres	Prefix	Všesměr
2	126	192.168.0.0/25	192.168.0.127
1	201	192.168.1.0/24	192.168.1.255
3	601	192.168.4.0/22	192.168.7.255

Tabulka 4.4: Řešení 2 - od nejmenšího segmentu po největší

adresu 192.168.0.0 za počátek rozsahu, adresy sítě pro segment o velikosti 2^8 musí začínat na násobcích čísla 256. V případě IP adres jsou násobky následující:

- Násobek 0: 192.168.0.0 + 0 * 256 = 192.168.0.0
- Násobek 1: 192.168.0.0 + 1 * 256 = 192.168.1.0
- Násobek 2: 192.168.0.0 + 2 * 256 = 192.168.2.0
- ...

První násobek 192.168.0.0 je již obsazená adresa, tu použít nemůžeme. Druhý násobek je však volný a můžeme ho použít: 192.168.1.0/24. Všesměrová adresa bude pak 192.168.1.255.

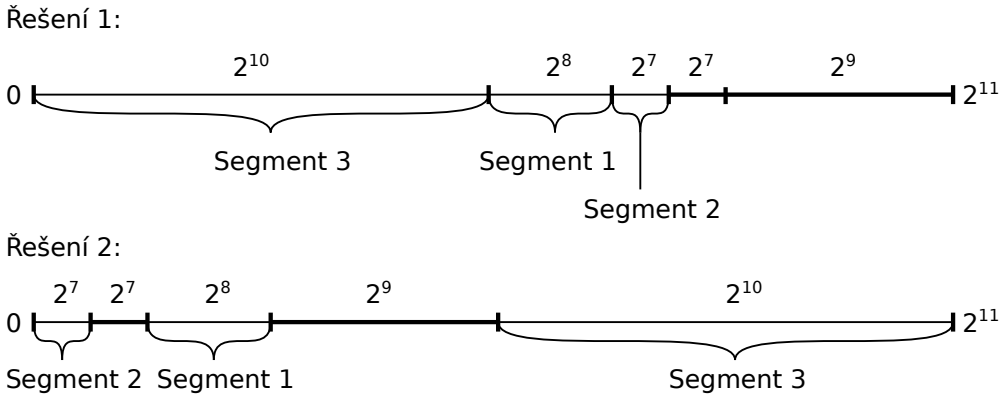
V případě segmentu 3 bude situace jestě poněkud komplikovanější. Pravidlo, které jsme použili pro segment 1 lze použít i pro segment 3, ale nalezeni nasobku velikosti segmentu není tak intuitivní. Problémem je maska o velikosti 22 bitů, která nám rozděluje adresu uvnitř třetího bajtu. Znamená to, že potřebujeme hledat násobky čísla 2^{10}. Nejjednodušší způsob jak je najít, je převést celou adresu do binární podoby a považovat ji za jednou dlouhé 32-bitové číslo. Násobky čísla 2^{10} je pak jakékoliv binární číslo, které končí na přesně na deset nul (což koresponduje s exponentem dvojkové mocniny). Velikost segmentu 3 v binární podobě je 10000000000_b a násobky pak vypadají následovně:

- Násobek 0: 0 * 10000000000_b = 0 -> 192.168.0.0
- Násobek 1: 1 * 10000000000_b = 100.00000000 -> 192.168.4.0
- Násobek 2: 2 * 10000000000_b = 1000.00000000 -> 192.168.8.0
- Násobek 3: 3 * 10000000000_b = 1100.00000000 -> 192.168.12.0
- ...

Pro naše řešení vybereme druhý násobek, protože adresa, která odpovídá prvnímu násobku, je už obsazena. Adresa sítě pro segment 3 je tedy: 192.168.4.0/22. Všesměrová adresa je 192.168.7.255. Pro přehlednost uvádíme celé řešení v tabulce 4.4.

Několik úvah k dělení adresního prostoru

Obě navržená řešení nabízí jiný přístup k dělení adresního prostoru, přičemž to první se intuitivně jeví jako "lepší". Důvodem je, že neobsahuje viditelné mezery a proto vypadá, že hospodaří s adresním prostorem lépe. V této kapitole bychom rádi ukázali, že tomu



Obrázek 4.12: Vizualizace dělení adresního prostoru pro řešení 1 a 2

tak ve skutečnosti vůbec není. Nejdříve se podívejme na obrázek 4.12, kde máme graficky znázorněno využití adresního prostoru v grafické podobě. Je jasně vidět, že po rozdělení nám zbyly dva adresní bloky o velikosti 7 a 9 bitů v obou případech. Pořadí přidělených adresních bloků nemělo v tomto případě žádný vliv.

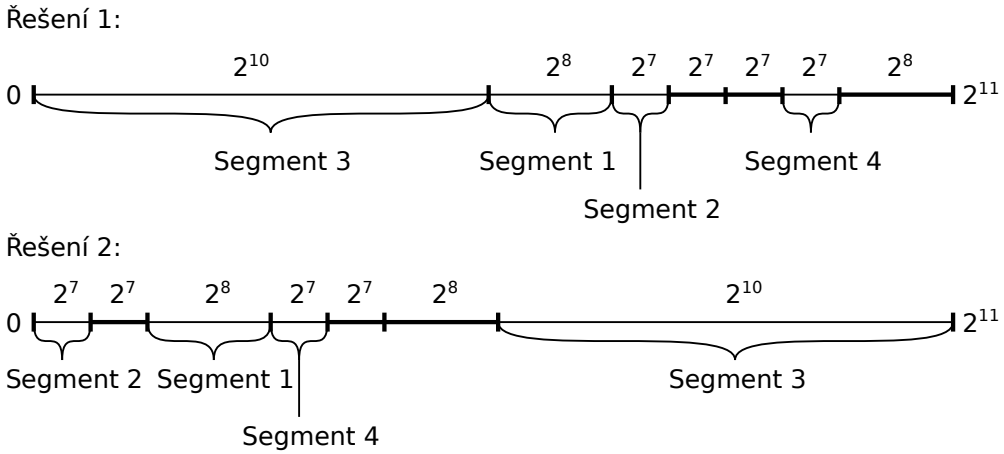
Nyní si představme, že z původního rozsahu 192.168.0.0/21 potřebujeme přidělit další blok o velikosti 7 bitů pro nově připojený segment 4. V obou případech můžeme použít volný blok o stejné velikosti a stejně tak nám v obou případech zbyde jeden volný blok o velikosti 9 bitů. Nový blok bychom ovšem mohli přidělit i tak, že bychom jej odebrali odněkud z bloku o velikosti 9 bitů. Ukázku jedné z variant pro řešení 1 a 2 můžeme vidět na obrázku 4.13. V obou případech nám zbyly shodně tři bloky o velikostech 7, 7 a 8 bitů. Je zřejmé, že počáteční seřazení bloků podle velikosti nám nepřineslo žádnou výhodu, kromě hezkého uspořádání na začátku. Nejlepším způsobem, jak nejlépe využít adresní prostor je, přidělit každému novému segmentu adresy z nejmenšího možného bloku. Jedině tak nebude docházet ke zbytečnému rozdělování větších bloků.

Na obrázku 4.13 si můžeme všimnout ještě jedné zajímavosti. U řešení 1 máme u vedle sebe dva volné bloky o velikosti 7 bitů. Na první pohled by se mohlo zdát, že je lze sloučit do jednoho volného bloku o velikosti 8 bitů. To však není možné, protože dva menší bloky s velikostí 7 bitů nevznikly rozdělením jednoho bloku s velikostí 8 bitů. Adresy v obou 7-bitových blocích mají různé prefixy a nedají se vyjádřit společným o jeden bit menším prefixem.

Speciální adresní rozsahy a adresy

Z celého adresního prostoru bylo vyčleněno několik speciálních rozsahů pro různé účely. Rozsahy uvedené v této kapitole mají pouze lokální rozsah a nelze je použít pro komunkaci v Internetu.

- **Privátní adresy** [5] - tři rozsahy, kde každý z nich byl vyčleněn z jedné z původních adresních tříd A, B a C. Tyto rozsahy jsou určeny pouze pro lokální sítě a nesmí použít např. v rámci internetu. Privátní adresy mají platnost pouze v rámci lokální



Obrázek 4.13: Vizualizace dodatečného zaplňování adresního prostoru pro řešení 1 a 2

sítě a tudíž se mohou napříč lokálními sítěmi opakovat. Komunikace s ostatními sítěmi je pak zajištěna pomocí technologie NAT (viz kapitola 4.4.4).

  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16

- **Link-local** - adresy pro lokální konfiguraci bez přístupu k internetu. Mohou se přiřazovat automaticky (viz kapitola 4.4.6).

  - 169.254.0.0/16

- **Loop-Back** - rozsah pro komunikaci v rámci jednoho zařízení. V podstatě se jedná o komunikaci sám se sebou. V operačních systémech se většinou nalézá virtuální síťové rozhraní, které vrací všechnu komunikaci zpět a obvykle se mu přiděluje adresa z tohoto rozsahu. Toto rozhraní většinou slouží pro komunikaci interních systémových procesů nebo pro lokální testování síťové komunikace mezi programy.

  - 127.0.0.0/8

- **Multicast** - blok adres, které se používají v lokálních sítích pro komunikaci se skupinou adres.

  - 224.0.0.0/4

- **Shared Address Space** - speciální rozsah privátních adres, které jsou speciálně určeny pro poskytovatele internetu, aby je mohli přidělovat svým klientům za NATem (viz kapitola 4.4.4).

  - 100.64.0.0/10

- **This host on this network** - nedefinovaná adresa, používá se např. pro speciální záznamy ve směrovacích tabulkách, nebo v situacích, kdy IP adresa musí být vyplněna, ale není známa.

  - 0.0.0.0/8

- **Limited Broadcast** - limitovaná všesměrová adresa pro komunikaci uvnitř segmentu bez znalosti síťového prefixu pro daný segment. Pro srovnání s klasickou všesměrovou adresu viz 4.8.4.

  - 255.255.255.255/32

Úplný výčet speciálních rozsahů lze nalézt v [6].

Veřejné IP adresy

Za veřejné adresy se považují všechny adresy, které nemají přiřazený nějaký speciální účel (viz předchozí kapitola 4.3.3). Veřejné adresy se používají pro komunikaci na Internetu a každý poskytovatel má přidělenou určitou část těchto adresních rozsahů. Veřejné rozsahy jsou spravovány a přidělovány organizací IANA [7].

4.4 Doplňkové služby a protokoly k protokolu IPv4

4.4.1 ICMP protokol

Internet Control Message Protocol slouží pro zákadní diagnostiku počítačové sítě. Jedná se o nezbytný doplněk protokolu IPv4. Protokol definuje několik základních zpráv, které zajišťují přenos informace o chybách nebo problémech v počítačové síti. Každá zpráva mimo jiné důležité položky - typ a kód zprávy. Typ rozlišuje jednotlivé druhy zpráv (např. *cíl nedostupný*, nebo *čas vypršel*) a kód pak upřesňuje informaci ve zprávě (pro typ zprávy *cíl nedostupný* upřesňuje co přesně bylo nedostupné, např. síť, port). V této sekci se podíváme na několik vybraných typů zpráv, se kterými se při práci s počítačovou sítí setkáváme nejčastěji. Více detailů najdeme v příslušném souboru RFC a jeho updatech [8].

Echo a Echo Reply

Tyto dvě zprávy jsou využívány utilitou "ping", která odešle danému cíli zprávu Echo (podle některých zdrojů se nazývá Echo Request) a očekává, že dostane zpět zprávu Echo Reply. Tento mechanismus zjistí dvě informace:

- Cíl je připojen k síti a odpovídá. (Pozn.: drtivá většina zařízení na zprávu Echo ve výchozím stavu odpovídá, nicméně toto chování lze vypnout. Například rané verze operačního systému Windows 10 měly tak agresivní nastavení firewallu, že na Echo zprávy neodpovídaly.)
- Jak dlouho trvá cesta tam a zpět, neboli odezva.

Ukázku funkčního příkazu ping v GNU implementaci si můžeme prohlédnout na následujícím výpisu:

```
$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=2.54 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=2.63 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=2.56 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=2.51 ms
^C
--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 2.508/2.560/2.632/0.045 ms
```

Poznámka: ^C je znaková reprezentace stisku kombinace kláves Ctrl+C, která poslala signál SIGTERM a následně přerušila provádění příkazu.

Z výpisu vidíme, že došlo k odeslání čtyř zpráv a že jsme dostali čtyři odpovědi. To znamená, že IP adresa 192.168.1.1 je aktivní. Na každém ze čtyř řádků vidíme velikost dat Echo zprávy. Dále vidíme sekvenční číslo zpráv (icmp_seq) a výchozí hodnotu TTL. Poslední položka na řádku (time) udává, za jak dlouho dorazila odpověď. Po té, co je příkaz ukončen, tak vypíše statistiku. Více detailů lze najít v manuálové stránce příkazu ping.

Zajímavost: Název utility ping není odvozen z mezinárodního názvu hry stolního tenisu (ping-pongu), ale podle autora utility byl odvozen od akustického sonaru, který odeslílá akustické signály do okolí (foneticky se zvuk sonaru označuje jako "ping") a pak zaznamenává odrazy od případných překážek. [9]

Destination Unreachable

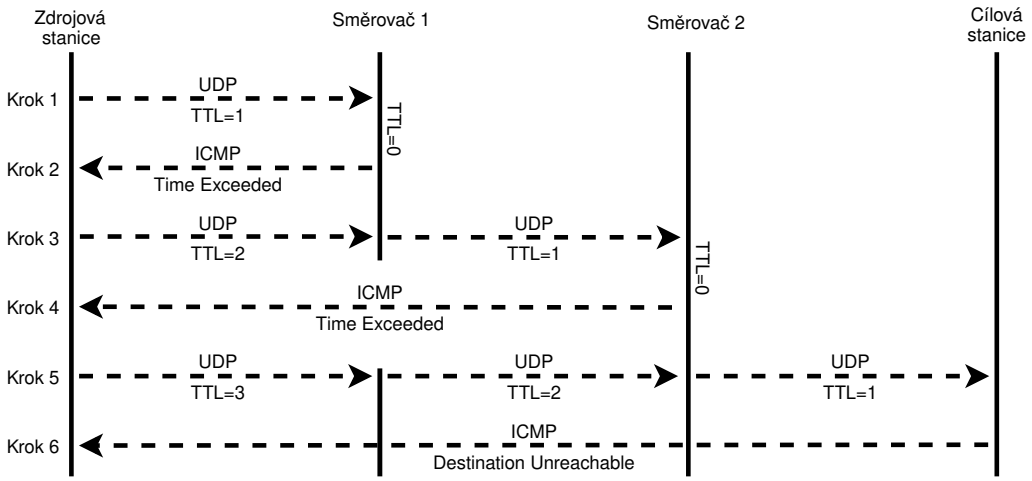
Tuto zprávu může (ale nemusí) posílat směrovač, který zjistí, že požadovaný cíl komunikace není dostupný. Zpráva se posíla zpět odesílateli. Součástí zprávy je detailnější infomace o tom, co nebylo dostupné, nebo co konkétně selhalo:

- Síť je nedostupná
- Cíl je nedostupný
- Protokol je nedostupný
- Port je nedostupný
- Fragmentace nutná a DF bit je nastaven (viz 4.4.2)

Pro úplný výpis položek odkazujeme čtenáře na RFC [8].

Time Exceeded

Tuto zprávu posílají zpravidla směrovače, které průchozímu paketu sníží hodnotu TTL na 0. V takovém případě je paket zahozen a na zdrojovou IP adresu z IP hlavičky zahozeného paketu se pošle zpráva Time Exceeded. Odeslání této zprávy opět není povinné.



Obrázek 4.14: Jak funguje utilita traceroute

Takové chování se využívá například v utilitě "traceroute". Tato utilita umožňuje zjistit kudy vede cesta k cílové IP adrese. Jinými slovy vypíše IP adresy všech směrovačů, které se nachází po cestě k cílové IP adrese. Na obrázku 4.14 máme naznačen princip, na jakém utilita traceroute pracuje. Zdroj posílá zprávu na cílovou adresu, ale schválně nastavuje příliš malé TTL, aby si vynutil od směrovačů odeslání zprávy Time Exceeded. Začne na TTL o hodnotě jedna a pak ji v dalších krocích po jedničkách zvyšuje, dokud nedostane odpověď přímo od cilové IP adresy. (Pozn.: zprávy obsahují standadní IP hlavičku na síťové vrstvě a na transportní vstvě mají hlavičku protokolu UDP s cílovým portem zpravidla 33434.

Na následujícím výpisu máme ukázku výstupu GNU implementace utility traceroute:

```
$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
1  194.212.194.26 (194.212.194.26)  49.937 ms  49.863 ms  51.324 ms
2  * * *
3  89-24-86-5.customers.tmcz.cz (89.24.86.5)  53.256 ms  53.243 ms  57.661 ms
4  * * *
5  dns.google (8.8.8.8)  71.097 ms  71.081 ms  71.065 ms
```

Z výpisu je patrné, že zpráva projde přes čtyři směrovače (řádky očíslované 1 až 4) než dorazí na cílovou IP adresu (řádek s číslem 5). Na řádku je většinou uvedeno doménové jméno (pokud je známo, více o doménách v kapitole **??**) a IP adresa. Dále vidíme tři časové hodnoty v *ms*, které říkají jaká je odezva (tato implementace odesílá pro každou hodnotu TTL tři zprávy a proto vidíme všechny tři naměřené hodnoty). Na řádcích s čísly 2 a 4 vidíme pouze tři hvězdičky, což znamená, že dva směrovače po cestě k cíli neodesilají zprávu Time Exceeded.

Další funkce ICMP protokolu

Výčet všech funkcí ICMP by byl poněkud vyčerpávající. Kompletní informace lze dohledat v [8].

4.4.2 Fragmentace

Pokud posíláme paket na IPv4 adresu mimo aktuální segment, tak komunikace musí projít přes jeden a více směrovačů. Každý směrovač stojí na rozhraní dvou a více segmentů. Cesta od zdroje k cíli pak vede přes dva a více segmentů, což může představovato problém. Každý segment může používat jiný linkový protokol a jednotlivé protokoly se mohou lišit v použité hodnotě MTU (Maximal Transmission Unit - maximální velikost dat, které může přenášet rámec linkové vrstvy). Když tedy odešleme paket s MTU aktuálního segmentu, některý ze segmentů po cestě k cíli může mít MTU menší. To představuje problém pro směrovač před tímto segmentem, který se snaží vytvořit nový rámec pro následující segment, ale data z příchozího rámce se do nového nevejdou. Směrovač má pak pouze dvě možnosti většinou v závislosti zda je fragmentace povolena v IP hlavičce - buď příchozí paket rozdělí (fragmentuje) na menší a pošle je dál nebo paket zahodí a informuje odesílatele. Podívejme se podrobněji na obě situace. [1]

Fragmentace povolena

V popisu IPv4 hlavičky jsme ještě nevysvětlili tři položky, které se týkají fragmentace - Identification, Flags a Fragment Offset. Položka Identification je dlouhá 16 bitů a obsahuje identifikátor, který vyplňuje odesílatel paketu a pomáhá v opětovném zkompletování fragmentovaného paketu. V položce Flags (tři bity) mohou být následující hodnoty:

- Bit 0: rezervován, musí být 0
- Bit 1: (DF) 0 = paket může být fragmentován, 1 = paket se nemá fragmentovat
- Bit 2: (MF) 0 = poslední fragment, 1 = více fragmentů

Položka Fragment Offset má 13 bitů a udává počet 64-bitových jednotek od začátku datové části původního paketu.

Fragmentaci provádí směrovače po cestě k cíli pouze pokud je fragmentace povolena, tedy když je DF (don't fragment) bit nastaven na 0. Odesílatel paketu nastaví tedy DF na 1 a zároveň zvolí Identification tak, aby hodnota byla unikátní pro danou kombinaci IP adres odesílatele a příjemce a použitého protokolu. Unikátnost hodnoty je myšlena tak, aby byla unikátní alespoň po dobu, dokud je paket doručen do cíle a jeho identifikátor není zaměnitelný za jinou hodnotu z podobné komunikace. Identifikátor pak pomáhá zkompletovat fragmentované pakety na straně příjemce. Odesílatel dále nastaví MF bit a Fragment Offset na nuly.

Paket s tímto nastavením pak putuje přes jednotlivé směrovače a dokud nenarazí na směrovač před segmentem jehož MTU je menší než velikost paketu v bajtech dohromady s velikostí hlavičky linkového protokolu pro následující segment. V takovém případě se paket jednoduše nevejde do rámce pro segment s menším MTU a musí dojít k fragmentaci.

Datová část původního paketu se pak rozdělí na potřebné množství fragmentů. Každý fragment dostane novou IP hlavičku, ve které se nastaví nově velikost paketu (Total Length) podle aktualní velikosti fragmentu. Dále se nastaví příznak MF ve Flags položce na 1, tedy na příznak více fragmentů a nakonec se nastaví Fragment Offset na příslušnou hodnotu podle toho, na kolik 64-bitových jednotek je aktuální fragment vzdálen od začátku dat. Mezi daty, které se fragmentují jsou i všechny hlavičky ve vyšších vrstvách.

Data se pak defragmentují až u příjemce, ne na síťových prvcích po cestě. K defragmentaci pak stačí přijmout všechny fragmenty i s IP hlavičkami. Podle Identification položky se určí, které fragmenty patří k sobě. Příznak MF položce Flags udává, zda se jedna o poslední fragment (bit je roven 0). Položka Fragment Offset pak jednoduše určuje pořadí jednotlivých fragmentů.

