VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Komunikační technologie

Garant předmětu:
doc. Ing. Jan Jeřábek, Ph.D.

Autoři textu:
doc. Ing. Jan Jeřábek, Ph.D.

BRNO **2020**

Autor	doc. Ing. Jan Jeřábek, Ph.D.
Název	Komunikační technologie
Vydavatel	Vysoké učení technické v Brně Fakulta elektrotechniky a komunikačních technologií Ústav telekomunikací Technická 12, 616 00 Brno
Rok vydání první verze	2013
Náklad	elektronicky
ISBN první verze	978-80-214-4713-4
Poslední aktualizace	20.8.2020

Tato publikace neprošla redakční ani jazykovou úpravou

Obsah

1 Úvod

Internet představuje v současnosti jedno z nejrozsáhlejších celosvětových inženýrských děl. Internet sestává z miliard propojených počítačů a jim podobných zařízení různého typu a účelu, komunikačních linek a také různorodých propojujících zařízení. Všechny tyto prostředky slouží k tomu, aby byla umožněna vzájemná elektronická komunikace mezi zařízeními prakticky libovolného typu. Mezi tato zařízení patří tradičně stolní počítače, jejich přenosné varianty a servery. Nemůžeme však opomenout ani chytré telefony, tablety, herní konzole, televize, různé monitorovací, přístupové a zabezpečovací systémy, specializované počítače, nositelnou elektroniku (např. chytré hodinky), různé senzory a snímače, průmyslové řídicí systémy či např. některé dopravní prostředky. Je tedy zřejmé, že Internet je komplexní a přitom velmi různorodý a že tento globální systém musí nezbytně fungovat podle jistých základních pravidel a musí být také respektovány určité základní principy pro jeho strukturu. Pochopení těchto principů a pravidel nám umožní porozumět základům Internetu a obecně základům prakticky libovolných počítačových komunikačních systémů a sítí.

Elektronická komunikace představuje sdělování informací mezi několika místy podle dohodnutých pravidel. Existuje velké množství technik a technologií, které nám elektronickou komunikaci umožňují, a to různými způsoby a v různých podmínkách. Již delší dobu je dobře patrná konvergence veškeré elektronické komunikace do vzájemně propojených digitálních sítí propojených Internetem. Dobrá znalost komunikačních technik představuje jednu ze základních charakteristik absolventů telekomunikačních a počítačových studijních programů či oborů, význam má však i v dalších oblastech, jako je např. automatizace.

Předmět se zaměřuje na způsoby komunikací a především na popis základních principů současných komunikačních systémů, metod přenosu informace a architekturu komunikace. Poskytuje znalosti o všech sedmi vrstvách, na které jsou elektronické komunikační systémy běžně členěny. Velká pozornost je věnována síťovému modelu ISO/OSI i TCP/IP a celé řadě souvisejících protokolů.

2 Zařazení předmětu ve studijních programech

Předmět „Komunikační technologie“ je vyučován v zimním semestru bakalářského studia několika programů či oborů studia. Pro některé z nich se jedná o povinný oborově zaměřený předmět, jehož cílem je poskytnout studentům základní znalosti a přehled v oblasti fungování Internetu a komunikačních technologií. Předmět je jako volitelný nabízen i studentům jiných programů či oborů Fakulty elektrotechniky a komunikačních technologií.

Předmět nepředpokládá žádné výchozí odborné znalosti z oblasti Internetu a komunikačních technologií, důležité je pouze všeobecné povědomí o počítačových sítích, telekomunikacích, informatice, matematice a fyzice. Na předmět navazují předměty zabývající se architekturou sítí, vybavením počítačových sítí, přenosem multimediálních dat přes tyto sítě, zajištěním kvality služeb v komunikačních systémech a taktéž předměty, ve kterých se věnuje pozornost všem bezpečnostním aspektům digitální komunikace a v neposlední řadě kryptografii, tedy nauce o šifrování.

3 Technika sítí a protokolů

3.1 Základní stavební prvky současného Internetu

Jak bylo uvedeno již v úvodu tohoto textu, Internet se skládá z počítačů různého typu a jejich různorodého propojení. Tyto počítače označujeme jako koncový systém (*end system*), popř. koncové zařízení (*end device*), někdy pak pouze jako host (*host*).

Koncová zařízení jsou propojena díky komunikačním trasám (linkám) a také propojujícím zařízením. Existuje mnoho typů komunikačních linek využívajících různá fyzická přenosová média. Řada z nich je zpravidla vhodná pro použití v konkrétních typech sítí, přičemž některé z těchto sítí jsou ukázány v **Obr. 3-1**. Komunikační trasy jsou schopny přenášet data různě rychle, tedy mají různou přenosovou rychlost (vyjádřena v bitech za sekundu, *bits per second*, *bps*). Přenášená data jsou nejčastěji rozdělována před zahájením přenosu u odesilatele (*source system*) na menší jednotky, tzv. pakety, a tyto pakety jsou pak přenášeny po přenosových trasách mezi koncovými systémy, přičemž ke složení původních dat dojde až u příjemce (*destination system*).

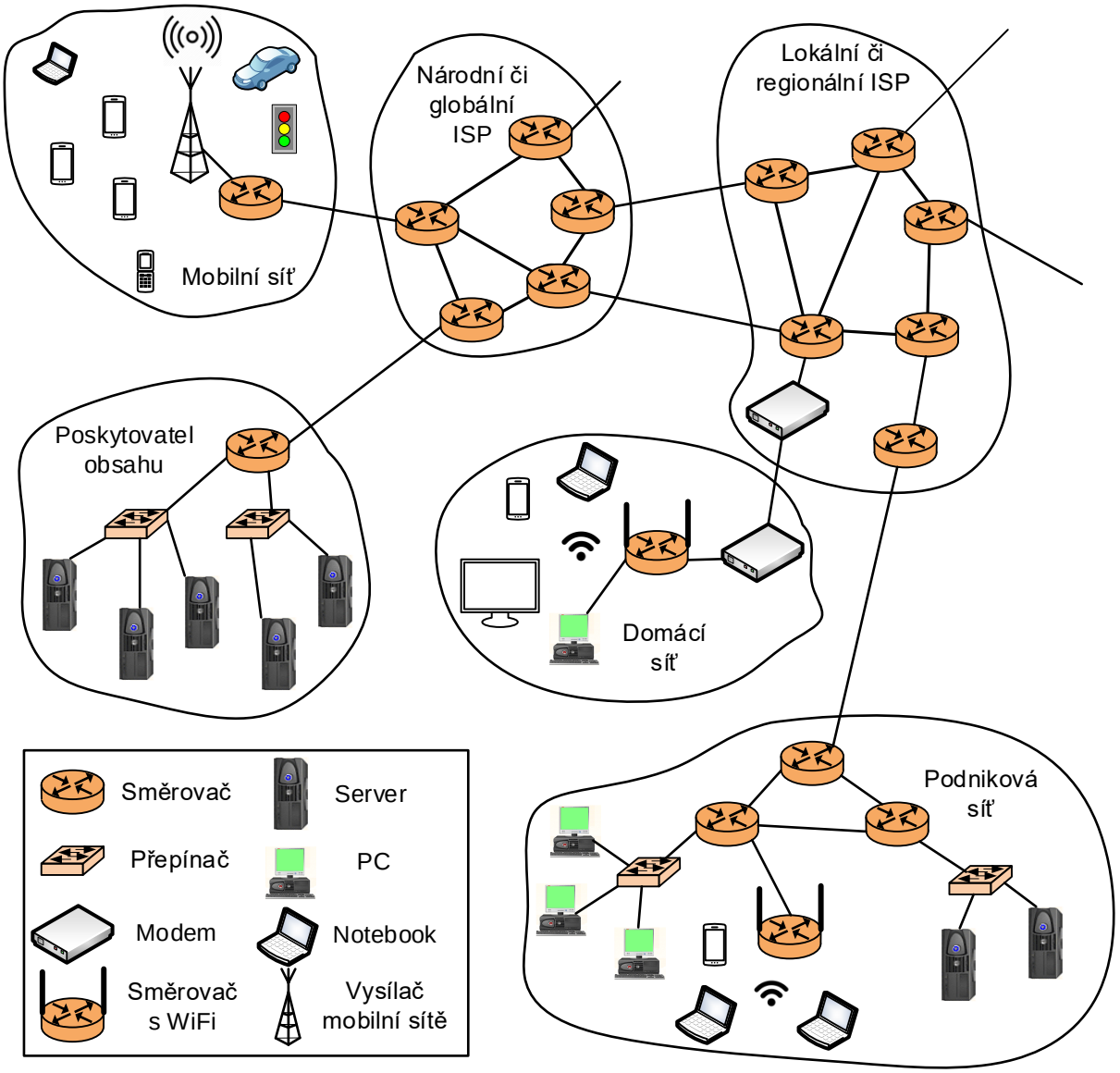
Propojující (mezilehlé) zařízení bývají označována mnoha způsoby v souvislosti s typem komunikačních tras, způsobem fungování či typem sítí, kde se používají. Nejčastěji se setkáme s označením **směrovač** (*router*) či **paketový přepínač** (*packet switch*). Tato zařízení jsou schopna přijmout paket na některém ze svých rozhraní a předat (*forward*) jej dál na komunikační linku směrem k adresátovi. Cesta, kterou je paket předáván směrem od odesilatele k příjemci přes komunikační linky a propojující zařízení, bývá anglicky označována jako *route* nebo *path*.