Fragmentace není povolena

Proces fragmentace v praxi způsobuje mnoho problémů:

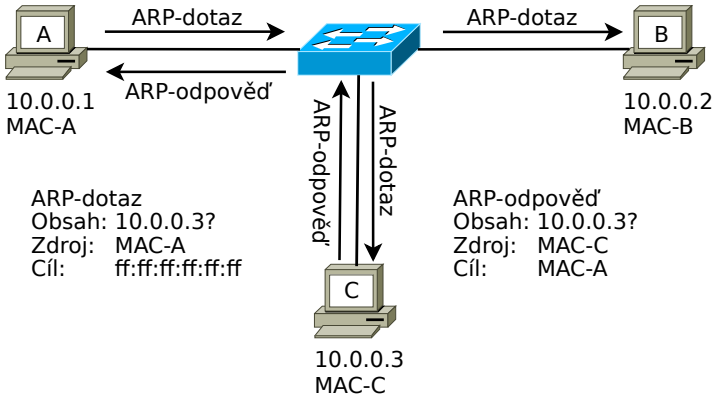
- Zvyšuje výpočetní zátěž na směrovačích, které fragmentaci provádí.
- Komplikuje analýzu síťového provozu, protože hlavičky a data vyšších vrstev nejsou přítomna ve všech fragmentech.
- Na síťových prvcích komplikuje funkce, které vyžadují přístup k datům vyšších vrstev (např. NAT nebo QoS)

Pro uvedené důvody se snažíme fragmentaci spíše zabránit. Toho docílíme jednoduše tak, že v odesílaném paketu nastavíme v hlavičce v položce Flags příznak DF na 1 (Don't fragment). Takový paket směrovače nesmí fragmentovat a pokud se paket nevejde do MTU, je zahozen a odesílateli se pošle ICMP zpráva "Fragmentace nutná a DF nastaven". Součástí této zprávy je informace o velikosti MTU v následující lince (segmentu) a takže odesílatel ví, jak velký paket může odeslat, aby takovým segmentem prošel.

Mechanismus popsany předchozím odstavci využívá služba pro oběvování MTU pro celou cestu k cíli (Path MTU Discovery [10]).

4.4.3 Protokol ARP

Adress Resolution Protocol tedy ARP [11] je úzce spojen s linkovým protokolem Ethernet. V podstatě implementuje jednoduchý mechanismus, který umožňuje zjistit příslušné MAC adresy [cite Linkova vrstva] k síťovým adresám (nejčastěji protokolu IPv4, ale podporuje i jiné síťové protokoly). Protokol se použije typicky až v okamžiku, kdy je potřeba zjistit MAC adresu pro cílovou adresu tak, aby bylo možno vyplnit hlavičku protokolu Ethernet. Při práci s ARP si operační systémy udržují překladovou tabulku, která obsahuje v každém záznamu vazbu mezi IP a příslušnou MAC adresou. Pro potřeby tohoto textu ji nazývejme ARP tabulkou.



Obrázek 4.15: Ukázka protokolu ARP

ARP dotaz a odpověď

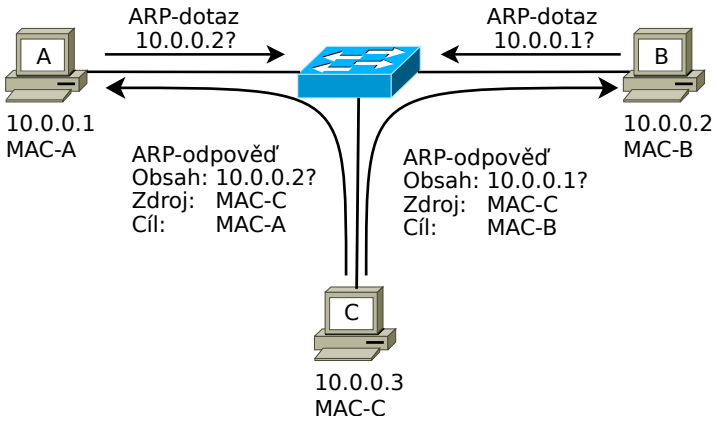
Primárně ARP slouží pro zjištění MAC adresy k dané IPv4 adrese. Operační systém čerpá informace o MAC adresách z ARP tabulky. Pokud není příslušný záznam nalezen, nebo záznam expiroval, musí operační systém záznam do tabulky doplnit právě odesláním ARP dotazu a čeká na ARP odpověď. Ukázkovou situaci máme na obrázku 4.15. Máme zde tři stanice A, B a C, každá s přidělenou IP adresou a vlastní MAC adresou. V naší ukázce potřebuje stanice A doplnit MAC adresu k IP adrese 10.0.0.3 do své ARP tabulky, takže pošle ARP dotaz. Cílová MAC adresa dotazu je MAC všesměrová, tedy ff:ff:ff:ff:ff:ff, což způsobí, že přepínač přepošle zprávu na všechny své výstupní porty. Povšimněme si, že na zprávu odpoví pouze stanice, které se dotaz týká (v tomto případě stanice C) a že odpověď pošle s cílovou MAC adresou odesílatele, tedy MAC-A. Stanice A si z příchozí zprávy vezme zdrojovou MAC adresu, tedy adresu MAC-C a upraví svou ARP tabulku.

Jednotlivé záznamy v ARP musí po určitém čase expirovat. Pokud by k tomu nedocházelo a přitom by došlo např. k výměně síťového adaptéru a tedy ke změně MAC adresy, tak by přidružený záznam v ARP tabulce obsahoval neplatnou hodnotu. Čas expirace se může lišit v různých operačních systémech. Např. Windows a GNU Linux používají výchozí hodnotu 60 vteřin.

Utilita arping

Služeb ARP lze využít (nebo zneužít) na zjištění dostupnosti IPv4 adres v rámci segmentu sítě. K tomuto účelu byla vyvinuta utilita arping, která využívá faktu, že každé zařízení v síti s přidělenou adresou musí odpovídat na ARP dotazy. Utilita má prakticky stejné rozhraní jako utilita ping a vypisuje dokonce podobné informace:

```
$ arping 192.168.1.1
ARPING 192.168.1.1 from 192.168.1.126 wlp1s0
Unicast reply from 192.168.1.1 [8E:19:C5:C3:8A:15]  3.175ms
Unicast reply from 192.168.1.1 [8E:19:C5:C3:8A:15]  3.144ms
Unicast reply from 192.168.1.1 [8E:19:C5:C3:8A:15]  3.119ms
^CSent 3 probes (1 broadcast(s))
Received 3 response(s)
```



Obrázek 4.16: ARP Spoofing - otrava ARP tabulky

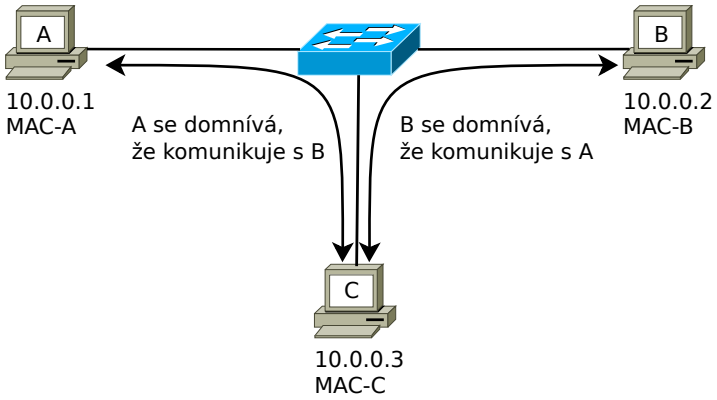
Poznámka: ^C je znaková reprezentace stisku kombinace kláves Ctrl+C, která poslala signál SIGTERM a následně přerušila provádění příkazu.

Další služby ARP

- Proxy ARP - zařízení připojené v segmentu sítě, které odpovídá na ARP dotazy směřující na IP adresy, které se v segmentu nenachází. Proxy ví, kde se nachází cíl takového síťového provozu a nabízí v odpovědi svou vlastní MAC adresu a tak způsobí presměrování provozu na sebe.
- Gratuitous ARP - dobrovolná a nevyžádaná ARP odpověď, která slouží jako oznámení ostatním stanicím ve stejném segmentu jako oznámení o přidělení IP adresy k dané linkové adrese. Stanice ji mohou odeslat na všesměrovou MAC adresu po té co si nastaví novou IP adresu. [12]
- Inverse ARP (InARP) - dodatek k protokolu ARP, který umožňuje zjistit IP adresu k dané MAC adrese. Používal se například s protokolem Frame Relay. [13] [14]

Zranitelnost protokolu ARP

Hlavním problémem protokolu ARP je, že není zabezpečený a počítá s poctivostí všech stanic. Protokol tak může být zneužit pro přesměrování komunikace mezi dvěma přes zařízení útočníka. Předpokládejme segment jako na obrázku 4.16. Předpokládejme, že stanice A a B chtějí vzájemně komunikovat a pro zjištění MAC adres použijí protokol ARP. Utočník, tedy stanice C, se bude snažit odpovědět na ARP dotazy dříve než stanice A respektive B. V odpovědi jim oběma pošle svou vlastní MAC adresu, čímž takzvaně otráví ARP tabulku stanic A a B. Protože odpověď útočníka dorazí dříve, je pravá odpověď ignorována. Technika se proto nazývá ARP Spoofing, nebo ARP Poisoning. Jako výsledek posílají obě stanice svou komunikaci na MAC adresu stanice C, která data může přeposílat a zároveň zaznamenávat, aniž by se o tom stanice A a B dozvěděly (viz obrázek 4.17).



Obrázek 4.17: ARP Spoofing - přemostění komunikace

4.4.4 NAT

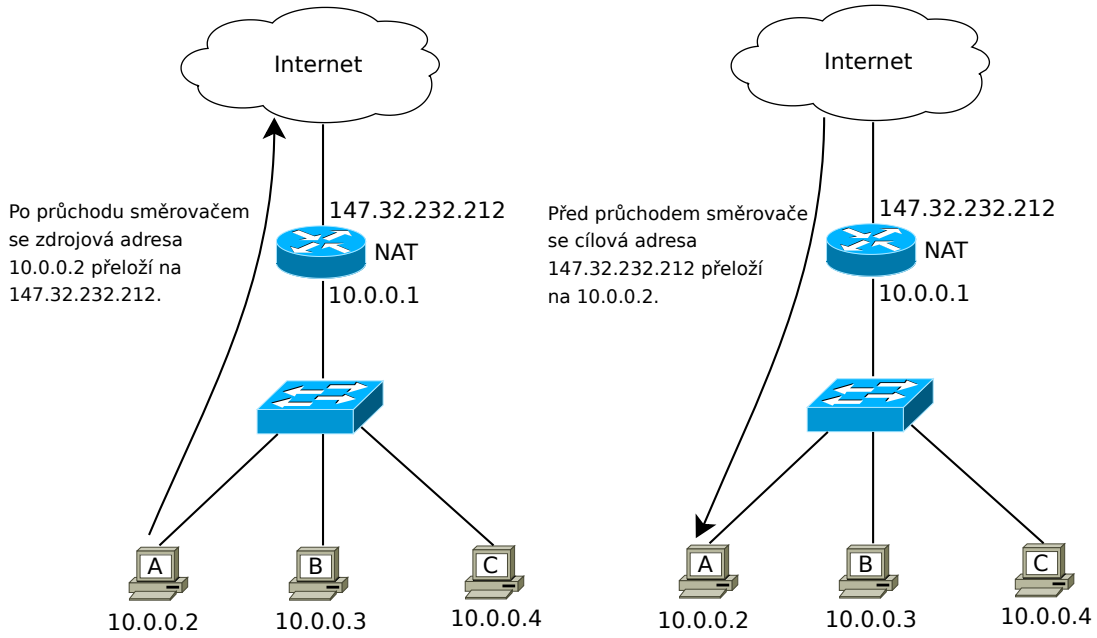
Překlad síťových adres (Network Address Translation - dále jen NAT) je sada technik, které obecně umožňují mapovat adresy z jedné podsítě do druhé a díky tomu lze obejít pravidla pro dělení adresního prostoru. V dnešní době se jednoznačně nejvíce používá technika dynamického překladu, který lze nalézt pod více názvy (např. masquerade nebo overload).

Hlavní myšlenkou dynamického NATu bylo jasně oddělit síť privátní od sítě veřejné (typicky Internetem). Adresy z privátní sítě vždy zůstávají za NATem a vůbec privátní síť neopouští. Diky tomu je možné adresy z privátních rozsahů recyklovat napříč privátními sítěmi po celém světě, což znamená obrovskou úsporu. Ke komunikaci na internetu je pak potřeba alespoň jedna tzv. veřejná IP adresa, která je vždy unikátní v rámci Internetu. Na hranici mezi privátní a veřejnou sítí pak stojí směrovač, který překládá adresy z privátní na jednu čí více veřejných IP adres. Na privátní adresy byly vyčleněny tři rozsahy:

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16

Adresy z těchto rozsahů se nesmí vyskytnout na Internetu. Diky tomu lze každou lokální síť (LAN) považovat za izolovanou síť a proto nevadí, když se tyto rozsahy používají v lokálních sítích znovu a znovu. (Poznámka: V některých situacích to problém je, např. když jsou dvě lokální sítě propojeny přes VPN a tím dojde ke kolizi. Řešení takových situaci je pak velmi problematické.)

Ukázka použití dynamického NATu je naznačena na obrázku 4.18. Nalevo je situace, kdy stanice A odesílá zprávu směrem na internet. Směrovač po průchodu zprávy nahradí (přeloží) zdrojovou IP adresu v hlavičce za vlastní IP adresu (147.32.232.212) a odešle ji do vnější sítě. Cílová stanice nebo server vyrobí odpověd na přijatou zprávu a odešlou ji na IP adresu směrovače (147.32.232.212). Po té, co směrovač tuto zprávu obdrží, přeloží cílovou IP adresu v IP hlavičce na adresu stanice A a přepošle ji cílové stanici. Tato technika je dynamická, ptotože umožňuje tímto způsobem komunikovat na internetu s



Obrázek 4.18: NAT - základní princip

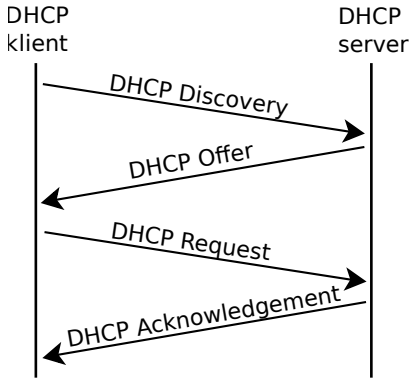
jedinou IP adresou více stanicím najednou. Jakým způsobem dokáže směrovač rozlišit jednotlivé zprávy patřící různým stanicím, si vysvětlíme v kapitole 6.

4.4.5 Protokol DHCP

Jednotlivá síťová rozhraní lze konfigurovat staticky nebo dynamicky. Pro dynamickou konfiguraci se využívá služby DHCP (Dynamic Host Configuration protokol)[15]. Protokol byl definován v roce 1993 na základech staršího protokolu BOOTP[16], který primárně sloužil pro načtení konfigurace a souboru s operačním systémem pro počítače bez pevného disku. Ze svého předchůdce si služba DHCP převzala pouze část pro konfiguraci síťových parametrů a svým klientům nabízí (mimo jiné) následující parametry:

- IP adresu
- Síťovou masku
- Adresu výchozí brány
- Jednu nebo více IP adres služby DNS
- Doménu
- Dobu zapůjčky adresy (lease time)
- Dobu obnovení adresy (renewal time)

Kompletní list všech DHCP parametrů (options) naleznete v příslušném RFC 2132 a jeho updatech. [17]



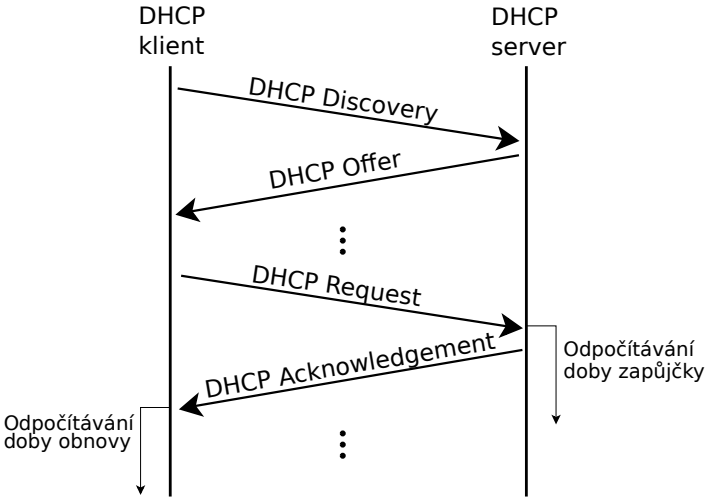
Obrázek 4.19: DHCP - získání konfigurace

Obdržení konfigurace

Služba DHCP se obvykle spouští na směrovači, ale může být spuštěna na jakémkoliv zařízení v segmentu lokální (v některých případech i mimo segment viz. DHCP relay [18]). Důležité je, aby byla dostupná v rámci jednoho síťového segmentu. Zařízení, na kterém běží služba DHCP, nazýváme DHCP serverem. Zařízení, které žádá DHCP server o konfiguraci, se nazývá DHCP klient.

Pro obdržení konfiguraci si musí DHCP klient s DHCP server vyměnit celkem čtyři zprávy (viz obrázek 4.19). Následuje detailní popis zpráv:

- **DHCP Discovery** - klient hledá DHCP server a žádá o konfiguraci. Zpráva se odesílá s cílovou všesměrovou MAC (ff:ff:ff:ff:ff:ff) a cílovou limitovanou všesměrovou IP (255.255.255.255), aby bylo zajištěno, že zprávu dostanou všechna síťová rozhraní v segmentu. Zdrojová MAC adresa se nastaví dle MAC adresy rozhraní klienta a zdrojová IP se nastaví na nedefinovanou adresu (0.0.0.0). Klient ve zprávě může specifikovat speciální příznak, zda se mají odpovědi od serveru posílat na klientovu zdrojovou MAC adresu nebo opět na všesměr, přestože je klientova MAC adresa serveru už známá. Díky tomu ostatní síťová rozhraní dostanou tyto zprávy a ví, co se v síti děje. Může to být užitečné i v situaci, kdy je v segmentu přítomen záložní DHCP server, který tak ví, že byl klient obsloužen. V případě souběhu více klientů najednou, obsahují DHCP zprávy unikátní identifikátor, aby se zprávy jednotlivých klientů snadno odlišily.
- **DHCP Offer** - server po obdržení klientovy žádosti alokuje volnou IP adresu. Před tím, než ji klientovi přidělí, ověří si pomocí ARP protokolu, že tato adresa není již staticky přidělena nějakému síťovému rozhraní v segmentu. Poté pošle klientovi nabídku konfigurace se svou IP adresou jako zdrojovou IP a svou MAC adresou jako zdrojovou MAC. Pro cílovou IP adresu zvolí limitovanou všesměrovou adresu (255.255.255.255) a jako cílovou MAC zvolí adresu klienta nebo všesměr podle příznaku ze zprávy DHCP Discovery.
- **DHCP Request** - klient v této zprávě zopakuje údaje, které obdržel ve zprávě DHCP Offer. Cílová IP a MAC adresa se zvolí buď konkrétní hodnoty nebo opět všesměry v závislosti na příznaku ze zprávy DHCP Discovery. Zdrojová MAC je klientova vlastní a zdrojová IP je opět nedefinovaná IP adresa.



Obrázek 4.20: DHCP - obnovení adresy

- **DHCP Acknowledgement** - v poslední zprávě se zopakuje vlastně vše jako ve zprávě DHCP Offer. Po této zprávě je adresa přiřazena klientovi a klient ji může začít používat pro komunikaci.

DHCP server alokuje IP adresy z přednastaveného rozsahu (anglicky "pool"). Některé servery umožňují staticky alokovat konkrétní IP adresu ke konkrétní MAC adrese. To je velmi užitečné, když chceme zajistit, aby určité zařízení v síti dostali vždy konkrétní adresu (např. databázový server, nebo tiskárna).

Platnost přidělené adresy

Při přidělení adresy klientovi je možné stanovit časový limit, po který je adresa u DHCP služby alokována (položka "Doba zapůjčky"neboli "Lease time"). DHCP klient si může dobu zapůjčky prodloužit odesláním zprávy DHCP Request a opětovným obdržením zprávy DHCP Acknowledgement. Časový interval v jakém by si měl klient adresu obnovit je určen položkou "doba obnovení"(renewal time) v DHCP Offer zprávě. Celá situace je naznačena na obrázku 4.20. Doba obnovení adresy se obvykle nastavuje jako polovina doby zapůjčky.

Uvolnění přidělené adresy

V případě, že klient adresu už nepotřebuje (např. se začne vypínat), tak o tom informuje DHCP server. K tomu slouží zpráva "DHCP Release", kterou klient odešle typicky třikrát. DHCP server tyto zpravy nepotvrzuje.

4.4.6 Automatická bezstavová konfigurace

Proces automatické bezstavové konfigurace (stateless address autoconfiguration) se nazývá také automatická lokální linková konfigurace (link-local address autoconfiguration).

Tento proces se používá ve situaci, když síťové rozhraní nemá statickou adresu a v síti není přítomna služba DHCP. Mnoho operačních systémů používá automatickou link-local konfiguraci jako záložní mechanismus automaticky, když se nepodaří najít DHCP službu. Adresy se pridělují z rozsahu od 169.254.0.0/16, přičemž prvnich a posledních 256 adres jsou rezervovány pro budoucí použití. Operační systém si pak zvolí náhodnou IP adresu z tohoto rozsahu a pak použije protokol ARP pro zjištění, zda tato adresa není již používána jiným síťovým rozhraním v segmentu. [19]

Komunikace v takovém segmentu je značně omezená, protože tento způsob konfigurace nazajišťuje připojení mimo síťový segment. Směrovače ani adresy z link-local rozsahu vůbec nepřeposílají, takže lze komunikovat opravdu pouze s účastníky stejného síťového segmentu.

4.5 Protokol IPv6

V této sekci se podíváme na základní vlastnosti protokolu IPv6. Pro zájemce o tuto problematiku si dovolíme odkázat na vynikající knihu od sdružení NIC.CZ, která je dostupná zdarma a v českém jazyce. [20]

4.5.1 Historie

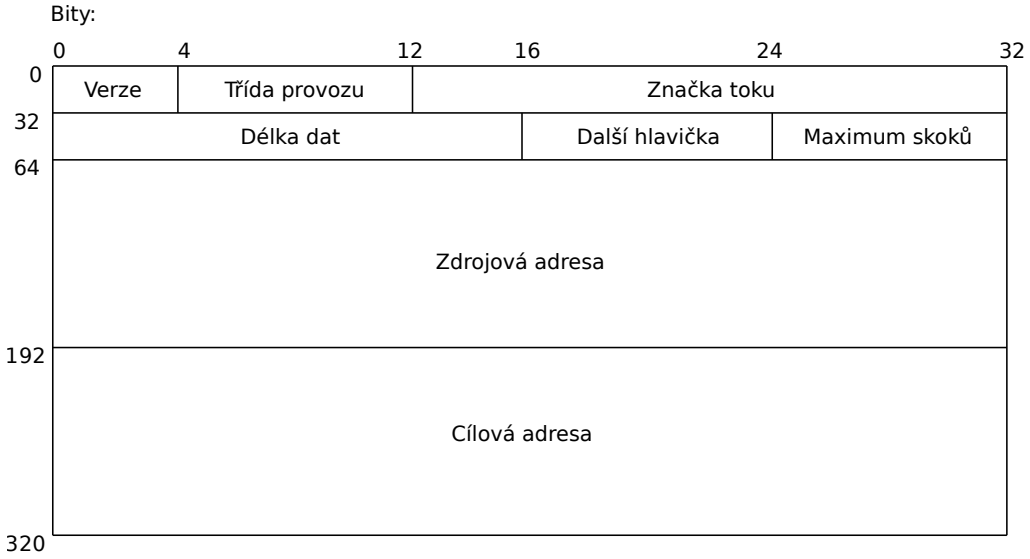
První návrh protokolu IPv6 je již z roku 1995 [21], i když jeho návrh začal mnohem dříve a ještě v době psaní tohoto textu není ani zdaleka tak rozšířený, jak by bylo potřeba. Vznikl jako důsledek nedostatku adres v protokolu IPv4. Délka adresy se oproti starsimu protokolu zečtyřnásobila. Adresní prostor je tak velký, že bychom mohli každému čtverečnímu metru povrchu planety Země přiřadit až $4x10^{18}$ unikátních adres.

4.5.2 Hlavička

Hlavička protokolu IPv6 se oproti svému předchůdci zjednodušila. Tvůrci protokolu se poučili a vynechali například kontrolní součet, který stejně provádí protokoly nižších i vyšších vrstev ISO-OSI modelu. IPv6 paket se skládá z hlavičky a dat. Data obsahují typicky hlavičky a data vyšších protokolů. Hlavička se skládá ze základní hlavičky a rozšiřujících hlaviček. Základní hlavička je v paketu přítomna vždy, rozšiřující hlavičky jsou nepovinné a když už jsou přítomny, lze je řetězit jednu za druhou. Hlavička podporuje fragmentaci, nicméně se už vůbec nepočítá s tím, že by fragmentaci prováděly směrovače. Základní hlavičku si můžeme prohlédnout na obrázku 4.21. V následujících odstavcích se podíváme na nejdůležitější položky. [22]

Verze

Stejně jako v IPv4 na začátku hlavičky máme 4-bitový identifikátor verze protokolu, v tomto případě IPv6.



Obrázek 4.21: Hlavička IPv6

Třída provozu

8-bitový identifikátor třídy provozu umožňuje rozlišit různé typy komunikace, které mají jiné nároky na kvalitu služeb. Například lze takto označit pakety telefonické komunikace a ty jsou pak síťovými prvky zpracovány prioritně.

Značka toku

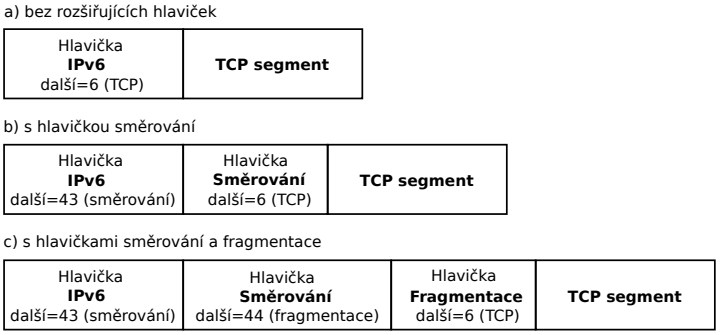
Následujících 20 bitů IPv6 hlavičky je vyhrazeno pro tzv. značku toku. Tento identifikátor pomáhá určit, které pakety patří do stejné komunikace, což by mělo pomáhat síťovým prvkům k efektivnějšímu zpracování. Tato položka hlavičky může být síťovými prvky ignorována, aniž by to mělo vliv na doručení paketu.

Délka dat

Položka určuje velikost přenášených dat bez hlaviček v bajtech. Vhledem k tomu, že se jedná o 16-bitovou hodnotu, tak je maximální velikost dat 64KB.

Další hlavička

Určuje typ následující rozšiřující hlavičky, která následuje za hlavní hlavičkou (viz 4.5.3) nebo typ protokolu na vyšší vrstvě (stejně jako položka "Protokol"v hlavičce protokolu IPv4, viz kapitola 4.2.3.



Obrázek 4.22: Rozšiřující hlavičky v IPv6

Max. skoků

Tato položká má stejný význam jako položka TTL v hlavičce IPv4 paketu, viz kapitola 4.2.3.

Zdrojová a cílová adresa

Posledními položkami v hlavičce jsou adresy pro komunikaci, které zabírají většinu velikosti hlavičky.

4.5.3 Rozšiřující hlavičky

Rozšiřující hlavičky jsou nepovinné. Umísťují se mezi základní hlavičku a data paketu. Každý typ rozšiřující hlavičky má své unikátní internetové číslo protokolu, tedy úplně stejně jako jakýkoliv protokol, který se enkapsuluje do hlavičky IPv4 nebo IPv6 protokolu. Jednotlivé hlavičky lze řetězit pomocí položky *Další hlavička*, která se nachází v základní hlavičce i v každé rozšiřující hlavičce. Položka může tedy odkazovat na následující rozšiřující hlavičku nebo na hlavičku protokolu na vyšší vrstvě. Ukázku zřetězení hlaviček můžeme vidět na obrázku 4.22.

Díky principu zřetězení hlaviček nemusí být např. informace o fragmentaci přítomny v každém paketu a zároveň rozšiřující hlavičky mohou nabídnou podrobnější informace, než původní položky v protokolu IPv4. Dalším zajímavým faktem je, že ne všechny síťové prvky musí podporovat všechny rozšiřující hlavičky. Některé z hlaviček jsou označeny jako povinné (např. směrování, fragmentace, šifrování) a některé jsou označeny jako volitelné (např. mobilita) [20]. V případě, že je více rozšiřujících hlaviček v jednom paketu, jejich pořadí je doporučeno v [22].

4.6 Dělení adresního prostoru v IPv6

Způsob adresace prošel významnými změnami. Adresa se rozšířila na 128 bitů. Princip masky zůstal zachován, jen se změnila pravidla, kde lze adresu maskou rozdělit. Dále došlo ke změně zápisu adresy a masky. Další významnou změnou je, že síťové rozhraní může mít

více adres (a v drtivé většině případů dokonce musí mít). Asi nejdůležitější změnou však je, že koncový uživatel nedostává jednu konkrétní adresu, ale rovnou celý prefix. Velikost unikátního adresního prostoru pro každého zákazníka může být od 2^{64} až do 2^{80}. Vše o adresaci v IPv6 nalezneme v [23].

4.6.1 Zápis adresy

Adresa se zapisuje hexadecimálně, což vede ke snazšímu převodu adresy do binární a dělení adresního prostoru je tak pohodlnější. Hexadecimální zápis je zároveň úspornější a dovoluje nám trochu zkrátit zápis již tak velmi dlouhých adres. Zápis adresy lze dále zkracovat dalšími technikami. Čtveřice cifer se v adrese oddělují dvojtečkou.

Podívejme se na ukázku zápisu IPv6 adresy:

```
1234:0217:2344:0000:2345:0000:0000:1111
```

Toto je nejdelší forma zápisu IPv6 adresy. Naštěstí pro uživatele lze adresu zkrátit vynecháním nevýznamných nul zleva. Zde máme zápis stejné adresy:

```
1234:217:2344:0:2345:0:0:1111
```

To však není všechno. Pokud se v adrese vyskytne sekvence dvou a více 0 oddělených dvojtečkami, lze je nahradit symbolem "::", viz následující ukázka:

```
1234:217:2344:0:2345::1111
```

Na tomto místě je nutné poznamenat, že takové zkrácení lze v rámci jedné adresy provést pouze jednou. Protože zápis "::"nespecifikuje kolik nul bylo takto zkráceno, vícenásobné použití této techniky by vedlo k nejednoznačnému zápisu adresy. Následující ukázka představuje adresu se samými nulami:

```
::
```

Taková adresa má uplatnění např. ve směrovacích tabulkách.

4.6.2 Rozdělení adres

Adresy se v IPv6 rozdělují podle prefixu. V tabulce 4.5 máme přehled základních druhů adres. V následujících sekcích se podíváme podrobněji na nejdůležitější druhy.

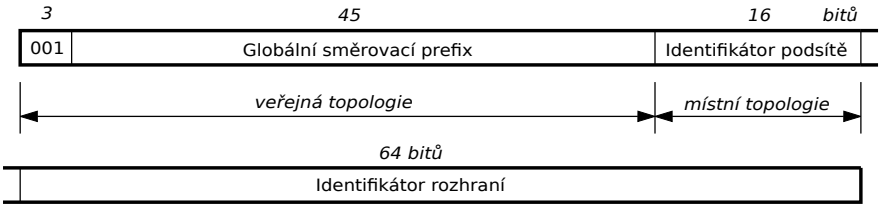
4.6.3 Globální individuální adresy

Obrovský adresní prostor v IPv6 umožňuje přidělit každému síťovému rozhraní na světě globálně unikátní adresu. Už není potřeba schovávat privátní adresy za jedinou veřejnou

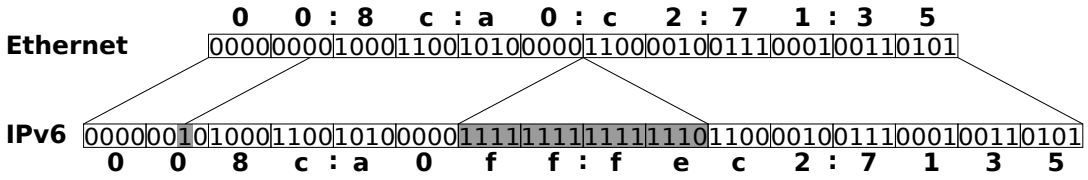
Prefix	Význam
::/128	nedefinovaná adresa
::1/128	smyčka (loopback)
fc00::/7	unikátní individuální lokální
fe80::/10	individuální lokální linkové
ff00::/8	skupinové adresy
64:ff9b::/96	adresy s vloženým IPv4
64:ff9b:1::/48	lokální adresy pro přechodové mechanismy
2001::/32	Teredo
2001:db8::/32	adresy pro příklady v dokumentech
2002::/16	6to4
ostatní	individuální globální

Tabulka 4.5: Vybrané prefixy

adresu. Adresy se jednotlivým poskytovatelům přidělují podobným způsobem, jako se přidělovaly bloky veřejných adres v IPv4. Uživatel pak místo jedné adresy dostane jeden globální směrovací prefix, tedy blok adres o minimální velikosti 2^{64}. První tři bity mají typicky hodnotu 001. Posledních 64 bitů adresy slouží vždy jako unikátní identifikátor rozhraní. Na obrázku 4.23 vidíme typické rozdělení globální individuální adresy. Za povšimnutí stojí fakt, že adresy v IPv6 nelze maskou rozdelit kdekoliv jako tomu bylo v IPv4. Dálka prefixu se může pohybovat pouze od 48 do 64 bitů. Minimální velikost podsítě je tedy 2^{64}. Pokud tedy uživatel dostane od poskytovatele přidělený prefix o délce 64 bitů, nemůže jej dále dělit na menší podsítě. Pokud dostane naopak prefix o délce 48 má k dispozici celých 16 bitů na tvorbu unikátních identifikátorů podsítě.



Obrázek 4.23: Obvyklá struktura globální idnividuální adresy



Obrázek 4.24: Tvorba EUI-64 z MAC adresy

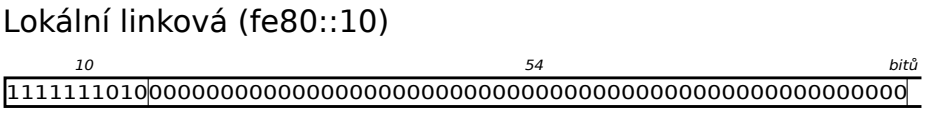
4.6.4 Identifikátory rozhraní

Nyní se zaměřme na druhou část adresy, tedy identifikátor rozhraní. Jeho délka je opravdu enormní a hlavní motivací pro takto velký adresní prostor bylo maximálně zjednodušit automatickou konfiguraci sítě. [20] Identifikátor rozhraní v adrese rozhraní můžeme nastavit třemi způsoby:

- Manuální konfigurace - identifikátor rozhraní máme plně pod kontrolou, ale nemáme jistotu, že je v rámci segmentu unikátní.
- Automatické vygenerování - specifikace IPv6 umožňuje automatické vygenerování identifikátoru např. na základě MAC adresy rozhraní. Díky mechanismu objevování sousedů je automaticky zajištěno, že je identifikátor unikátní v rámci segmentu (viz SLAAC 4.7.4).
- Přidělení službou DHCPv6 - adresy lze přidělovat i centrálně podoně jako tomu bylo v IPv4. Lze dokonce zafixovat konkrétní IPv6 adresu konkrétnímu klientovi (viz DHCPv6 4.7.4).

Automatické vygenerování identifikátoru se nejčastěji provádí z MAC adresy, což je dáno především dominancí protokolů Ethernet a WiFi, které shodně používají stejný typ adres. Tento typ identifikátoru je nazýván EUI-64 [24]. Způsob generování je naznačen na obrázku 4.24. Protože je identifikátor rozhraní od dva bajty delší než MAC adresa, tak se roztáhne na osm bajtů vložením dvojice bajtů s hodnotami 0xFFFE doprostřed původní adresy. Dále se invertuje sedmý bit identifikátoru. Důvodem je, že tento bit určuje zda je adresa spravována lokálně nebo globálně a proto se invertuje typicky z hodnoty 0 na 1, protože MAC adresa má tento bit vždy nastavený na 0. Takto vytvořený identifikátor by měl být unikátní, za předpokladu, že MAC adresy jsou unikátní. Unikátnost adres je navíc zajištěna technikou nazývanou "objevování sousedů"(viz kapitola 4.7.2).

V praxi se ukázalo, že není z bezpečnostního hlediska vhodné, když je identifikátor stále stejný a ještě navíc z něj lze odvodit MAC adresu. Řešením je generovat identifikátor zcela náhodně v určitých časových intervalech (typicky dva dny). Stanice pak pro komunikace používá adresu s posledním náhodným identifikátorem, nicméně adresy s identifikátorem odvozeným z MAC adresy jsou stále platné a lze na nich příjímat komunikaci. Každé zařízení v síti pak může mít dokonce několik platných globálních individuálních adres najednou. Unikátnost náhodných identifikátorů je pak zajištěna stejným způsobem jako u těch odvozených z MAC adres, tedy pomocí technniky objevování sousedů, viz 4.7.2. [25]



Obrázek 4.25: Obvyklá struktura indviduální linkové adresy

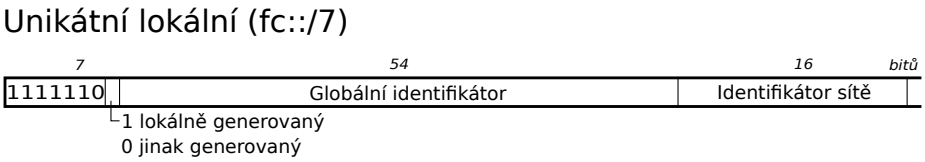
4.6.5 Individuální linkové adresy

Narozdíl od globálních individuálních adres, které lze použít pro komunikaci v rámci celého internetu, mají individuální linkové adresy omezený rozsah pouze v rámci segmentu. Složení lokálních linkových adres je na obrázku 4.25. Všechny mají shodný prefix *fe80::10*, který je typicky doplněn nulami až do poloviny adresy. Druhá polovina obsahuje identikátor rozhraní vygenerovaný z MAC adresy. Účelem lokálních linkových adres je zajistit komunikaci uvnitř segmentu i bez explicitní konfigurace. Adresy se generují automaticky a každé síťové rozhraní používající IPv6 protokol pro komunikaci musí mít právě jednu lokální linkovou adresu. Lokání linkové adresy se nesměrují, tedy stejně jako MAC adresy nikdy neopouští síťový segment. (Poznámka: Prefix individuálních linkových adres je obsažen ve směrovacích tabulkách, ale protože je pro všechny segmenty stejný, nelze jej pro směrovaní použít.)

4.6.6 Unikátní individální lokální adresy

Rozsah je určen pro unikátní adresy v rámci organizace. Z jistého pohledu se dá říci, že se jedná o obdobu privátních adres z IPv4. Jejich účel však není skrývat se za NA-Tem, ale poskytnout větším organizacím jednotný adresní prostor, který lze spravovat, bez závislosti na prefixu sítě obdrženého od poskytovatele připojení. V rámci organizace má pak smysl vázat služby raději na unikátní indiviuální lokální adresy než na globální individulání adresy. Pokud dojde k výpadku služeb od poskytovatele připojení a globální prefix není k dispozici, všechny lokální služby tak zůstanou nadále dostupné. Individuální lokální adresy se směrují pouze uvnitř lokální sítě, nesmí se směrovat v Internetu. Strukturu individuální lokální adresy lze vidět na obrázku 4.26. Její struktura je v principu podobná globální individuální adrese, jen má povinný prefix *fc::/7* a osmý bit určuje, zda byla adresa generována lokálně čí jinak. Zbytek první poloviny adresy připadá na globální identifikátor a identifkátor podsítě, druhá polovina pak obsahuje identifikátor rozhraní.

Globální identifkátor má 40 bitů a generuje se náhodně. Používá se v situaci, kdy má např. jedna organizace více fyzicky oddělených sítí v různých pobočkách. Každá pobočka má pak svůj originální prefix. Identifikátor podsítě má 16 bitů a odlišuje jednotlivé adresy podsítí v rámci jedné pobočky. Více informací viz. [20].



Obrázek 4.26: Obvyklá struktura idnividuální lokální adresy

Adresa skupiny	Význam
ff02::1	Všechna IPv6 zařízení
ff02::2	Všechny IPv6 směrovače
ff02::5	Směrovače s protokolem OSPF 5.3.3
ff02::FB	Zařízení s protokolem mDNSv6 ??

Tabulka 4.6: Vybrané skupinové adresy

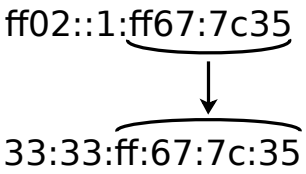
4.6.7 Skupinové adresy

Skupinové adresy představují zásadní koncept, který výrazně odlišuje protokol IPv6 od svého předchůdce. Adresace IPv6 opustila všesměrovou komunikaci a nahradila ji skupinouvou. Hlavní myšlenka je, že pokud chceme komunikovat jen s určitou podmnožinou zařízení v síti, nemusíme komunikací obtěžovat všechny. Každé zařízení patřící do určité skupiny pak naslouchá na příslušné adrese a ostatní zařízení mohou tuto komunikaci ignorovat. IPv6 definuje celou řadu skupin. Skupinové adresy začínají vždy prefixem *0xff*. Bajt následující za prefixem určuje volby a dosah adresy. Podrobnější rozbor volem a dosahu adresy je nad rámec tohoto textu, tak se spokojíme s informací, že hodnota *0x02* znamená, že dosah adresy je pouze v segmentu sítě (tedy lince). Nás v tomto textu budou zajímat pouze adresy s tímto dosahem.

V tabulce 4.6 úvádíme několik vybraných skupinových adres. Každá skupina sdružuje zařízení s nějakou specifickou funkcí. Díky tomu je možné jednoduše oslovit např. všechny směrovače ve stejném segmentu (adresa ff02::2). Skupinová adresa *ff02::1* reprezentuje všechna IPv6 zařízení v segmentu, takže by se dala považovat za jedinou všesměrovou adresu, protože do této skupiny se přihlašují automaticky všechna síťová rozhraní provozující protokol IPv6.

4.6.8 Skupinové adresy a linková vrstva

Skupinové IPv6 adresy mají pro MAC adresy rezervovaný 16 bitový prefix 33:33. Zbylých 48 bitů se vezme z posledních 48 bitů skupinové IPv6 adresy, viz obrázek 4.27. Přepínače musí tento prefix podporovat a zacházet s ním stejně jako s všesměrovou MAC adresou -



Obrázek 4.27: Vazba mezi multicastovou IPv6 adresou a MAC adresou

tedy rozeslat rámec s touto cílovou MAC adresou na všechny výstupní porty, kromě toho, ze kterého rámec přišel.

Pokud se skupinová adresa z pohledu přepínače chová jako všesměrová, v čem tedy spočívá její výhoda? Odpověď nalezneme na síťové kartě, která musí vždy přijmout rámce s všesměrovou cílovou MAC adresou. V případě MAC adres s prefixem 33:33, lze na kartě nastavit, které z nich bude přijímat a které ignorovat. Díky tomu lze přímo hardwarově filtrovat příchozí zprávy a kontrolovat jestli patří do sledované skupiny, aniž by se tím musel zabývat operační systém.

4.6.9 Skupinová adresa vyzývané stanice

Každé síťové rozhraní má ještě navíc přidělenu jednu nebo více tzv. skupinových adres vyzývané stanice. Tyto adresy se vytváří automaticky pro každou existující ne-skupinovou IPv6 adresu a jsou vždy ve tvaru *ff02::1:ffxx:xxxx*, kde posledních 24 bitů označených písmenem x odpovídá posledním 24 bitům přidružené ne-skupinové IPv6 adresy. Pokud je na síťovém rozhraní více ne-skupinových IPv6 adres, které mají shodných posledních 24 bitů, vytváří se pouze jedna skupinová adresa vyzývané stanice. Ve výsledku má každé síťové rozhraní v síťovém segmentu unikátní skupinovou adresu (tedy pokud v síti neexistuje náhodou více síťových rozhraní se shodnými posledními 24 bity v ne-skupinové IPv6 adrese). Skupinové adresy vyzývané stanice se uplatňují při zjišťování linkových adres v lokální síti (více v sekci 4.7.3)

4.6.10 Kolik a jakých adres má rozhraní v IPv6?

Každá stanice v síti s IPv6 protokolem má obvykle hned několik IPv6 adres. Na ukázku si vezmeme stanici, která má MAC adresu například *00:11:22:33:44:55*. Dále předpokládejme, že stanici nastavíme manuálně adresu *2001:db8::ab:cdef/64*. Na rozhraní pak můžeme nalézt následující adresy:

- ff02:01 - skupinová adresa, kterou má povinně každé síťové rozhraní s IPv6 protokolem. Tato adresa je v podstatě obdobou všesměrové adresy.
- fe80::0211:22ff:fe33:4455/64 - lokální linková adresa vytvořená z MAC adresy.
- ff02::1:ff33:4455 - skupinová adresa vyzývané stanice přidružená k lokální linkové adrese
- 2001:db8::ab:cdef/64 - individuální globální adresa

- ff02::1:ffab:cdef - skupinová adresa vyzývané stanice přidružená k individuální globální adrese

Síťovému rozhraní lze přidělit ještě více adres manuálně nebo pomocí automatické konfigurace (viz sekce 4.7.4). Pro každou takovou adresu se automaticky vytváří skupinová adresa vyzývané stanice, pokud již neexistuje.

4.7 Doplňkové služby k protokolu IPv6

Protokol IPv6 přináší mnoho nových vlastností a služeb. Dále se k němu přidružují specifické protokoly jako ICMPv6 a DHCPv6. V následující části se podíváme na vybrané služby IPv6 a jemu přidružené protokoly. Budeme se však zabývat jen těmi nejzajímavějšími.

4.7.1 Protokol ICMPv6

Pro diagnostiku a další doplňkové služby v IPv6 slouží "šestková"varianta protokolu ICMP [26]. Základní funkce předchozí verze byly zachovány a zároveň byly přídány funkce nové. Každá funkce má svůj typ zprávy se specifickým číselným identifikátorem. Komplentí výčet typů nalezneme v příslušném RFC dokumentu. V tomto textu se budeme dále zabývat jen zprávami 133 až 136, 141 a 142.

4.7.2 Objevování sousedů

Metoda objevování sousedů se využívá v provozu IPv6 hned na několik funkcí:

- **Automatická bezstavová konfigurace** - mechanismus, který dokáže nahradit většinu funkcionality služby DHCP, více v sekci 4.7.4.
- **Zjišťování linkových adres v lokální síti** - mechanismus, který zcela nahrazuje protokol ARP 4.4.3, se kterým IPv6 vůbec nepočítá. Více v sekci 4.7.3.
- **Hledání směrovačů** - zjistění, přes které směrovače je možné přeposílat provoz určený pro stanice umístěné v jiných sítích.
- **Přesměrování** - směrovač může informovat stanice v síti, že existuje lepší směrovač s kratší cestou k cíli.
- **Ověřování dostupnosti sousedů** - každá stanice v síti může jednoznačně určit dostupnost jiné stanice. Jedná se v podstatě o obdobu služby ARPing (4.4.3).
- **Detekce duplicitních adres** - hledání a deaktivace chybných adres.

Všechny uvedené funkce využívají pro svou činnost pouhých pět zpráv ICMPv6 protokolu, konkrétně se jedná o:

- Router Solicitation (typ 133) - výzva směrovači

- Router Advertisement (typ 134) - ohlášení směrovače
- Neighbor Solicitation (typ 135) - výzva sousedovi
- Neighbor Advertisement (typ 136) - ohlášení souseda
- Redirect (typ 137) - informace o směrovači s kratší cestou k cíli.

Úplný výčet funkcí naleznete v [27].

4.7.3 Zjišťování linkových adres v lokální síti

Funkcionalita protokolu ARP (4.4.3) byla v IPv6 kompletně nahrazena metodou objevování sousedů. Stejně jako u protokolu IPv4 potřebujeme zjišťovat linkové adresy k daným IPv6 adresám, abychom mohli správně vyplnit hlavičku linkové vrstvy a umožnili tak přepínačům efektivně přeposílat rámce. Zjišťování linkové adresy probíhá v následujících krocích.

Krok 1 - sestavení adresy vyzývané stanice

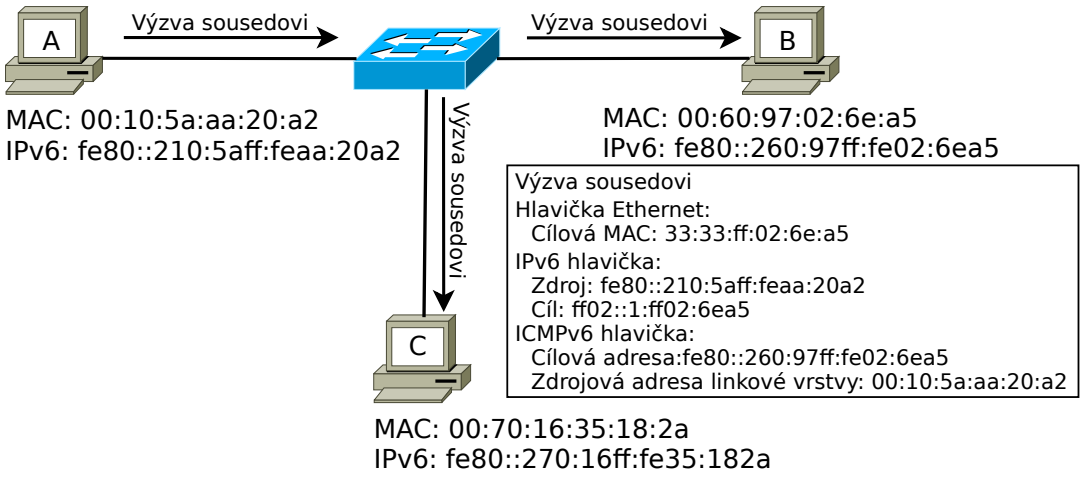
Na začátek si připomeňme, jak linkovou adresu získává ARP protokol - odešle zprávu s všesměrovou linkovou adresou (v případě Ethernet protokolu se jedná o *ff:ff:ff:ff:ff:ff*), kterou musí přijmout každá síťová karta a přijatou zprávu pak musí zpracovat operační systém. Ve výsledku to znamená, že operační systém reaguje na všechny ARP dotazy, i na ty, které pro něj nejsou určeny.

Protokol pro objevování sousedů využívá pro hledání linkové adresy speciální skupinovou adresu vyzývané stanice (viz 4.6.9). Díky tomu je zajištěno, že komunikace odeslaná na tuto adresu je zpracována pouze potenciálním příjemcem zprávy (až na výjimku popsanou v 4.6.9) a ostatní stanice v síti se touto komunikací vůbec nezabývají (viz 4.6.8).

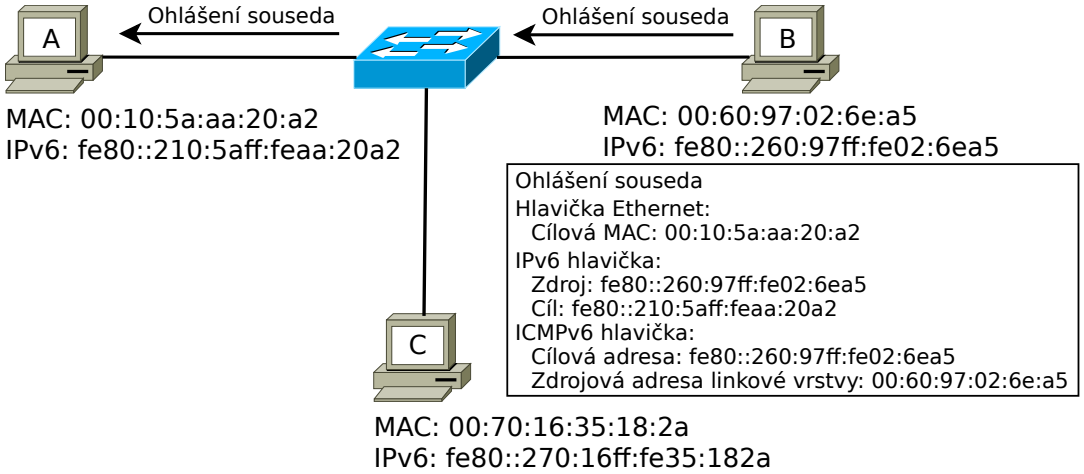
Skupinovou adresu vyzývané stanice vytvoříme kombinací prefixu *ff02::1:ff/104* a posledních 24 bitů cílové IPv6 adresy. Pokud je cílová IPv6 např. *fe80::260:97ff:fe02:6ea5*, tak odpovídající skupinová adresa vyzývané stanice bude *ff02::1:ff02:6ea5*.

Krok 2 - odeslání výzvy sousedovi

Pro vysvětlení dalšího kroku se podívejme na modelovou situaci na obrázku 4.28. Máme zde dvě stanice A a B, přičemž stanice A chce zjistit linkovou adresu stanice B. K tomu potřebuje odeslat výzvu sousedovi (*Neighbor solicitation*, viz 4.7.2) na skupinovou adresu vyzývané stanice B, kterou jsme sestavili v prvním kroku. Zároveň se k tomu vytvoří i cílová skupinová MAC adresa *33:33:ff:02:6e:a5* (viz 4.6.8), díky které je zpráva přepínači doručena všem stanicím ve stejném segmentu sítě. Tuto zprávu zpracují pouze stanice, které poslouchají na adrese *ff02::1:ff02:6ea5*. Ostatní stanice zprávu zahodí ještě na linkové vrstvě, protože jejich síťová karty nepřijímají MAC adresu *33:33:ff:02:6e:a5*. Na situaci z obrázku tak přijme a zpracuje zprávu pouze stanice B, protože se její adresa shoduje s cílovou adresou v ICMPv6 hlavičce.



Obrázek 4.28: Výzva sousedovi

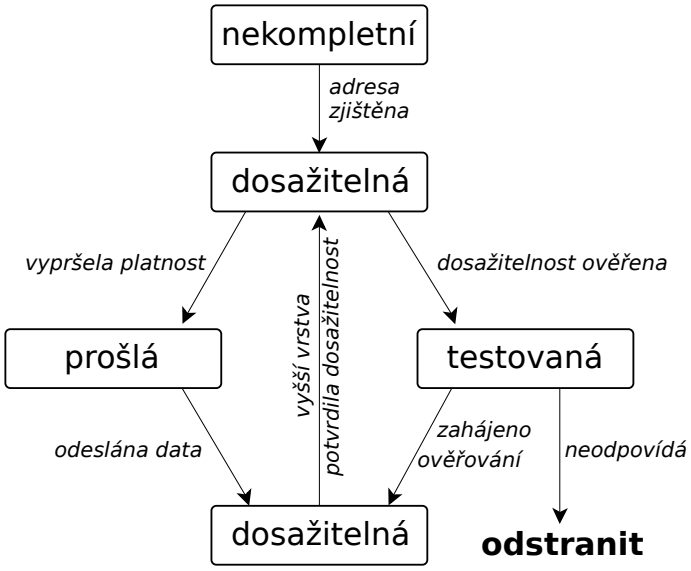


Obrázek 4.29: Ohlášení souseda

Poznámka: Ve stejném síťovém segmentu se mohou vyskytnout další stanice, které mají shodnou skupinovou adresu se stanicí B, ale jejich IPv6 adresa se bude lišit od cílové adresy z ICMPv6 hlavičky a budou tedy zprávu ignorovat.

Krok 3 - ohlášení souseda

Stanice B reaguje na výzvu odesláním ohlášení souseda (*Neighbor Advertisement*, viz 4.7.2), kterou pošle na lokální linkovou adresu stanice A získanou z výzvy sousedovi. Součástí odpovědi v hlavičce ICMPv6 je i položka *Zdrojová adresa linkové vrstvy*, která obsahuje linkovou adresu stanice B, tedy MAC adresu, kterou se stanice A pokoušela zjistit. Situaci máme znázorněnu na obrázku 4.29.



Obrázek 4.30: Přehled stavů záznamu v tabulce sousedů

Krok 4 - vložení záznamu do tabulky sousedů

Stanice A si získanou linkovou adresu uloží do tabulky sousedů a může začít komunikaci. Tabulka sousedů je v podstatě obdobou ARP tabulky (4.4.3), kde se uchovávají vazby mezi síťovou adresou a linkovou adresou. Záznamy v tabulce mají omezenou platnost, proto se tabulka chová v podstatě jako cache a záznamy je nutné obnovovat. Stavy záznamu a přechody mezi nimi jsou naznačeny na obrázku 4.30.

4.7.4 Automatická konfigurace v IPv6

Automatická konfigurace síťových rozhraní je v IPv6 zajištěna dvěma metodami - bezstavovou (SLAAC, DHCPv6) a stavovou (DHCPv6). Oba typy konfigurace spolu mohou koexistovat v jednom segmentu. Záleží na implementaci klientské stanice, zda jí stací bezstavová konfigurace, nebo zda ještě navíc požádá o IP adresu přes službu DHCPv6. V takovém případě bude mít klientská stanice přiřazených více adres, viz sekce 4.6.10.

Automatická bezstavová konfigurace - SLAAC

Bezstavovou konfiguraci (neboli *StateLess Address Auto Configuration*[28]) musí podporovat každý směrovač operující na protokolu IPv6. Díky tomu není již vůbec potřeba zadávat IP adresy ručně. Směrovač nepřiděluje konkrétní adresy jednotlivým stanicím v síti, ale pouze jim předá informaci, aby si každá stanice mohla adresu vygenerovat sama. Díky tomu směrovač nepotřebuje udržovat informace o přidělené adrese v paměti (tedy nemusí držet stav). Proces přidělení adresy lze popsat následujícími kroky:

1. Stanice obdrží zprávu *Router Advertisement* (RA), kterou směrovače rozesílají periodicky nebo na výzvu *Router Solicitation* (RS). Výzvu může odeslat každá stanice

a urychlit tím odeslání RA (4.7.2). Zpráva RS se posílá na skupinovou adresu směrovačů, tedy na *ff02::2*. Zpráva RA se posílá na skupinovou adresu všech, tedy na *ff02::1*.

2. Zpráva RA obsahuje (mimo jiné) síťový prefix, pro aktuální síťový segment. Stanice má po jeho obdržení dostatek informací, aby si mohla vygenerovat unikátní globální (4.6.3) nebo unikátní lokální adresu (4.6.6) za pomoci automaticky vygenerovaného identifikátoru rozhraní (4.6.4).

3. Po vygenerování adresy musí stanice ověřit její unikátnost za pomoci metody, která je v principu podobná metodě pro zjišťování linkových adres v lokální síti 4.7.3. Jediným rozdílem je, že se nesnažíme zjistit cílovou linkovou adresu, ale pouze zda existuje v síti stanice s danou IPv6 adresou. Pokud taková stanice existuje a odpoví na naši výzvu, adresu nelze použít a je nutné vygenerovat jinou.

Bezstavová metoda umožňuje stanicím obdržet síťový prefix a adresu výchozí brány - což je lokální linková adresa směrovače v RA zprávě. Pokud stanice potřebuje získat i adresu DNS služby, musí použít DHCPv6.

Poznámka: Operační systémy určené pro uživatele (Microsoft Windows, MacOS, GNU Linux) ještě automaticky přidělují síťovému rozhraní ještě dodatečnou globální individuální adresu s obdrženým prefixem a náhodně vygenerovaným identifikátorem rozhraní. Tato adresa se pak používá pro komunikaci do vnější sítě (např. do internetu). Důvodem je fakt, že adresa vytvořená z MAC adresy je fixní a představovala by snadný cíl útoků z venku. Adresy s náhodně vygenerovaným identifikátorem rozhraní mají omezenou platnost a po nějaké době se přestanou používat a vygenerují se nové. Konkrétní chování je závislé na implementaci operačního systému.

Automatická stavová konfigurace - DHCPv6

Bezstavová automatická konfigurace je plně dostatečná pro připojení klientů k síti. Její nevýhodou je, že takto vygenerované adresy jsou závislé na MAC adrese síťové karty. Pokud chceme v sítí provozovat zařízení, jehož IPv6 adresa má být statická (např. tiskárna), potřebujeme mechanizmus, který zajistí přidělení identifikátoru nezávislém na MAC adrese. Dalším problémem je, že SLAAC nedává informaci o DNS. Tuto funkcionalitu zajišťuje protokol DHCPv6 [29], který funguje v principu stejně jako jeho předchůdce DHCP. Rozdílem je, že může pracovat ve dvou režimech (oba reřimy lze provozovat současně):

- Stavový režim - přidělování IPv6 adres. V takovém případě si DHCPv6 server musí pamatovat jaké adresy a komu přidělil (tedy musí si pamatovat stav). Přidělení jednotlivých IP adres lze pak zafixovat podle jednoznačného identikátoru DUID (*DHCP Unique IDentifier*), kterým se jednotlivé stanice identikují vůči DHCPv6 serveru. Pro přidělení adresy je potřeba čtyř zpráv, stejně jako u DHCP.
- Bezstavový režim - používá se vetšinou jako doplněk k SLAAC pro informaci o adrese DNS. V takovém případě si server nemusí pamatovat žádnou informaci o klientovi a pro získání informace stačí pouze dvě zprávy.

4.7.5 Přechodové mechanismy

Vzhledem k historické rozšířenosti protokolu IPv4, je jasné, že jej nelze jednoduše vypnout a okamžitě přejít na IPv6. Přechod musí být pozvolný a musí k němu dojít za chodu. Z těchto důvodů vzniklo mnoho tzv. přechodových mechanizmů, které umožňují oběma protokolům koexistovat než dojde k úplnému přechodu. Některé z těchto mechanizmů uvádíme v následujících sekcích.

NAT64

Jedná se o mechanizmus umožňující přístup ze sítě IPv6 do sítě IPv4 [30]. Cílem je umožnit stanici ze sítě s IPv6 dosáhnout stanici v síti IPv4. Toto řešení se nazývá NAT64 (původní název byl NAT-PT). Zjednodušeně by se dalo říci, že se jedná o překlad IPv6 na IPv4 adresy na speciálních bránách, na kterých běží současně oba protokoly. Hlavní myšlenko tohoto řešení spočívá v namapování celého adresního prostoru IPv4 do podprostoru v IPv6. Pro mapování se pak používá prefix *64:ff9B::/96*, nebo podle specifikace správce sítě. Posledních 32 bitů pak odpovídá přesně 32 bitům příslušné IPv4 adresy. Pro zjednodušení zápisu přechodové IPv6 adresy je možné zapsat posledních 32 bitů adresy v notaci pro IPv4 adresu [31]. Například přechodová IPv6 adresa pro IPv4 adresu *233.252.196.130* lze zapsat jako *64:ff9B::233.252.196.130/96*. Překlad adres mezi IPv6 a IPv4 je vcelku komplikovaný, kvůli poměrně značným rozdílům mezi jednotlivými protokoly. Současné implementace jsou již z principu plny kompromisů a tudíž NAT64 nedokáže zpřostředkovat úplnou kompatibilitu mezi IPv6 a IPv4.

Výhody

- Adresy protkolu IPv4 jsou dostupné v adresním prostoru IPv6
- Uživatelské stanice i infrastruktura poskytovatele může provozovat pouze protokol IPv6

Nevýhody

- Adresy IPv6 nejsou z principu dostupne z IPv4 adres
- Pokročilé funkce IPv6 (např. mobilita, rozšiřující hlavičky) jsou nedostupné po překladu do IPv4
- Vyžaduje existenci speciálních NAT64 serverů, přes které musí proudit veškerá komunikace mezi IPv6 a IPv4, což vyžaduje poměrně velkou zátěž

Dual-Stack Lite

Dual-Stack Lite nabízí nekonvenční řešení, jak umožnit komunikovat stanicím v síti přes obě verze protokolu IP, aniž by bylo nutné udržovat síťovou infrastrukturu s oběma protokoly zároveň. Páteř sítě je tvořena pouze protokolem IPv6, přičemž jednotlivé stanice v síti tvoří tunel pro protokol IPv4 směrem ke stroji, kde běží NAT. Jednotliví hosté mají k dispozici nativní IPv6 síť a zároveň mají na tunelu k dispozici i spojeni přes IPv4. [32]

Výhody

- Poskytovatel připojení k internetu na svých interních prvcích může provozovat pouze IPv6.
- Velmi laciné řešení.
- IPv4 je řešena pouze na síťových prvcích uživatelů a na přístupovém bodu k vnější síti ze strany poskytovatele.

Nevýhody

- Uživatelé provozující IPv4 jsou za NATem.
- Komunikace přes tunel je vždy méně efektivní (je nutné přenést více dat, pakety protokolu IPv4 se musí dělit na menší části než jsou zabaleny do IPv6 hlavičky, atd...)

TEREDO

Technologie TEREDO umožňuje stanici v síti IPv4 komunikovat přes IPv6. [33] Jedná se v podstatě o tunel, který propojí konkrétní stanici v IPv4 s IPv6 sítí. Takové spojení je naznačeno na obrázku **??** a probíhá v následujících krocích:

1. Stanice P kontaktuje s využitím IPv4 TEREDO server, který mu sdělí adresu místa pro vytvoření tunelu (TR = TEREDO relay).
2. Mezi P a TR se vytvoří spojení přes IPv4 TEREDO tunel.
3. Veškeré pakety IPv6 od P se přenesou prostřednictvím tunelu k TR.
4. TR zahájí směrování IPv6 paketů od P k cíli (IPv6 host) prostřednictvím své připojené (native) IPv6 sítě.
5. Jakmile přijde odpověď od cíle k TR, je tato dále přeposlána k P opět prostřednictvím vytvořeného tunelu.

Výhody

- Poskytuje plnou funkcionalitu IPv6 i pro stanice v síti s IPv4
- Mechanismus je zcela transparentní a nevyžaduje žádnou dodatečnou konfiguraci od poskytovatele připojení

Nevýhody

- Komunikace přes tunel je vždy méně efektivní (přenáší se více data, data je nutné dělit na menší části)
- Vyžaduje existenci existenci TEREDO serverů, které mohou být uzkým hrdlem při komunikaci

Obrázek 4.31: Logo IPv6 eady

Certifikace zařízení IPv6

Pro úplné ukončení přechodu na IPv6 je nutné, aby byl podporovaný na všech síťových prvcích. Přechod neznamená jen doimplementovat podporu IPv6 v operačních systémech, ale je nutné přidat i přidružené služby jako jsou ICMv6 a DNS. Návrh protokolu je však natolik komplexní, že obsahuje i služby, které není nutné implementovat vždy (např. DHCPv6). Pro rozlišení, jaké IPv6 služby síťový prvek nabízí, byly zavedeny certifikáty, které se udělují síťovým prvkům na základě toho, kolik vyžadovaných služeb mají implementovaných. Sadu testů a následný certifikát poskytuje organizace pojmenovaná jako IPv6 Ready Logo. [34] Po získání certifikátu může výrobce opatřit svůj produkt logem IPv6 Ready, viz obr. 4.31.

4.7.6 Mobilita

V případě, že jakékoliv mobilní zařízení v síti musí být stále dostupné a zároveň toto zařízení cestuje z jednoho segmentu sítě do jiného, lze zajistit, aby toto zařízení mělo stále stejnou adresu bez ohledu na to, kde se nachází. V principu to znamená, že pokaždé, když zařízení změní síťový segment, dostane novou adresu s prefixem specifickým pro daný segment. Zároveň o této nové adrese informuje tzv. domácáího agenta (home agent) což je služba, která vytváří vazbu mezi originální a nově přidělenou adresou. Pokud přijde komunikace na originální adresu, home agent zajistí přeposlání komunikace na aktuální adresu zařízení. Tento mechanizmus lze uplatnit nejen v lokálních sítích, ale i na globální úrovni v rámci IPv6. [35]

4.7.7 Fragmentace

Pokud nastane stejná situace, jak je popsána v 4.4.2, musí protokol IPv6 také provést fragmentaci. Rozdílem je, že fragmentace je uz prováděna pouze odesílatelem. Mezilehlé směrovače fragmentaci neprovadejí. Pokud se data paketu nevejdou do rámce pro následující segment, pošle se odesílateli zpráva Packet Too Big, kterou mu sdělí, jak velké fragmenty jsou povoleny. Tento proces se nazývá Path MTU Discovery. [36]

4.8 Typy adres IPv4 vs IPv6

Logické adresy lze rozdelit do čtyř skupin - unicast, multicast, anycast a broadcast. Pojetí těchto pojmů se mírně liší v obou implementacích IP protokolů.

4.8.1 Unicast adresy

Každá unicast adresa představuje vždy jednoho konkrétního příjemce. Tento typ adresy je podporován oběma IP protokoly.

4.8.2 Multicast adresy

V případě multicastových adres může jedna adresa zastupovat jedno a více zařízeních v síti. Zpráva poslaná na takovou adresu je doručena všem příjemcům. IPv6 obsahuje multicastovou komunikaci přímo v návrhu, ale bohužel umožňuje komukovat pouze uvnitř stejného segmentu. IPv4 pouze vyhrazuje blok adres určených pro multicast komunikaci a samotný multicast vůbec neřeší. Multicastová komunikace mezi segmenty se řeší pomocí externího protokolu PIM[37]. Jako příklad použití lze uvést streamování televizních přenosů po IPTV.

4.8.3 Anycast adresy

Stejně jako u multicast adres, každá anycast adresa představuje jedno a více zařízení v síti, ale zpráva poslána na tuto adresu je doručena jen jednomu ze zařízení. Tento princip se používá například v situaci, kdy máme několik zařízení, které poskytují identické služby a je tedy jedno, na které zařízení se zpráva doručí. Anycast adresy slouží k transparentní rozdělení zátěže mezi zařízení stejného typu (např. DNS). Podporovány jsou oběma IP protokoly, což je dáno spíše tím, že anycast adresa závisí na implementaci směrovače a ne na protokolech samotných. Směrovače jsou pak zodpovědné za výběr příjemce zprávy např. na základě jeho odezvy.

4.8.4 Broadcast adresy

Broadcast adresy zastupují vždy všechna zařízení v segmentu. Používá se pouze v IPv4 a rozeznáváme dva typy:

- Directed broadcast - je dán kombinací IPv4 adresy a její masky. Např. 192.168.1.255/24 nebo 10.0.127.255/17. Je platný pro jeden konkrétní síťový segment a lze jej směrovat (zprávy s touto všesměrovou adresou lze posílat mezi jednotlivými segmenty).
- Limited broadcast - jedná se o jednu konkrétní adresu - 255.255.255.255. Tato adresa je platná vždy pro segment, ve kterém se nachází zařízení, které odešle paket s touto adresou jako cílem. Tato adresa se nesměruje, nelze tedy za pomoci této adresy komunikovat mimo aktuální segment.

Poznámka: Všesměrové adresy sice zaručují doručení zpráv cílovým zařízení, ale už neručí za to, že bude zpráva na cílovém zařízení přijata. Mnohé systémy nebo konkrétní služby tyto všesměrové zprávy ignorují, pokud nejsou nutné pro jejich funkčnost.

Poznámka: V IPv6 lze prakticky dosáhnout broadcast komunikace uvnitř segmentu použitím multicastu na FF02::1, na kterém musí poslouchat všechna IPv6 zařízení v síti.

Kapitola 5

Směrování v počítačových sítích

Snahou této kapitoly je seznámit čtenáře se všemi základními principy směrování, které se používají v moderních počítačových sítích. Budou vysvětleny nutné pojmy pro pochopení problematiky, principy směrování a jejich implementace prostřednictvím směrovacích protokolů.

Hlavním úkolem směrování (routingu) v počítačových sítí je nalezení cesty pro doručení dat mezi různými stanicemi, které se nachází v různých počítačových sítích. O přeposílání dat se starají směrovače (routery), které bývají obvykle zapojeny do rozvětvené hierarchické struktury. Směrovače přeposílají data na základě toho, kterému příjemci jsou data určena, tudíž dle cílové IP adresy. Pro učinění rozhodnutí, kterým směrem se data přepošlou, se využívají směrovací tabulky. Směrovací tabulky mohou být nastavovány správci sítí manuálně, pak se jedná o statické směrování. Statické směrování je konfiguračně poměrně náročné, málo odolné vůči výpadkům a tak se u rozsáhlejších sítí využívají protokoly, které vytvářejí směrovací tabulky automaticky. Tomuto procesu se říká dynamické směrování. Pokud jsou data přeposílána v rámci sítě, která spadá pod jeden společný adresní rozsah, jedná se o tzv. vnitřní směrování. Typickým příkladem je např. síť jednoho poskytovale Internet - Internet Service Provider (ISP), popřípadě lokální síť za jedním směrovačem.

Pro vnitřní dynamické směrování se běžně využívají protokoly Open Shortest Path First (OSPF) nebo Routing Information Protocol (RIP), které budou představeny v další části této kapitoly. V případě, že jsou data šířena mezi různými sítěmi s odlišnými adresními rozsahy, jedná se o vnější směrování. Typickým příkladem je směrování mezi různými sítěmi v Internetu. Nejběžnějším protokolem, který se používá pro tento typ směrování je Border Gateway Protocol (BGP) a jeho princip se od protokolů pro vnitřní směrování zásadně liší. Veškeré do této chvíle zmíněné směrovací mechanismy ovšem předpokládají, že směrovací tabulky se vytváří dříve, než dochází k odesílání dat. Tomuto přístupu se říká proaktivní směrování a je vhodný pro statické sítě neměnící svoji topologii, typicky např. optické či metalické sítě. Pro sítě, které se často mění (např. mobilní bezdrátové či ad-hoc sítě), se ovšem proaktivní přístup nehodí, neboť než by došlo k sestavení směrovacích tabulek, byly by záznamy v nich obsažené již neaktuální a proto se využívá principu vytváření cesty těsně před vysíláním dat - tzv. reaktivní směrování.

5.1 Základy směrování

5.1.1 Proces směrování

V následujícím textu se předpokládá, že směrování probíhá v běžných paketových sítích. O paketech a jejich doručování jsme hovořili v kapitole č. 4. Směrování paketu začíná jeho příchodem na vstupní rozhraní směrovače. Po přijetí paketu dochází k jeho zpracování, tj. načtení jeho zdrojové a cílové IP adresy. Verze použitého IP protokolu není důležitá, proces směrování probíhá vždy stejně. Po načtení zdrojové a cílové IP adresy paketu dochází k jeho směrování.

Směrování paketu je proces, při kterém směrovač na základě zdrojové a cílové IP adresy obsažené v hlavičce paketu, vybírá způsob a odchozí rozhraní pro jeho přeposlání.

Způsob přeposlání může být dvojího typu buď přímý anebo nepřímý. Přímým způsobem se myslí finální přeposlání paketu k cílové stanici prostřednictvím linkové vrstvy, což je možné pouze za předpokladu, že směrovač a koncová stanice leží ve stejné síti. Více informací o doručování dat na úrovni linkové vrstvy lze nalézt v kapitole č. 3. Nepřímý způsob znamená přeposlání paketu sousednímu směrovači. Pokud je sousedních směrovačů více, je optimální sousední směrovač vybrán na základě předem zvoleného směrovacího kritéria (typicky clková vzdálenost k cílové stanici). Sousednímu směrovači je paket doručen prostřednictvím linkové vrstvy. Nepřímý způsob přeposlání paketu se mezi směrovači rekurzivně opakuje tak dlouho, dokud není možné paket doručit přímo cílové stanici.

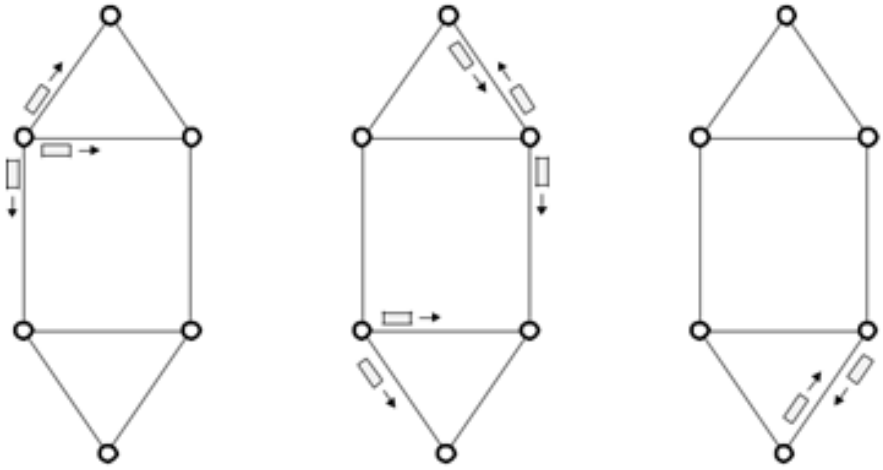
5.1.2 Záplavové směrování

Záplavové směrování je základem řady směrovacích protokolů, které potřebují zajistit doručování dat všem stanicím, které se v síti nachází, popř. všude tam, kde nelze z využívat směrovací tabulky (např. je-li topologie sítě mobilní).

Algoritmus záplavového směrování je následující:

- Záplavové směrování začíná ve zvoleném počátečním uzlu.
- Počáteční uzel osloví všechny svoje sousední uzly s požadavkem na směrování. Požadavek je opatřen identifikátorem (ID), který má za úkol zabránit vytváření smyček během směrování. Důvodem použití ID je potřeba odlišit to, že záplava může být spuštěna z různých uzlů najednou.
- Každý uzel, který obdržel požadavek na směrování nejprve zkontroluje, zda již dříve obdržel požadavek s tímto ID. Pokud ano, požadavek zahodí. V opačném případě provede přeposlání požadavku všem sousedům s výjimkou toho, od kterého požadavek přišel.
- Přeposílání sousedům se opakuje až do doby, kdy jsou dosaženy všechny uzly grafu.

Záplavové směrování je nejjednodušším směrovacím postupem, který má navíc tu příjemnou vlastnost, že vždy využije nejkratší cestu. Uzel přijatý paket (pokud není adresován jemu) rozešle do všech linek s výjimkou té, z níž paket přijal. Nebere přitom ohled na jakoukoliv další informaci o adrese adresáta, topologii sítě nebo vlastní stav.



Obrázek 5.1: Princip záplavového směrování

Výhodou metody je vysoká spolehlivost, nevýhodou je nadbytečné zatížení sítě kopiemi paketu a nutnost kopie likvidovat. Zaplavové směrování je vhodné pro sítě s neznámou nebo proměnlivou topologií (vojenské aplikace) a v situacích, kdy je nutné rychle dopravit shodnou informaci všem uzlům.

5.1.3 Izolované směrování

Další z metod, která se opírá pouze o lokální stavovou informaci (a neberou v úvahu informace, které mají ostatní uzly sítě), a které označujeme ji jako metodu izolovanou je metoda "horkého bramboru".

Uzel sítě při rozhodování o směru, ve kterém odešle přijatý paket, bere v úvahu pouze délky front na výstupních linkách (a jejich kapacitu). Paket zařadí do nejkratší fronty (vylučujíc frontu ve směru, odkud byl paket přijat), tedy frontu, kde bude paket nejdříve odeslán (respektujeme různé délky paketů a kapacity linek). Metoda samotná nezaručuje doručení paketu v konečném čase, v praxi je užitečná jako součást kombinovaných metod (delta směrování, Transpac).

Zajímavou metodou je "metoda zpětného učení", u které se uzel při odhadu směru ne-jkratší cesty opírá o informace přenášené pakety přicházejícími z jednotlivých uzlů sítě. Pro funkci metody je podstatné, že každý paket přenáší jako služební informaci údaj o době svého dosavadního putování sítí (v podobě globálního časového údaje o čase odeslání, nebo v podobě počtu dosud navštívených uzlů). Uzly, kterými paket prochází, si údaj o trvání cesty poznamená-vají do tabulky, indexované odesílatelem uzlu, spolu s údajem o směru, ze kterého paket přišel. Poznámku upravují při zjištění, že přicházející paket prošel cestou kratší, než byla cesta dosud zaznamenaná. Vytvářená tabulka odpovídá směrovací tabulce v dále uváděných metodách.

Nepříjemností metody je, že je "optimistická", nereaguje na zhoršení situace v síti, nebo je její reakce velmi pomalá. Pro praktické použití je nutné ji doplnit o "zapomínání", např. ovlivňováním poznámek v tabulce pakety, které prošly delší cestou. Se zjednodušenou

formou metody se setkáme při směrování v lokálních sítích propojených mosty.

5.1.4 Směrování řízené směrovací tabulkou

Směrovač provádí po přijetí paketu směrovací rozhodnutí. Pokud nelze paket přeposlat přímo cílové stanici, je nutné zvolit k přeposlání paketu optimální sousední směrovač. Způsob přeposlání paketu popř. výběr optimálního směrovače se typicky provádí srovnáním cílové IP adresy obsažené v hlavičce paketu s cílovou sítí všech směrovacích záznamů obsažených ve směrovací tabulce. Pokud cílová IP adresa patří do více cílových sítí obsažených v různých směrovacích záznamech, zvolí se pro směrovací záznam ten, který je nejvýhodnější, tj. má nejnižší cenu. Cena linky se běžně označuje jako metrika. Způsob výpočtu ceny cesty se mezi různými směrovacími algoritmy zásadně liší, neboť požadavky na směrování mohou být v různých situacích diametrálně odlišné.

Směrovací záznam je uspořádaná čtveřice {cílová síť, sousední směrovač, síťové rozhraní, metrika}.

Směrovací tabulka je množina směrovacích záznamů, kterou směrovač využívá k provádění směrovacích rozhodnutí.

5.1.5 Statické směrování

Statické směrování je směrování řízené směrovací směrovací tabulkou. Směrovací tabulka je konfigurována manuálně před začátkem směrovacího procesu.

Typické operace ve směrovací tabulce:

- Přidání záznamu.
- Odebrání záznamu.
- Změna záznamu (např. sousední směrovač nebo metrika).

5.2 Směrovací algoritmy

V následujícím přehledu budou zmíněny základní matematické algoritmy, které jsou implementovány ve směrovacích protokolech určených pro dnynamické směrování v moderních počítačových sítích.

5.2.1 Dijkstrův

Nejznámějším algoritmem nejkratší cesty je asi algoritmus známý jako Dijkstrův algoritmus. Opírá o ohodnocení hran grafu nezápornou funkcí l(u, v) a nalezne pro zvolený uzel s grafu G strom nejkratších cest vedoucích ze všech ostatních uzlů grafu G do uzlu s.

1. Každý z uzlů grafu označíme trojicí hodnot (L, P, D). Hodnota L udává dosud zjištěnou vzdálenost k uzlu s, hodnota D udává index (označení) uzlu, který je

sousedem na nejkratší dosud zjištěné cestě k uzlu s. Hodnota P reprezentuje fakt, že získané údaje L a D se už v dalším výpočtu nebudou měnit. Výpočet zahájíme s hodnotami:

L(s) = 0, D(s) = s, P(s) = 1 pro uzel s.

L(v) = 1, D(v) = v, P(v) = 0 pro uzly v $\neq$ s.

Uzel s si pro další krok označíme jako uzel u.

2. Pro každý uzel v, pro který je P(v) = 0 vypočteme hodnotu M = min{L(v), L(u)+l(u,v)} Pokud platí M < L(v) změníme označení uzlu v na L(v) = M, D(v) = u.

3. Mezi uzly v, pro které P (v) = 0 vybereme uzel s minimální hodnotou L(v). Je-li takových uzlů více, zvolíme ten, pro který je cesta do uzlu s nejkratší i z hlediska počtu hran. Pro tento uzel, který pro příští krok bude naším uzlem u, změníme hodnotu P(u) na P(u) = 1.

4. Pokud v grafu zbývá alespoň jeden uzel s hodnotou P(v) = 0, vrátíme se k bodu 2.

Použití algoritmu si budeme ilustrovat na obrázku 5.2, silně vytažené jsou hrany postupně vytvářeného stromu nejkratších cest s kořenem s. Algoritmus poskytuje jeden řádek směrovací tabulky pro každý uzel grafu G. Abychom zkonstruovali úplné tabulky, musíme Dijkstrův algoritmus spustit pro každý z uzlů grafu jako jeho kořen.

5.2.2 Bellman-Fordův / Ford-Fulkersonův

Malou modifikací Dijkstrova algoritmu získáme symetrický distribuovaný algoritmus vytvářející strom nejkratších cest vedoucích do uzlu s v grafu G.

Fáze algoritmu jsou následující:

1. Pro každý z uzlů grafu udržujeme dvojici hodnot (L, D). Hodnota L udává dosud zjištěnou vzdálenost k uzlu s, hodnota D udává index (označení) uzlu, který je sousedem na nejkratší dosud zjištěné cestě k uzlu s. Výpočet zahájíme s hodnotami:

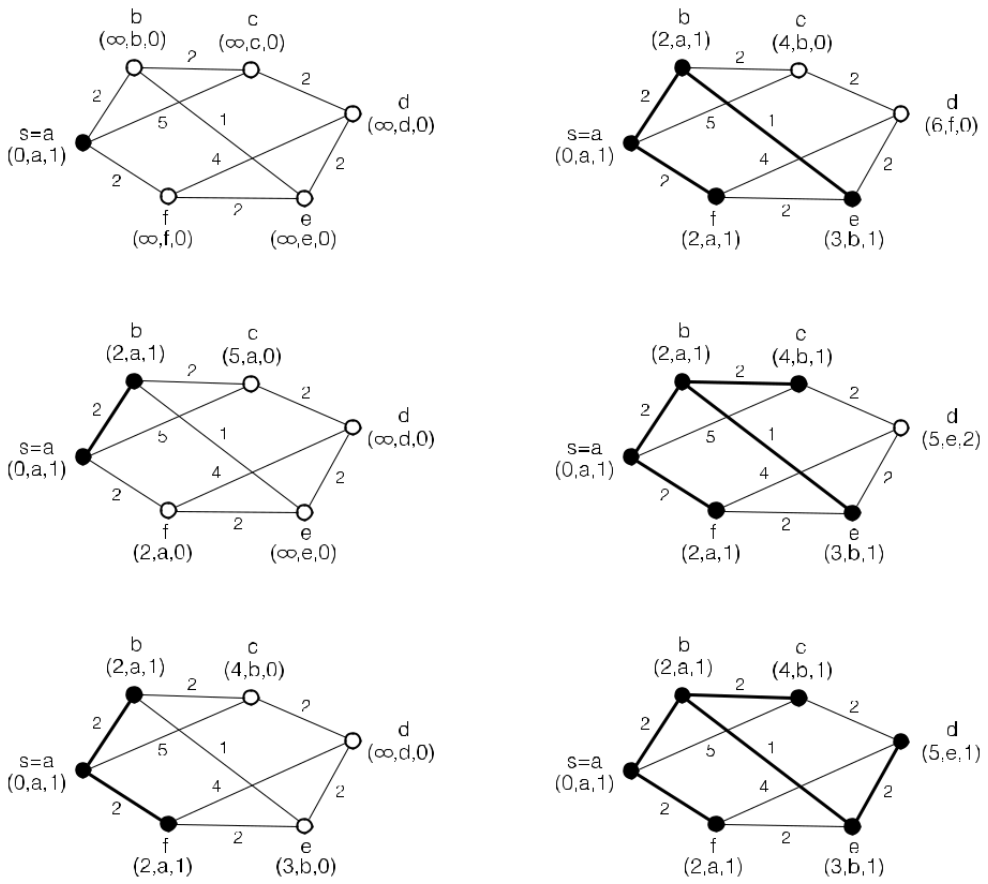
   L(s) = 0, D(s) = s pro uzel s

   L(v) = ∞, D(v) = v pro uzly v $\neq$ s.

2. Pro každý uzel grafu v 2 G vypočteme hodnotu M(v) = min{L(v), L(u)+l(u,v)}, $u \in G$.

3. Pro každý uzel, pro který platí M(v) < L(v) změníme označení na L(v) = M(v), D(v) = u, kde u je označení sousedního uzlu, pro který bylo zjištěno minimum.

4. Pokud bylo změněno označení u alespoň jednoho uzlu, vrátíme se k bodu 2.



Obrázek 5.2: Aplikace Dijkstrova algoritmu nejkratší cesty

5.2.3 Algoritmus pro určení maximálního toku

Pro síť s konečnou kapacitou linek může být zajímavé znát, jak velké informační toky lze sítí přenášet. Odpověď na takovou otázku dává algoritmus maximálního toku. Algoritmus je formulován pro orientovaný graf, jehož hrany jsou ohodnoceny kapacitami. Výsledkem je maximální tok, který lze přenést mezi dvěma zvolenými uzly.

1. Máme orientovaný graf, jehož hrany jsou ohodnocené kapacitou. Každé hraně připíšeme další ohodnocení, velikost toku, který hranou prochází. Koncové uzly, pro které budeme vyhodnocovat maximální tok si označíme jako s (vstup toku sítě) a t (výstup toku ze sítě). Algoritmus startujeme s nulovým tokem z uzlu s do uzlu t.

2. Nalezneme nejkratší cestu z uzlu s do uzlu t, přičemž bereme v úvahu pouze hrany, které a)jsou orientovány ve směru cesty a jejich kapacita je větší než dosud procházející tok, b)jsou orientovány proti směru cesty a prochází jimi nenulový tok.

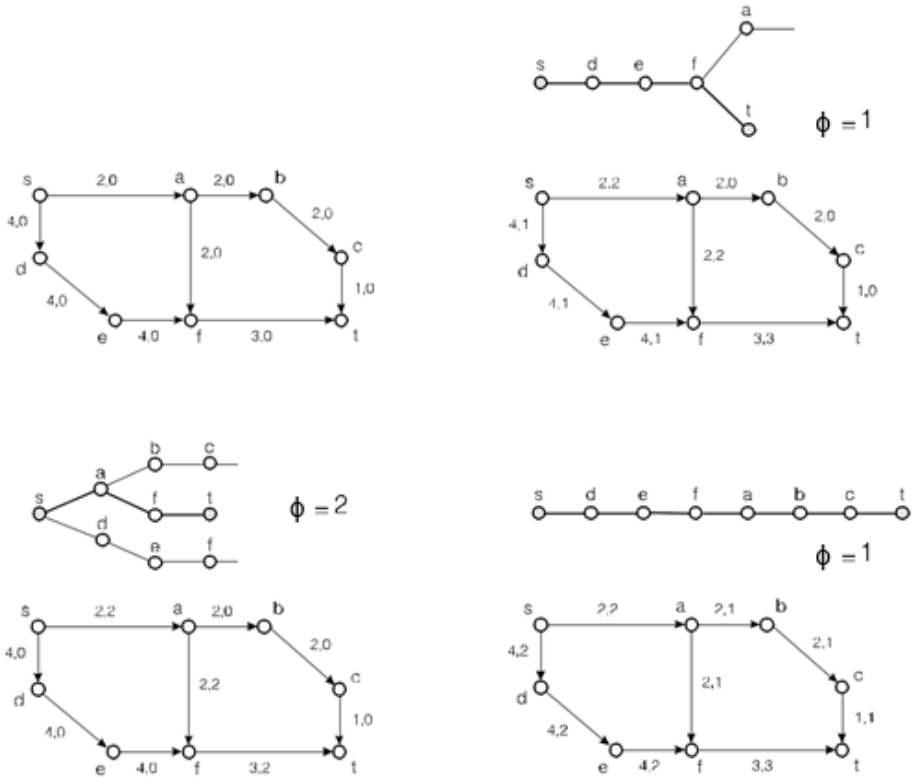
3. Pokud taková cesta neexistuje, nelze již k toku, který dosud mezi uzly s a t teče, přidat žádný tok. Dosavadní tok je hledaným tokem maximálním. Pokud vhodnou cestu nalezneme, najdeme

   a) minimum z P rozdílů kapacit a dosavadního toku pro hrany orientované ve směru cesty

   b) dosavadního toku pro hrany orientované proti směru cesty.

4. Proložený tok přičteme k dosavadnímu toku mezi uzly s a t a vrátíme se k bodu 2.

Postupné přidávání toku nejkratším cestám nalezeným v bodě 2 algoritmu ilustruje obrázek 5.3.



Obrázek 5.3: Určení maximálního toku

Hrany jsou ohodnoceny dvojicemi (kapacita, dosavadní tok). Schémata u jednotlivých kroků popisují postup vyhledávání nejkratší cesty, které lze ještě nějaký tok přidat. Poslední krok našeho příkladu odpovídá využití opačně orientované hrany na cestě mezi koncovými uzly.

5.3 Dynamické směrování

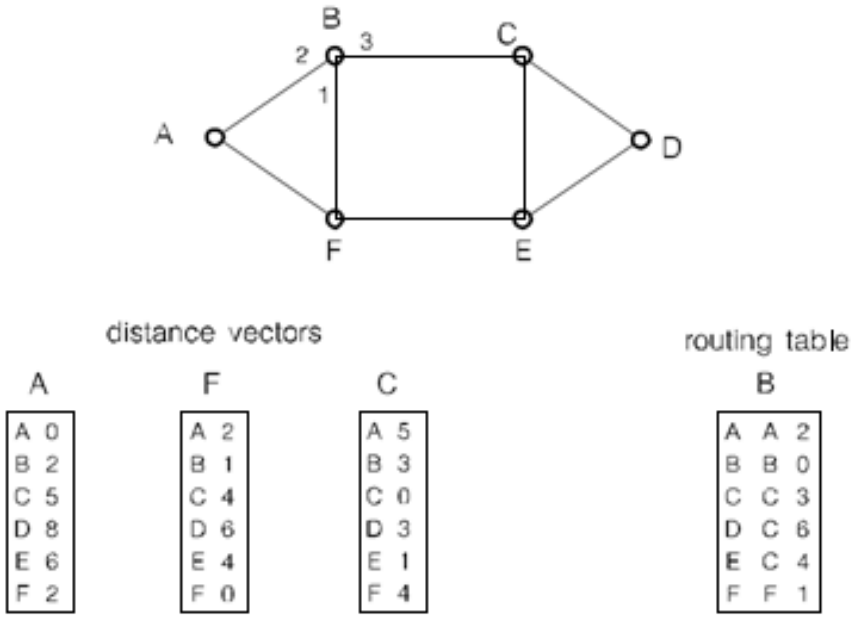
Statické směrování je směrování řízené směrovací směrovací tabulkou. Směrovací tabulka je konfigurována dynamicky s využitím směrovacích protokolů před začátkem směrovacího procesu.

5.3.1 RIP protokol

Ford-Fulkersonův algoritmus, který jsme si uvedli dříve lze přímočaře implementovat v síti propojených počítačů. Krok 2 algoritmu odpovídá výměně dosud zjištěné hodnoty L se sousedy. V praxi budeme pracovat s vektorem vzdáleností ke všem uzlům sítě, algoritmus

vytváří přímo směrovací tabulky. Pod jménem Distance-Vector algoritmus je využíván v jednodušších autonomních systémech Internetu (RIP - Routing Information Protocol).

Pro výpočet směrovacích tabulek se s výhodou používá distribuovaného výpočtu. Uzly si v pravidelných intervalech vyměňují informace (odhady) o svých vzdálenostech k ostatním uzlům sítě (obr.). Kombinací přijatých vektorů vzdáleností a znalosti o vzdálenostech sousedů lze vybudovat směrovací tabulky. Přestože je perioda výměn poměrně dlouhá, může být režie mechanismu pro sítě s velkým počtem uzlů a pomalé linky mezi uzly příliš vysoká.

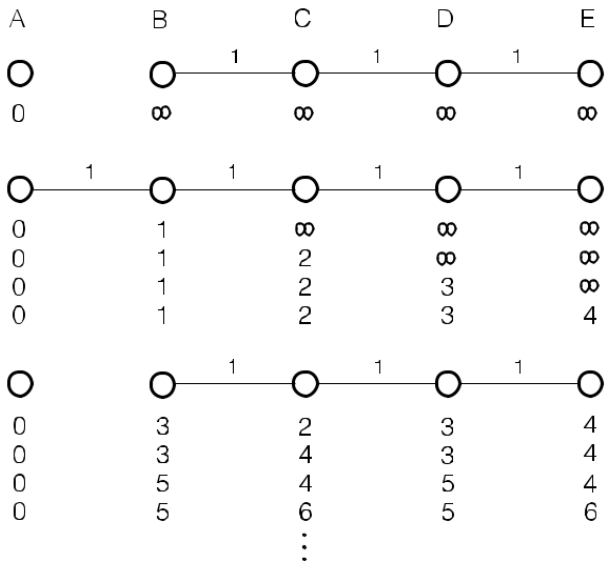


Obrázek 5.4: Distance-vector algoritmus pro výpočet směrovacích tabulek

Neupravený algoritmus má nepříjemnou vlastnost, reaguje rychle na "dobré zprávy"ale pomalu na "špatné zprávy", navíc může vyvolat směrování paketů v cyklech. Problém lze ilustrovat na následujícím příkladě, který uvádí změny vzdáleností (určených tímto algoritmem) k uzlu A od ostatních uzlů sítě po připojení uzlu A k síti a po jeho odpojení od sítě (obr. 5.5).

Ke zjištění, že uzel A není propojen se zbývajícími uzly sítě lze dospět rychleji při vhodné volbě "nekonečna". Algoritmus RIP sítě Internet považuje za "nekonečno"hodnotu 15 (hran mezi uzly).

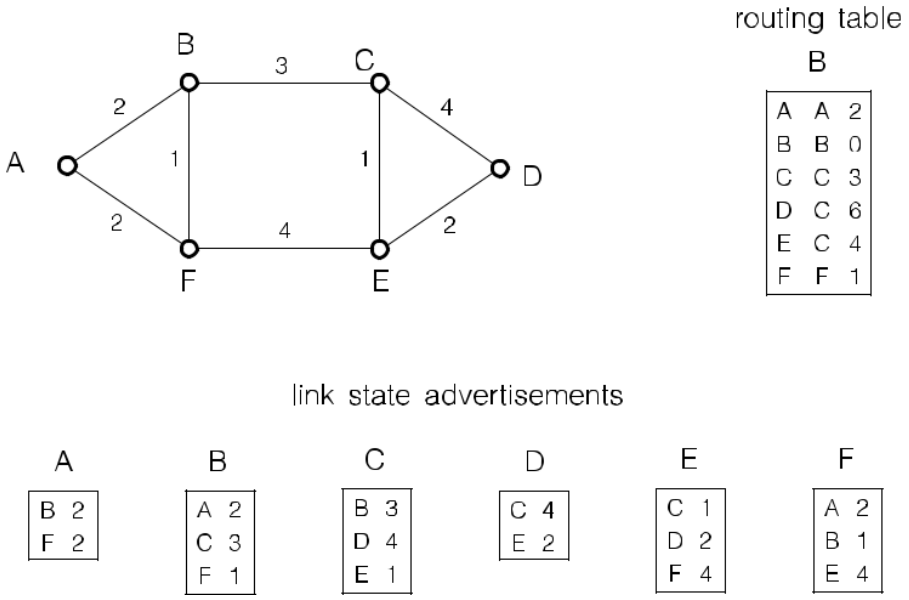
Do okamžiku, než všechny uzly zjistí výpadek linky (dopočítají do patnácti), může docházet k cyklickému směrování paketů (v našem příkladě například mezi uzly B a C). Zjištění výpadku linky lze urychlit, pokud uzel C, který získal informaci o nejkratší cestě k uzlu A od uzlu B, svůj údaj uzlu B zpět nepředá (tato úprava algoritmu je označována jako split horizon). Dalšího urychlení lze dosáhnout, pokud uzel C předává uzlu B v takovém případě údaj o nekonečné vzdálenosti (tato úprave je označována jako inverse poisson). Nekonečné vzdálenosti s uzlem A nebude uzlu C indikovat pokud uzel nepředává informaci o vzdálenosti při ztrátě spojení se lze bránit například "setrvačností"směrovacího algoritmu, obrácení smyslu směrování po lince dovolíme až po určitém počtu kroků.



Obrázek 5.5: Reakce na připojení a odpojení uzlu A

5.3.2 Směrování řízené stavem linky

Velice výhodnou metodou výpočtu směrovacích tabulek je postup opírající se o broadcastem šířené informace o stavech linek, s nimiž jednotlivé uzly sítě sousedí. Takové informace (LSA = Link State Advertisement), které si uzly ukládají ve směrovací databázi, dovolí jednotlivým uzlům vytvořit si úplný obraz sítě (obr. 5.6). Metoda je základem postupu OSPF (Open Shortest Path First), který je široce využíván v Internetu; pro výpočet směrovací tabulky lze s výhodou použít Dijkstrova algoritmu.



Obrázek 5.6: Link-state algoritmus pro výpočet směrovacích tabulek

5.3.3 OSPF protokol

Určitým problémem distribuce LSA paketů je nutnost zaručit jejich příjem ve správném pořadí. Toho lze dosáhnout jejich číslováním, které dovolí starší pakety ignorovat. Ignorování starších LSA paketů přináší jiný problém: tím je restart směrovače, po něm by mohly být jím rozesílané LSA pakety ignorovány. Tomu se sice lze bránit omezením časové platnosti rozesílaných údajů a mechanismem, který dovolí jejich rychlé smazání v celé síti (rozesláním LSA paketů s nulovou časovou platností); dosáhnout však rozumné rovnováhy mezi rychlostí reakce na výpadek uzlu (teprve po vypršení časového limitu smí směrovač znovu zahájit provoz) a periodou opakování LSA paketů může být pro rozsáhlé sítě obtížné.

Restart směrovače lze detekovat, pokud zajistíme, že číslování začíná s hodnotami, ke kterým se již při modulo inkrementaci nevracíme (např. startujeme s nejzápornější hodnotou, ale po překročení největší kladné hodnoty se vracíme k nule. Libovolný směrovač v síti, který takovou situaci detekuje rozešle LSA pakety s nulovou časovou platností a tím současně informuje restartovaný směrovač o posledním použitém čísle LSA paketu.

Mechanismus rozesílání LSA paketů dovoluje zrekonstruovat informace o topologii sítě i po jejím rozpadu na izolované části a pozdějším propojení (případně po propojení dosud izolovaných sítí). Pro urychlení je doplněn o výměnu informací ze směrovacích databází po propojení izolovaných částí.

Detekci provozuschopnosti jednotlivých linek, o kterou se opírá rozesílání LSA paketů, podporuje jednoduchý protokol Hello. Jím se sousední směrovače vzájemně informují o své vlastní provozuschopnosti a o provozuschopnosti spoje mezi nimi. Případné zvláštní situace, které může způsobit selhání protokolu Hello (například vysílání Hello paketů s chybnou adresou odesílatele, které může zakrýt výpadek jiného směrovače) lze ošetřit stárnutím údajů ve směrovacích databázích.

Konečně, protokol OSPF podporuje dvouúrovňový hierarchický model sítě; rozsáhlejší sítě lze pak složit z více sítí vzájemně propojených sítí páteřních.

5.4 Vnější směrování

Směrovací protokoly, které byly popsány v předchozích sekcích, se používají výhradně pro vnitřní směrování a jsou řízeny směrovacími tabulkami. Čím větší jsou směrovací tabulky, tím je náročnější jejich údržba a aktualizace. Dynamické změny jednotlivých záznamů ve směrovacích tabulkách v rozlehlých sítích (Internetu) by znamenaly značnou režii a byly by neefektivní. Z tohoto důvodu se používá pro směrování v Internetu jiný princip a sice vnější směrování.

5.4.1 Autonomní systémy

Každý subjekt (poskytovatel Internetu) má přidělenou pod svou správu určitou množinu IP prefixů , které určují všechny IP adresy, které mohou být v jeho síti využívány. Skupina těchto IP prefixů a směrovačů, které se o směrování starají je sdružena pod jedním

společným označením - autonomní systém (AS). Každý autonomní systém má svůj unikátní 16ti resp. 32ti bitový identifikátor (ID), který se dále využívá ve směrování např. v protokolu BGP.

Autonomní systémy jsou mezi sebou navzájem propojené prostřednictvím hraničních (Border) směrovačů, které směrování zajišťují. Provoz z vnitřní sítě, který opouští daný autonomní systém prochází výhradně přes hraniční směrovače.

Dnešní kompletní směrovací tabulky pro síťové IP prefixy zahrnující celý Internet obsahují statisíce směrovacích záznamů.

5.4.2 Směrování založené na vektoru cesty

Vektor cesty (PATH) znamená posloupnost ID autonomních systémů, která se nachází mezi zdrojovou a cílovou stanicí. Hlavním problémem při vytváření směrovacích tabulek z PATH je namapování konkrétních IP adres na ID příslušného autonomního systému. Tato operace je však technicky triviální, neboť organizace, které se starají o správu adresních IP rozsahů pro určitý kontinent (např. v Evropě RIPE) takovýto seznam mapování (IP na AS) standardně poskytují. Informace o mapování IP rozsahů na ID autonomních systémů se musí (na směrovačích) denně aktualizovat, neboť k fluktuacím IP adresního prostoru dochází poměrně často.

Základem vytváření směrovacích tabulek na konkrétním hraničním směrovači je vzájemná výměna vektorů cest s okolním hraničními směrovači. Pro všechny vektory cesty jsou známé optimální sousední hraniční směrovače, které daný vektor cest poskytly a hraniční směrovače tak plní funkce bran. Vektor cest obsahuje úplnou informaci o tom, kterými autonomními systémy musí daný paket od zdroje k cíli projít. Pro konkrétní PATH se určuje jeho metrika. Určení metriky z PATH v protokolu BGP je v porovnání s RIP či OSPF mnohem složitější a samotná metrika je vyjádřena pomocí vektoru (i více než 10) hodnot.

Vlastní proces směrován paketů je již zcela shodný se všemi protokoly, které jsou řízené směrovacími tabulkami.

5.4.3 BGP protokol

Směrovací protokol BGP je dnes brán jako základní protokol směrování v Internetu a v porovnání se směrovacími protokoly RIP a OSPF je mnohem komplexnější. Dnes používaná verze protokolu BGP je verze 4, která obsahuje specifikaci pro IPv6. Kompletní specifikace BGP-4 je popsána v RFC 1771. RFC 6793 obsahuje specifikaci, která deklaruje rozšíření ID z 16 bitů na 32.

BGP protokol se dá používat jak pro vnější (EBGP) i vnitřní (IBGP) směrování, nicméně jeho primární využití je pro vnější směrování.

Protokol BGP definuje i pojem konfederace (confederation) autonomních systémů. Konfederace je shluk více autonomních systémů pod jedinou společnou jednotku. Použití konfederací umožňuje zkrácení délky vektoru cesty, avšak za cenu méně přesné informace o tom, kudy pakety procházejí.

Netransientní a trasientní autonomní systém

Jak již bylo nastíněno, dnešní Internet je množina vzájemně propojených autonomních systémů. Vzhledem k takovému uspořádání je nutné, aby pakety z určitých autonomních systémů procházely přes jiné autonomní systémy. Tato skutečnost znamená to, že některé autonomní systémy zajišťují dopravu paketů pro jiné. Doprava cizích paketů způsobuje pro tyto autonomní systémy dodatečné zatížení páteřních linek. Pokud autonomní systém umožňuje dopravu paketů pro jiné autonomní systémy, hovoříme o transientním autonomním systému a v opačném případě o netransientním autonomním systému.

Single-homed a multi-homed autonomní systém

Pokud je možno dosáhnout konkrétní autonomní systém pouze přes jediný sousední autonomní systém, hovoříme o tzv. single-homed autonomním systému, v opačném případě hovoříme o multi-homed autonomním systému.

5.4.4 Možnosti směrování v BGP protokolu

Metrika pro směrování, která je používána směrovacími algoritmy RIP a OSPF, souvisí úzce pouze se stavem či kapacitou konkrétní linky. V praxi se poměrně často stává, že kritérií určujících optimálnost směrování je více a mají různé váhy. Protokol BGP určuje a šíří pro každý vektor cesty několik různých atributů (viz tabulka níže 5.4.4), které slouží k ohodnocení dané cesty. Atributy z hlediska důležitosti mají různé avšak striktně definované váhy. Pokud existují alespoň dva vektory cest směřující do stejného cílového AS, dochází ke srovnání hodnot všech atributů dle jejich vah (nižší hodnota znamená vyšší váhu). Jako lepší se vybere vektor cest, který vykazuje lepší hodnotu atributu s vyšší váhou.

Atributy vektorů cest mohou být navíc čtyř různých typů, jedná se o tyto:

- (A) známé a povinné (well-known mandatory): musí být povinně připojeny ke každé cestě a všechny implementace BGP je musí podporovat.
- (B) známé a nepovinné (well-known discretionary): všechny implementace BGP je musí podporovat, avšak jejich šíření není povinné.
- (C) volitelné a povinně předávatelné (optional transitive): konkrétní implementace BGP je nemusí podporovat, avšak musí je předávat dále beze změny.
- (D) volitelné a nepovinně předávatelné (optional nontransitive): konkrétní implementace BGP je nemusí podporovat a ani nemusí je předávat dále.

5.5 Směrování v ad-hoc a mobilních sítích

Mobilní sítě se z hlediska směrování liší od klasických sítí tím, že jejich topologie se neustále mění. Algoritmy založené na směrovacích tabulkách jsou neúčinné, jelikož než dojde k plné

Tabulka 5.1: Přehled nejvíce používaných atributů a jejich význam v protokolu BGP

Váha	Typ	Název atributu	Význam atributu
1	A	WEIGHT	Umožňuje preferovat některou cestu v rámci všech směrovačů v rámci daného AS.
2	A	LOCAL_PREFERENCE	Umožňuje pro odchozí provoz preferovat cesty k nějaké síti, která je dostupná přes více alternativních linek.
3	-	-	Individuální preference směrovače, vlastní hledisko.
4	A	AS_PATH	Preferuje cesty, které mají kratší vektor PATH, tj. směrován proběhne přes méně autonomních systémů.
5	A	ORIGIN	Zohledňuje zdroj informace o směrování, odkud se vzala - např. z prostřednictvím vnitřního směrování v rámci AS (IBGP).
6	A	MED	Multi Exit Discriminator, ovlivňuje preferenci volby cesty sousedního AS, tzn. ovlivňuje příchozí provoz.
7	-	-	Preferuje cesty získané z vnějšího směrování (EBGP) před cestami z vnitřního směrování (IBGP).
8	A	NEXT_HOP	Preferuje cesty jdoucí přes menší počet směrovačů v rámci vlastního autonomního systému.
9	D	ROUTER_ID	Vybere se směrovač, který má nejvyšší hodnotu ID přidělenou pro jeho autonomní systém. Toto kritérium nemá žádný kvalitativní důvod, jeho přínos spočívá pouze v možnosti jednoznačného směrování.

aktualizaci směrovacích tabulek, je již topologie změněná a tím pádem směrovací tabulky již nemusí být aktuální.

Dříve uvedené směrovací protokoly RIP a OSPF patří mezi proaktivní směrovací protokoly. Proaktivní v tomto případě znamená, že směrovácí tabulky musí být sestaveny ještě před začátkem posílání dat. Mobilní sítě používají pro směrování algoritmy založené na hledání cest, které se začínají vytvářet až v okamžiku, kdy vznikne požadavek na posílání dat.

5.5.1 Reaktivní algoritmy

Reaktivní algoritmy jsou založeny na tom, že hledání cesty probíhá od zdrojové stanice (uzlu) prostřednictvím sousedních uzlů. Výsledkem reaktivních algoritmů je cesta nebo množina cest mezi zdrojovou a cílovou stanicí. Cesta existuje po dobu posílání dat a po té se zapomene. Pokud opětovně vznikne požadavek na hledání totožné cesty, musí se hledání opakovat.

Zmíněné algoritmy Dynamic Source Routing (DSR) a Ad hoc On-Demand Distance Vector Routing (AODV) vvcházejí z podobné myšlenky, liší se však ve způsobu vracení cesty a uchovávání směrovacích informací. Základní mechanismus, ze kterého tyto směrovací algoritmy vycházejí je záplavové směrování 5.1.2.

5.5.2 Dynamic Source Routing (DSR)

Algoritmus DSR začíná budovat cestu až v okamžiku, jakmile vznikne požadavek na směrování dat. Nutným požadavkem pro budování cesty je znalost síťové adresy (záleží na použitém adresačním mechanismu), která je nastavena na cílové stanici. Před zahájením hledání cesty je vygenerován unikátní identifikátor cesty (ID), který slouží jako ochrana proti zacyklení.

Algoritmus DSR rozlišuje následující druhy zpráv:

- Route Request (RR): Zpráva, která slouží na hledání cesty. RR obsahuje unikátní ID a adresy zdroje a cíle. Zpráva RR je propagována od postupně od zdroje k cíli celou sítí.
- Route Reply (RRP): Zpráva, která nese informace o nalezené cestě, tj. poslupnost všech uzlů přes které tato cesta vede. DSR může prostřednictvím RRP vracet cest více.
- Route Error (RE): Zpráva, která informuje zdroj o tom, že při posílání dat došlo k výpadku některého uzlu po cestě. Jakmile tuto zdrojový uzel obdrží musí začít používat pro posílání dat jinou cestu.

Algoritmus pro nalezení cesty je rekurzivní a poměrně jednoduchý a dá se popsat následujícími kroky:

1. Zdrojová stanice vygeneruje zprávu Route Request, kterou odešle všem svým sousedním uzlům.

2. Pokud některý z uzlů zprávu již někdy obdržel, což zjistí dle jejího ID, zahodí ji. V opačném případě ji přepošle všem svým sousedům s výjimkou uzlu, od kterého přišla.

3. Krok 2. se opakuje tak dlouho dokud se zpráva Route Request nedostane až k cílové stanici.

4. Jakmile cílová stanice obdrží zprávu Route Request, ve které je jako cílová adresa uvedna její adresa, vyšle cílová stanice zdrojové stanici zprávu RRP. Zpráva RRP se šíří prostřednictvím uzlů, díky kterým byla k cílové stanici doručena v obráceném pořadí. Vysílání zprávy Route Reply se opakuje tolikrát, kolikrát zpráva Route Request dorazí k cílové stanici. Maximální počet zpráv Route Request je roven počtu sousední uzlů cílové stanice.

5. Jakmile zdrojová stanice obdrží všechny odeslané Route Reply, vybere si cestu, která je nejkratší, tj. tvořena nejmenším počtem uzlů.

6. Po výběru cestu dochází k odesílání dat od zdrojové stanice k cílové.

5.5.3 Adhoc on Demand Distance Vector (AODV)

Alogritmus AODV se v principu podobá algoritmu DSR. Jeho hlavní odlišností oproti DSR je to, že vrací pouze jedinou cestu. Záznamy o nalezené cestě jsou dočasně udržovány v paměti jednotlivých uzlů, z čehož plyne název daného algoritmu. Algoritmus používá stejné druhy zpráv jako DSR, tj.RR, RRP a RE.

Algoritmus probíhá shodným způsobem jako u DSR, s tím rozdílem, že sousední uzly, které obdržely zprávu Route Request a budou tuto zprávu dále přeposílat, tuto skutečnost předchozím uzlům potvrdí. Jakmile zpráva Route Request doputuje k cíli, je vrácena obdobným způsobem nalezá cesta. Při vracení cesty zpět si jednotlivé uzly po nalezené cestě vytvoří dočasné směrovací tabulky, které se budou využívat po celou dobu odesílání dat.

Obecnou nevýhodou algoritmů DSR a AODV je to, že dochází k hledání cíle postupně prostřednictvím zaplavování, což má za následek, že zprávy Route Request se šíří napříč sítí i do oblasti, ve kterých cílová stanice neleží, což síť zbytečně zatěžuje. Algoritmus AODV se dnes běžně používá pro nalezení cesty v moderních mobilních sítích GSM.

Kapitola 6

Transportní vrstva

Služby síťové vrstvy by nám mohly připadat jako postačující k zabezpečení komunikace mezi stanice nacházejícími se v různých sítích. Přesto je nad tyto služby prakticky u všech síťových architektur přidávána vrstva další — vrstva transportní. Jejím hlavním úkolem je překrýt nepříjemné vlastnosti některých typů sítí a dát uživateli představu, že má k dispozici přenosový kanál odpovídající jeho konkrétním požadavkům, například v tom, že:

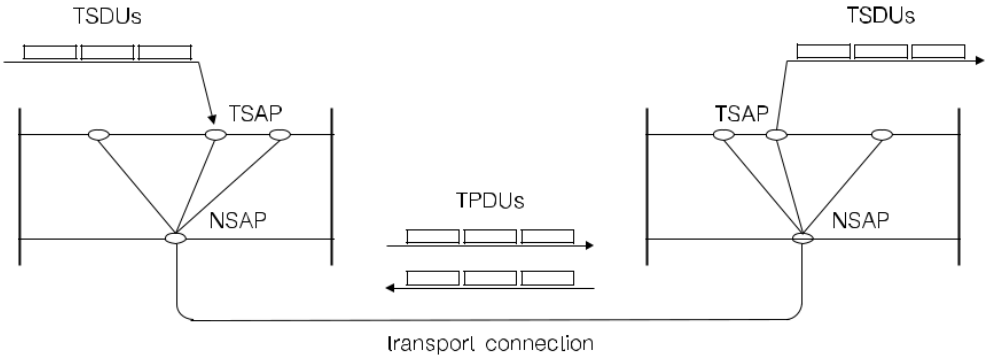
- data předává ve formátu definovaném aplikací, tedy jako zprávy, a ne ve formátu vyžádaném sítí, tedy jako pakety.
- doručuje aplikační zprávy nejen beze ztrát a duplikací, ale také v nezměněném pořadí.

Navíc transportní vrstva obvykle zajišťuje multiplex na úrovni aplikací — umožňuje nezávislou komunikaci více aplikacím na jednom počítači.

Úroveň síťových služeb v různých sítích je velice rozdílná. Veřejné datové sítě X.25 posky-tují virtuální kanály, zachovávají pořadí přenášených paketů a podporují koncové potvrzování. Lokální sítě z principu zachovávají pořadí paketů, ale koncové potvrzování většinou nemají. Chybovost kanálu, která vyvolává ztrátu paketů, je u nich velmi nízká. Naproti tomu data-gramové sítě, mezi které můžeme počítat Internet, mohou směrovat jednotlivé pakety různými cestami a narušit jejich pořadí. Pracují obvykle bez koncového potvrzování (strategie "best ef-fort", o vyjímečnosti ztráty paketu se u nich hovořit nedá. Základním úkolem transportní vrstvy je uvedené rozdíly zakrýt a dát uživateli dojem, že má k dispozici ideální komunikační kanál, který přenese všechny pakety bez chyby a v původním pořadí.

Transportní vrstva je obvykle označována jako ta nejvyšší ze základních síťových vrstev. Její služby jsou standardizovány, její rozhraní je snadno použitelné při psaní aplikací. Strukturu transportní vrstvy odpovídající normě modelu OSI/ISO si můžeme popsat na obrázku 6.1. Transportní vrstva poskytuje služby přenosu zpráv Transport Service Data Unit(TSDU) prostřednictvím transportních přístupových míst — socketů TSAP (Transport Service Access Point). Využívá přitom služeb síťové vrstvy, poskytovaných prostřednictvím síťových přístupových míst Network Service Access Point (NSAP). Pakety přenášející data a řídící informace označujeme jako transportní pakety Transport Protocol Data Unit (TPDU).

Transportní protokol TCP internetu poskytuje službu přenosu zpráv na socketech (BSD), přičemž data přenáší prostřednictvím síťového protokolu IP v transportních segmentech.



Obrázek 6.1: Transportní vrstva

Nadřazené vrstvy, ale i aplikace využívající přímo služeb transportní vrstvy, mají k dis-pozici primitiva, které dovolují otevřít transportní komunikační kanál s konkrétní úrovní posky-tovaných služeb, zajistit předávání dat po otevřeném kanálu a po ukončení datové komunikace kanál uzavřít.

6.1 Třídy transportních protokolů

Složitost implementace transportního protokolu závisí na vlastnostech síťové vrstvy. Pro potřeby porovnání dělíme síťové technologie z hlediska kvality poskytovaných služeb do tří tříd označo-vaných A, B a C.

Sítě třídy A zajišťují téměř perfektní komunikaci. Množství poškozených, ztracených nebo duplikovaných paketů je zanedbatelné, stejně jako je zanedbatelná potřeba navazovat přerušená spojení. Do této kategorie lze optimisticky zařadit pouze některé sítě lokální. Implementace transportní vrstvy je pro sítě třídy A velmi snadná a zahrnuje pouze rozklad zpráv na pakety, zpětné skládání paketů do zpráv a případný multiplex.

Přepojovací sítě s virtuálními kanály zajišťují bezchybový sekvenční přenos paketů, ale je občas nutné obnovit přerušené spojení (N-RESET). Při obnovování spojení může dojít ke ztrátě nebo duplikaci transportního paketu a je proto nutné mít vlastní potvrzování. Přepojovací sítě s virtuálními kanály zahrnujeme spolu s realisticky hodnocenými lokálními sítěmi do třídy B.

Nejkomplikovanější je implementace transportní vrstvy nad datagramovými sítěmi, u kterých je nutné zajistit koncové potvrzování a ukládání paketů přicházejících v porušeném pořadí. Tyto sítě označujeme jako sítě třídy C.

Podle třídy sítě, nad kterou transportní vrstva pracuje, a podle toho, zda jedno síťové spo-jení slouží pouze pro jediné spojení transportní nebo je vyžadován multiplex, zavádí norma OSI pět tříd transportních protokolů. Označuje je jako TP0 až TP4, jejich přehled uvádí následující tabulka:

V praxi se běžně používá protokol TP4, a to i tehdy, jestliže vlastnosti sítě tak

Tabulka 6.1: Třídní transportních protokolů

Protokol	Třída	Zajišťovaná funkce
TP0	A	-
TP1	B	koncové potvrzování
TP2	A	multiplex
TP3	B	koncové potvrzování a multiplex
TP4	C	koncové potvrzování, řazení a multiplex

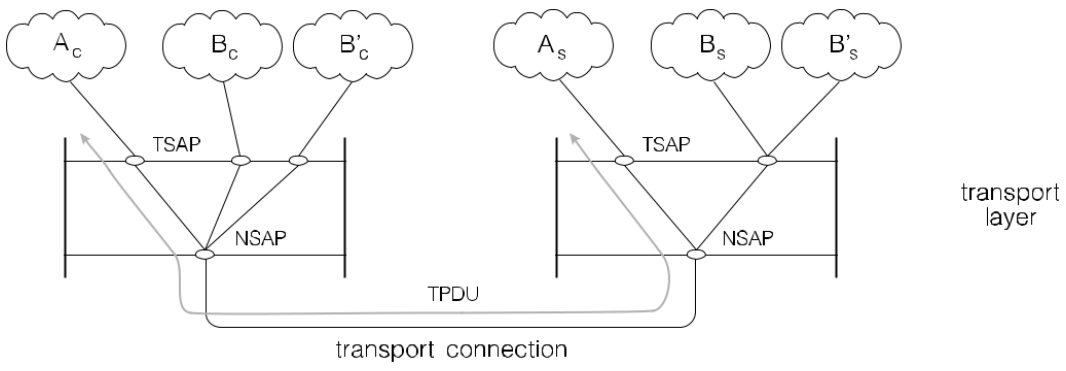
složitou obsluhu nevyžadují. Zvýšení zabezpečovací schopnost transportního protokolu se ve snížení jeho efektivity příliš neprojeví.

Transportní protokol TCP odpovídá svými možnostmi, s ohledem na vlastnosti síťového protokolu IP, transportnímu protokolu TP4 s tím, že ve své současné formě nepodporuje QoS.

6.2 Multiplex a adresace

Jako transportní multiplex označujeme schopnost transportní vrstvy vytvořit na jediném síťovém spojení mezi dvěma počítači více spojení transportních, tedy spojení mezi procesy, které na těchto počítačích běží. Důvodem pro toto řešení, označované jako multiplex směrem nahoru (upward multiplex), jsou u sítí s virtuálními kanály technické limity omezující počet síťových spojení, u sítí datagramových omezení adresace na adresu počítače (procesy nejsou na úrovni sítě rozlišitelné). Dalším důvodem mohou být tarify veřejné datové sítě znevýhodňující větší počet slabě využívaných virtuálních kanálů oproti menšímu množství sdílených kanálů s větší intenzitou provozu.

Pro rozlišení jednotlivých procesů využívajících jednotlivé kanály transportního multiplexu zavádíme adresaci přípojných míst TSAP v rámci počítače. Přípojné místo je jednoznačně identifikováno dvojicí (NSAP,TSAP), kde NSAP je adresa počítače (například odpovídající normě X.121 CCITT nebo adresaci Internetu).



Obrázek 6.2: Transportní spojení a multiplex

Tabulka 6.2: Položky transportních paketů

CR	Connection Request	CC	Connection Confirm
DR	Disconnect Request	DC	Disconnect Confirm
DT	Data Transfer	AK	Acknowledgement
ED	Expedited Data	EA	Expedited Acknowledgement
RJ	Reject	ER	Error

S jednoznačnou adresací procesů jsou spojeny ještě další problémy. Jedním z nich je nutnost znát kompletní adresy (ve tvaru (NSAP,TSAP)) pro všechny služby v síti, které chceme využívat. Částečným řešením je umístění standardních služeb na pevných adresách (např. služby telnet a ftp v Internetu najdeme vždy na socketu s pevnou adresou), další možností je služba name serveru na pevné adrese, která nám číslo socketu pro danou službu na požádání sdělí.

Další problém souvisí s potřebou provozovat souběžně více kopií téže služby na jednom systému. Pevné přidělení socketů kopiím službám by bylo omezující, východiskem je možnost souběžné činnosti více kopií služby (rodičovský proces navazuje komunikaci a synovské procesy zajišťují vlastní službu) na jednom socketu. Každé transportní spojení je pak jednoznacně identifikováno čtveřicí (N SAPc; T SAPc N SAPs; T SAPs). U protokolu TCP navíc tento údaj doplňujeme o pátý údaj, o identifikaci protokolu TCP.

Výjimečně se můžeme u transportní vrstvy setkat s multiplexem směrem dolů (downward multiplex). Toho se využívá, pokud pro jedno transportní spojení vyžadujeme větší přenosovou kapacitu, než je schopné poskytnout jedno spojení síťové. Kapacita jednotlivého virtuálního spoje je například omezena zpožděním při přenosu paketu sítí a okénkem koncového potvrzování. Dalším důvodem pro tento typ multiplexu může být zvýšení spolehlivosti transportního spojení, které multiplexujeme na dvě fyzicky oddělená spojení síťová.

6.3 Formáty transportních paketů

Pro zajištění své funkci si protokolové stanice transportní vrstvy vyměňují služební informace ve formátech datových transportních paketů (například číslování paketů a potvrzení) nebo jako speciální služební pakety. Jako příklad formátu transportního paketu si uvedeme formát paketu CR, kterým žádá stanice o otevření spojení. V jednotlivých polích je uvedena délka LI , jednoznačná identifikace spojení DST-REF a SRC-REF, požadovaná třída a parametry spojení, a případná aplikační data.

Protokol TCP používá jediný formát transportního segmentu, ten se používá jak pro vlastní spojení, tak pro přenos dat (obr. 6.7).

Řídicí funkce se opírají o příznaky SYN, ACK, URG, PSH, FIN a RST. Jejich kombinace dovolují vytvářet transportní segnenty pro jednotlivé funkce protokolu.

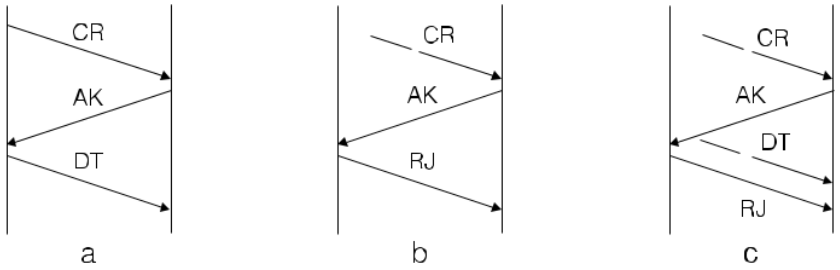
6.4 Navazování spojení

Důležitou a z hlediska implementace poměrně složitou funkcí transportní vrstvy je navazování spojení. Jednoduchá inicializace stavových proměnných, jak jsme ji poznali u vrstvy linkové, zde totiž nepostačuje. Příčin je více, nejpodstatnější je nedeterministické zpoždění transportních paketů při průchodu datagramovými sítěmi. Vliv nedeterministického zpoždění je obvykle ilustrován na následujícím příkladě:

Předpokládejme, že se náš počítač má spojit se vzdáleným bankovním počítačem, požádat o provedení transakce (například o převedení určité částky z našeho účtu) a spojení ukončit. Překročí-li doba přenosu žádosti o otevření spojení časový limit (time-out), náš počítač žádost zopakuje. V síti se pak pohybují dvě žádosti, prvá dorazí k bankovnímu systému a ten potvrdí otevření spojení. Náš počítač požádá o provedení transakce, ale paket opět "zabloudí"a bude po vypršení časového limitu zopakován. Bankovní systém po příchodu prvé ze žádostí transakci provede a potvrdí. Po příjmu potvrzení transakce požádáme o ukončení spojení. U něho se může situace s vypršením časového limitu a retransmisí opakovat.

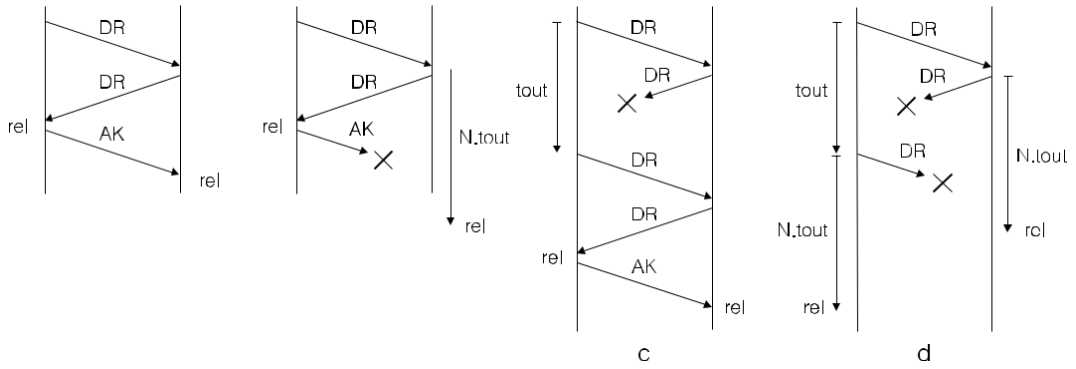
Výsledkem právě popsaného děje bude, že se v síti po uzavření transakce bude pohybovat kopie žádosti o otevření transportního spojení, kopie žádosti o provedení transakce a kopie žádosti o uzavření spojení. Dorazí-li tyto pakety k cílovému systému ve "vhodném"pořadí, je transakce zduplikována.

Chceme-li uvedené situaci zabránit musíme odlišit jednotlivé pokusy o navázání spojení a zabránit tomu, aby cílový systém považoval výše uvedenou posloupnost kopií za korektní posloupnost paketů. Pouhé číslování pokusů nepostačuje s ohledem na možné výpadky počítačů. Dobrou metodou je třífázový protokol (three-way handshake) využívaný jak v protokolu TP4, tak v protokolu TCP . Postup při korektním otevření spojení (a) a řešení problému opožděných duplikátů (b,c) uvádí obrázek 6.3.



Obrázek 6.3: Třífázový protokol otevírání spojení

Problémy nejsou spojené pouze s otevíráním spojení. Při uzavírání spojení je třeba zkontrolovat, zda se v síti nepohybují starší datové pakety. Opět zde pomůže třífázový protokol, problémy však může způsobit ztráta paketů. Příklad bezchybného ukončení spojení (a) a řešení problémů (b,c,d) uvádí obrázek 6.4.

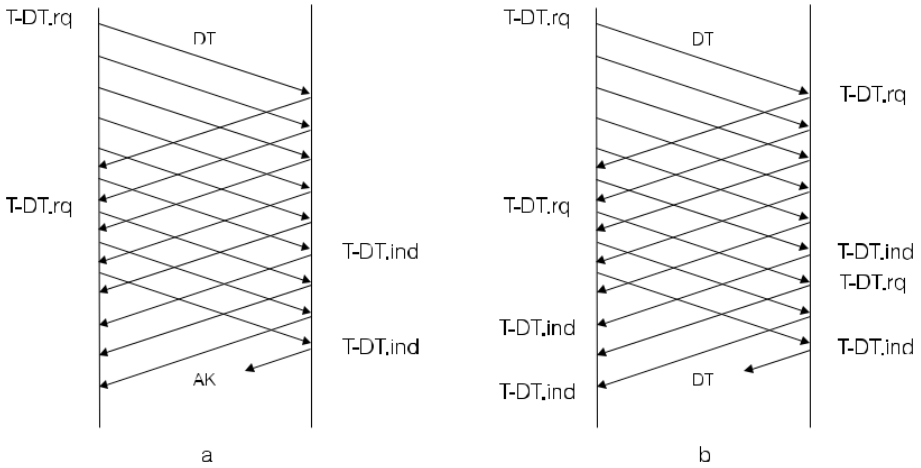


Obrázek 6.4: Čtyřfázový protokol uzavírání spojení

6.5 Koncové řízení toku

Úkolem řízení toku na úrovni sítě (ukázali jsme si metodu škrtících paketů) bylo zabránit zahlcení sítě zbytečným množstvím paketů. Úkol řízení toku na transportní úrovni je poněkud jiný. Nejde zde o ochranu sítě, cílem je rozumné využívání paměti v koncových počítačích a dostatečně volná synchronizace obou komunikujících partnerů.

Při otevírání spojení je u protokolu ISO specifikována požadovaná paměť na straně přijímače (v počtu paketů). Přijímající strana v potvrzeních kromě potvrzovacího čísla (u protokolu ISO udává číslo očekávaného paketu) uvádí i okamžitě dostupnou volnou paměť jako tzv. kredit. Ten se, na rozdíl od okénka, může i snižovat. Příklad přenosu datových TPDU uvádí obrázek 6.5, ve skutečném protokolu je redukován počet potřebných potvrzení využitím kumulativního (skupinového) potvrzování.

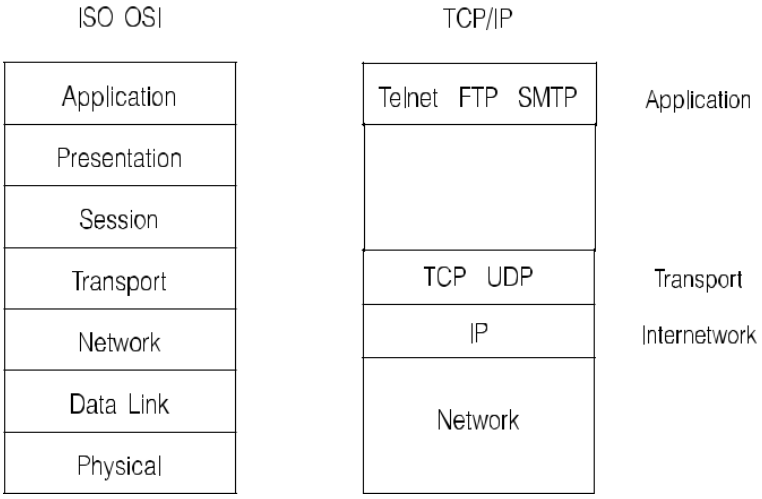


Obrázek 6.5: Koncové řízení toku

6.6 Model TCP/IP

Sloučení různorodých síťových technologií do jednotného komunikačního systému se stalo cílem návrhu technologie, označované jako internetworking. Technologie TCP/IP (Transmis-

sion Control Protocol/Internet Protocol) se stala základem současného Internetu. Návrhy, experimentální verze a konečné standardy TCP/IP jsou označovány jako RFC (Request for Comments) a pokrývají hlavně síťovou a transportní vrstvu (obr. 6.6).



Obrázek 6.6: TCP/IP model

Základem architektury TCP/IP je síťový protokol IP (Internet Protocol). Jeho pakety, případně rozložené do fragmentů, lze pro přenos mezi směrovači IP vkládat do rámců libovolné linkové technologie, nebo do paketů jiné síťové architektury (např. veřejné datové sítě X.25). Jednotlivé pakety, případně jejich fragmenty jsou k cíli směrovány nezávisle.

Vlastnosti protokolu IP nedovolují zabránit ztrátám, duplikaci a záměně pořadí paketů, resp. jejich fragmentů. Zvládnout tyto situace korektně přímo v aplikaci je obtížné, proto jsou běžně využívány transportní protokoly UDP (User Datagram Protocol) a TCP (Transmission Control Protocol), které potřebné podpůrné mechanismy zahrnují. Protokol UDP zajisťuje skládání fragmentů do paketů, multiplex (možnost využívat síťovou komunikaci více aplikacemi současně), a případně detekci chyb. Podstatně komplikovanější protokol TCP navíc podporuje koncové potvrzování (dovoluje eliminovat ztráty a duplikace paketů), rozklad aplikačních toků dat (stream) na pakety a poměrně komplikované mechanismy řízení toku.

6.7 Protokoly TCP a UDP

Protokoly TCP IP jsou v současnosti akceptovány jako de-facto standard pro komunikaci v rozsáhlých počítačových sítích. Architektura TCP/IP zahrnuje vlastní přenos IP paketů jednoduché datagramové rozhraní User Datagram Protocol (UDP) a dobře navržený protokol logického kanálu Transmission Control Protocol (TCP). TCP zajišťuje potvrzování v prostředí propojených sítí, ve kterých mohou být pakety dodávány v nezaručeném pořadí mohou být při přenosu štěpeny na fragmenty a mohou se ztrácet. Je vybaven důmyslným řízením toku a ochranou proti chybám vyvolaným opakovaným navazováním spojení. Aplikacím viditelné protokoly IP, UDP a TCP jsou podporovány služebními protokoly, které zajišťují transformace adres TCP/IP na adresy lokální sítě (ARP RARP), řízení sítě (ICMP) a podporu směrování (RIP, OSPF).

Dá se říct, že protokoly TCP/IP jsou v současné době k dispozici v libovolné lokální síti minimálně proto, aby zajistily spolupráci s počítači pod operačním systémem UNIX a propojení s Internetem. Aplikační rozhraní protokolů IP UDP a TCP jsou poměrně přesně definována v operačních systémech UNIX jako BSD sockety (BSD Sockets) nebo jako rozhraní TLI (Transport Layer Interface). Rozhraní v systémech Windows je obdobou BSD socketů doplněné o podporu asynchronního provádění funkcí.

Funkce rozhraní zahrnují vytváření (Socket) a rušení (Close) datových struktur řídících komunikaci na daném přípojném místě (portu) nebo po virtuálním kanále, jejich vazbu na logický kanál a vazbu na adresační informaci (Bind) a limit počtu neobsloužených požadavků na vstupu (Listen). Součástí rozhraní TCP jsou funkce pro pasivní a aktivní otevření kanálu (Accept a Connect) a pro jeho uzavření (Close). Přenos paketů a zpráv zajišťují volání funkcí Write a Read, spolu s několika formami funkcí Send a Receive.

6.7.1 Formát TCP a UDP paketu

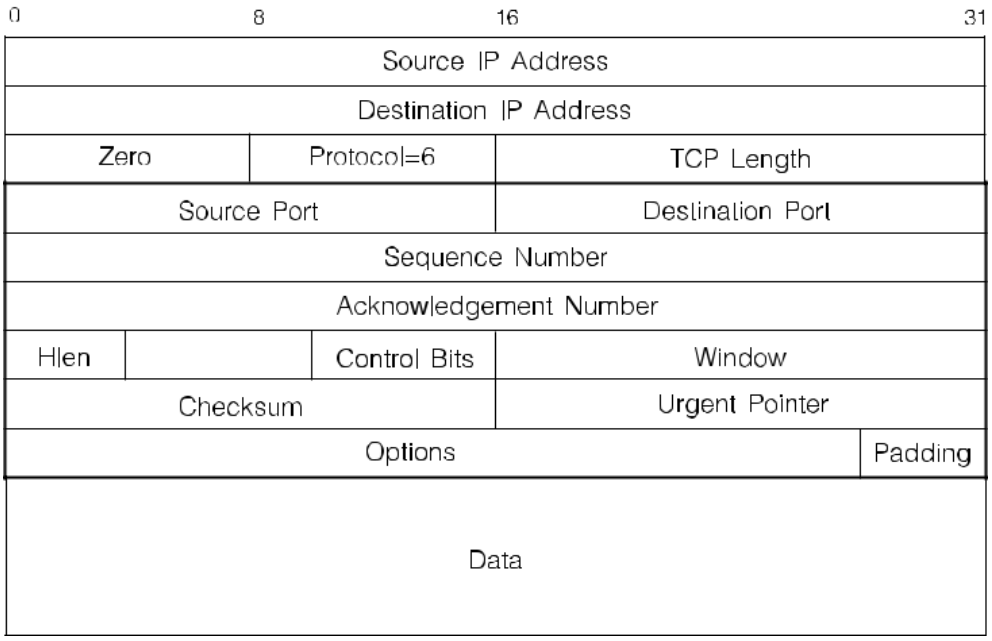
Hlavička IP paketu obsahuje údaj o verzi protokolu a o délce hlavičky ve slovech. Následující pole TOS definuje typ provozu (interaktivní, přenos dat) nebo požadavky na dobu odezvy, kapacitu kanálu a spolehlivost nebo bezpečnost přenosu. Pole Length uvádí délku paketu (nebo fragmentu) včetně hlavičky, pole Identification dovoluje identifikovat fragmenty paketu, na které se může paket při průchodu sítí rozpadnout. Tříbitové pole příznaků F dovoluje zakázat dělení paketu na fragmenty a rozpoznat poslední fragment v paketu, pole Offset definuje umístění fragmentu v paketu. V poli TTL najdeme počet 't sekund l' které zbývají paketu pro jeho cestu k adresátovi, hodnota je snižována nejméně o jedničku při průchodu každým směrovačem. Pole Prot identifikuje vyšší protokol, hodnota Prot=6 odpovídá protokolu TCP hodnota Prot=17 protokolu UDP. Následují adresy příjemce a odesílatele a případně pole Option pro služební informace.

Hlavičce TCP segmentu na obr. 6.7 předřazujeme "pseudohlavičku IP"která obsahuje podstatné údaje z IP hlavičky zahrnované do kontrolního součtu. Adresy portů jsou šestnáctibitové a jsou následovány údaji Sequence Number a Acknowledgement Number pro potvrzování. Pole HL uvádí délku hlavičky, příznaky Flgs slouží pro předávní služebních údajů při otevírání a rušení spojení, informují o platném potvrzení a prioritní informaci v segmentu. Pole Window dovoluje příjemci uvést velikost paměti alokované pro očekávaná data, slouží pro řízení toku. V poli Checksurn najdeme kontrolní součet segmentu včetně "pseudohlavičky" v inverznírm kódu). Pole Urgent Plit uvádí pozici prioritní informace v přenášených datech. Konečně, hlavička UDP datagramu nese pouze čísla portů, délku UDP datagramu a kontrolní součet.

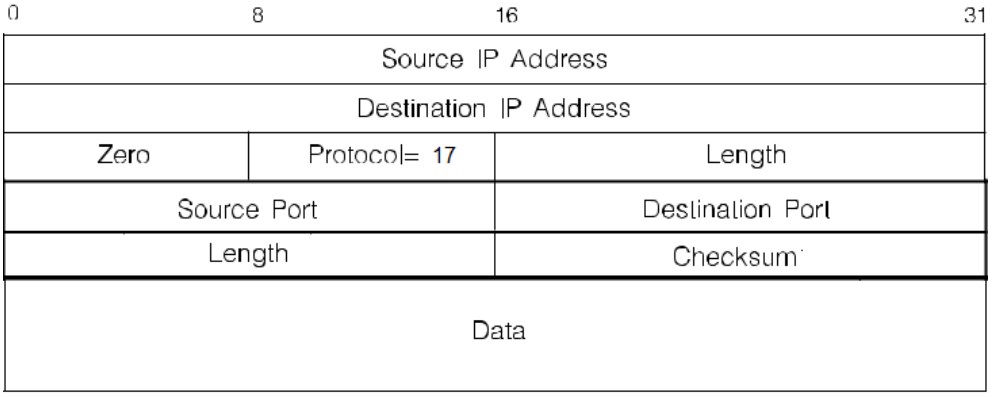
6.7.2 Mechanismy koncového řízení toku TCP

Dynamická změna okénka u transportního protokolu může být využita i pro řízení toku do sítě; to je zajímavé zvláště u protokolu TCP pro Internet, kde vnitřní mechanismy řízení toku na úrovni koncových spojení scházejí.

Základní mechanismus řízení toku u protokolu TCP se opírá o pole Window hlavičky. V něm příjemce dat sděluje, kolik znaků je schopen přijmout, počínajíc znakem s pozicí uvedenou v poli Ack. Mechanismus byl navržen pro přenos souborů (brání vyčerpání



Obrázek 6.7: Formát TCP paketu



Obrázek 6.8: Formát UDP paketu

vyrovnávací paměti na straně příjemce).

Při nízké rychlosti vstupu u interaktivní práce vede základní algoritmus k nízkému využití kapacity kanálu, jednotlivé znaky vstupu jsou totiž po vypršení časového limitu odesílány jako samostatné TCP segmenty, s příslušnou TCP a IP hlavičkou. Proto je protokol TCP doplňován o Nagleův algoritmus. Ten se snaží o maximální využití kanálů s dlouhou dobou přenosu. První znak je odeslán v samostatném TCP segmentu. Další znaky jsou ukládány do vyrovnávací paměti a odeslány až po příjmu potvrzení, a tento postup se dále opakuje. K mimořádnému odeslání TCP segmentu dochází při naplnění poloviny kreditu, nebo při naplnění maximální délky segmentu. V některých případech však může být Nagleův algoritmus na závadu, např. při komunikaci X-terminálu se vzdálenou aplikací.

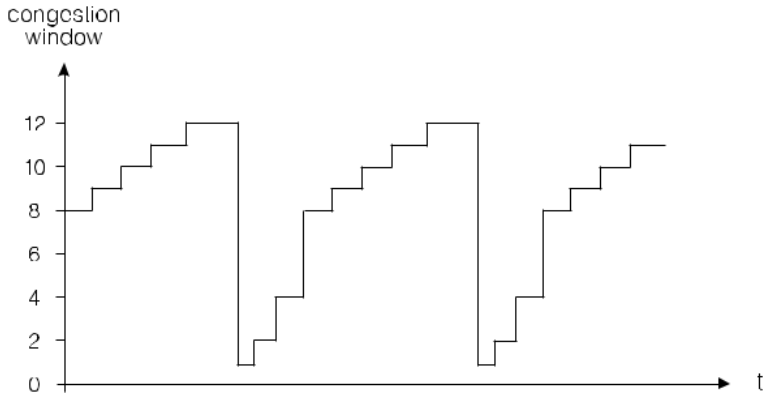
Podobně nepříjemné důsledky může mít i chování příjemce, který odebírá z vyrovnávací paměti jednotlivé znaky a oznamuje odesílateli jednoznakové kredity (Window =

1), řešením je odeslání potvrzení teprve po uvolnění většího bloku vyrovnávací paměti.

Mechanismus kreditu Window je primárně určen k omezení toku na velikost, kterou je schopen akceptovat příjemce. Velká hodnota kreditu dovolí vysílající straně odeslat do sítě řadu paketů a může vést k zahlcení sítě s nemění průchodností. Tomu lze bránit doplňkovými mechanismy řízení toku, které mají za cíl zahlcení sítě zabránit.

Předpokládejme, že známe časový limit, do kterého přijde potvrzení odeslaného segmentu na nepřetížené síti, tuto hodnotu označujeme jako RTT (Round Trip Time). Každé překročení tohoto časového limitu pak můžeme považovat za příznak zahlcení sítě (zahození paketu). Vysílající strana zahajuje přenos s hodnotou okénka (congestion window) o velikosti jednoho TCP segmentu o maximální délce (samozřejmě, pokud není kredit příjemce nižší, nebo pokud nevypršel časový limit Nagleova algoritmu). S příchodem každého potvrzení je velikost okénka zvětšována o jeden TCP segment o maximální délce. Výsledkem postupu je pomalé, exponenciální, zvětšování okénka slow-start, to tak může dosáhnout nejvýše 64 kB (počáteční hodnota limitu Threshold). Od tohoto limitu dochází k dalšímu otevírání okénka lineárně.

Výpadek potvrzení je příznakem zahlcení sítě a vede k návratu na velikost okénka o délce jednoho maximálního TCP segmentu a k současnému nastavení limitu Threshold na polovinu dosažené hodnoty okénka. Mechanismus, označovaný jako TCP-Tahoe chrání síť před zahlcením, jeho činnost z pohledu konkrétního transportního spojení ilustruje obrázek 6.9.



Obrázek 6.9: Chování TCP Tahoe při zahlcení

Oscilující koncové toky TCP se mohou snadno zasynchronizovat, výsledkem mohou být oscilace celé sítě. Technologickou ochranou proti takovému chování je náhodné zahazování TCP segmentů ještě před dosažením limitní délky fronty ve směrovači. Mechanismus je označován jako Random Early Detection (RED), pravděpodobnost zahození TCP segmentu se od dosažení určité délky fronty lineárně zvyšuje.

Nepříjemnou vlastností mechanismu je skutečnost, že i selektivní výpadek jednoho paketu vyvolá razantní přivření okénka. Modernější mechanismus TCP-Reno umí reagovat na příchod tří potvrzení se stejným potvrzovacím číslem, tedy na skutečnost, že došlo ke ztrátě segmentu, ale časová odezva sítě zůstala rozumná. Vysílač sníží limit Threshold na polovinu dosažené hodnoty okénka, a okénko na tutéž velikost. Navíc může odeslat segment jako reakci na každé z opakovaných potvrzení, i pokud by došlo k překročení limitu Threshold.

Konečně, novější technologie řízení toku TCP – TCP-Vegas se snaží odstranit skutečnost, že dosud popisované mechanismy reagují až na příznak zahlcení sítě. TCP-Vegas odhaduje schopnost sítě přenášet určitý tok dat na základě počtu potvrzení, která jsou přijata za dobu odpovídající dosud nejkratšímu Round Trip Time (RTT) - tento údaj považujeme za odezvu na nezatížené síti a tomu přizpůsobuje frekvenci odesílání TCP segmentů (a nastavení okénka).

6.7.3 Určení RTT

Dosud jsme předpokládali, že hodnota časového limitu je známa. V praxi hodnota časového limitu závisí na zpožděních v síti, je silně závislá na použité technologii a topologii sítě, a je nutné mít k dispozici mechanismus, který dovolí vhodný časový limit nastavit automaticky.

Základem pro nastavení časového limitu je určení hodnoty RTT . Jelikož se jedná o náhodnou veličinu, je rozumné vyfiltrovat střední hodnotu. Je používán exponenciální filtr (klouzavý průměr)

$$\text{sk} = as_{k-1} + (1 - a)sk$$

kde pro koeficient a je v TCP doporučena hodnota a = 0.9. Časový limit je pak nastavován na hodnotu Timeout = bs_k, kde pro koeficient b je v TCP doporučena hodnota b = 2.

Literatura

[1] RFC 791 - Internet Protocol. Dostupné z: https://datatracker.ietf.org/doc/html/rfc791

[2] IANA - protocol database. Dostupné z: https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml

[3] RFC 4632 - Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. Dostupné z: https://datatracker.ietf.org/doc/html/rfc4632

[4] RFC 1817 - CIDR and Classful Routing. Dostupné z: https://datatracker.ietf.org/doc/html/rfc1817

[5] RFC 1918 - Address Allocation for Private Internets. Dostupné z: https://datatracker.ietf.org/doc/html/rfc1918

[6] RFC 6890 - Special-Purpose IP Address Registries. Dostupné z: https://datatracker.ietf.org/doc/html/rfc6890

[7] IANA. Dostupné z: https://www.iana.org

[8] RFC 792 - INTERNET CONTROL MESSAGE PROTOCOL. Dostupné z: https://datatracker.ietf.org/doc/html/rfc792

[9] The Story of the PING Program. Dostupné z: https://ftp.arl.army.mil/~mike/ping.html

[10] RFC 1191 - Path MTU Discovery. Dostupné z: https://datatracker.ietf.org/doc/html/rfc1191

[11] RFC 826 - An Ethernet Address Resolution Protocol. Dostupné z: https://datatracker.ietf.org/doc/html/rfc826

[12] RFC 5227 - IPv4 Address Conflict Detection. Dostupné z: https://datatracker.ietf.org/doc/html/rfc5227

[13] RFC 2390 - Inverse Address Resolution Protocol. Dostupné z: https://datatracker.ietf.org/doc/html/rfc2390

[14] RFC 2427 - Multiprotocol Interconnect over Frame Relay. Dostupné z: https://datatracker.ietf.org/doc/html/rfc2427

[15] RFC 2131 - Dynamic Host Configuration Protocol. Dostupné z: https://datatracker.ietf.org/doc/html/rfc2131

[16] RFC 951 - BOOTSTRAP PROTOCOL (BOOTP). Dostupné z: https://datatracker.ietf.org/doc/html/rfc951

[17] RFC 2132 - DHCP Options and BOOTP Vendor Extensions. Dostupné z: https://datatracker.ietf.org/doc/html/rfc2132

[18] RFC 5010 - The Dynamic Host Configuration Protocol Version 4 (DHCPv4) Relay Agent Flags Suboption. Dostupné z: https://datatracker.ietf.org/doc/html/rfc5010

[19] RFC 3972 - Dynamic Configuration of IPv4 Link-Local Addresses. Dostupné z: https://datatracker.ietf.org/doc/html/rfc3972

[20] IPv6 Internetový protokol verze 6. Dostupné z: https://knihy.nic.cz/files/edice/IPv6-2019.pdf

[21] RFC 1883 - CIDR and Classful Routing. Dostupné z: https://datatracker.ietf.org/doc/html/rfc1883

[22] RFC 8200 - Internet Protocol, Version 6 (IPv6) Specification. Dostupné z: https://datatracker.ietf.org/doc/html/rfc8200

[23] RFC 4291 - IP Version 6 Addressing Architecture. Dostupné z: https://datatracker.ietf.org/doc/html/rfc4291

[24] Guidelines for 64-bit Global Identifier (EUI-64). Dostupné z: https://ccie.lol/wp-content/uploads/2016/10/eui64.pdf

[25] RFC 8981 - Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6. Dostupné z: https://datatracker.ietf.org/doc/html/rfc8981

[26] RFC 4443 - Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. Dostupné z: https://datatracker.ietf.org/doc/html/rfc4443

[27] RFC 4861 - Neighbor Discovery for IP version 6 (IPv6). Dostupné z: https://datatracker.ietf.org/doc/html/rfc4861

[28] RFC 4862 - IPv6 Stateless Address Autoconfiguration. Dostupné z: https://datatracker.ietf.org/doc/html/rfc4862

[29] RFC 8415 - Dynamic Host Configuration Protocol for IPv6 (DHCPv6). Dostupné z: https://datatracker.ietf.org/doc/html/rfc8415

[30] RFC 6146 - Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers. Dostupné z: https://datatracker.ietf.org/doc/html/rfc6146

[31] RFC 6052 - IPv6 Addressing of IPv4/IPv6 Translators. Dostupné z: https://datatracker.ietf.org/doc/html/rfc6052

[32] RFC 6333 - Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion. Dostupné z: https://datatracker.ietf.org/doc/html/rfc6333

[33] RFC 4380 - Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs). Dostupné z: https://datatracker.ietf.org/doc/html/rfc4380

[34] IPv6 Ready Logo Program. Dostupné z: https://www.ipv6ready.org/index.html

[35] RFC 6275 - Mobility Support in IPv6. Dostupné z: https://datatracker.ietf.org/doc/html/rfc6275

[36] RFC 8201 - Path MTU Discovery for IP version 6. Dostupné z: https://datatracker.ietf.org/doc/html/rfc8201

[37] RFC 7761 - Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised). Dostupné z: https://datatracker.ietf.org/doc/html/rfc7761

[38] RFC 1034 - DOMAIN NAMES - CONCEPTS and FACILITIES. Dostupné z: https://datatracker.ietf.org/doc/html/rfc1034

[39] RFC 959 - FILE TRANSFER PROTOCOL (FTP). Dostupné z: https://datatracker.ietf.org/doc/html/rfc959

[40] RFC 4033 - DNS Security Introduction and Requirements. Dostupné z: https://datatracker.ietf.org/doc/html/rfc4033

[41] RFC 1035 - DOMAIN NAMES - IMPLEMENTATION and SPECIFICATION. Dostupné z: https://datatracker.ietf.org/doc/html/rfc1035

[42] RFC 6762 - Multicast DNS. Dostupné z: https://datatracker.ietf.org/doc/html/rfc6762

Seznam obrázků

Seznam tabulek