Koncové systémy přistupují k Internetu zpravidla přes **poskytovatele připojení** (ISP = *Internet Service Provider*). Existuje celá řada typů poskytovatelů a způsobů připojení koncových zařízení. Síť ISP sestává zejména z komunikačních tras a propojujících zařízení. Tato infrastruktura má za úkol propojit koncová zařízení mezi sebou a to jak na úrovni jednoho poskytovatele připojení, tak mezi různými poskytovateli připojení či s poskytovateli obsahu (*content providers*). **Poskytovatelé obsahu** jsou schopni ze svých sítí (CDN = *content delivery network*) poskytnout koncovým systémům data různého typu (web, audio, video, software, …).

Všechna zařízení v sítích používají **protokoly**, které řídí odesílání, příjem i případně zpracování informací v Internetu. Dva z nejvýznamnějších protokolů se jmenují *Transmission Control Protocol* (TCP) a *Internet Protocol* (IP). Sada (soubor) velkého množství protokolů, které jsou důležité pro fungování internetu, se označuje právě podle těchto dvou protokolů, tj. **TCP/IP**.

Protokoly a další náležitosti fungování internetu **jsou přesně specifikovány v dokumentech**, které vydává organizace *Internet Engineering Task Force* (IETF) a jsou označovány jako *requests for comments* (RFC). Tyto dokumenty mají vždy číselné označení (např. RFC 8200) identifikující konkrétní dokument, určitý stav (standard, návrh, informační dokument, návaznost na další dokumenty, …) a těchto dokumentů v současné době existuje několik tisíc. Mimo to existují další standardy, které vydává organizace IEEE (např. dokumenty IEEE 802.11 popisující technologicky lokální bezdrátové sítě Wi-Fi a související problematiku). Dále pak je třeba vzít v potaz např. dokumenty mezinárodní telekomunikační

unie (International Telecommunication Union - Telecommunication Standardization Sector = ITU-T), definující některé telekomunikační standardy, např. způsoby kódování a komprese hlasu pro telefonii, či specifikace *The 3rd Generation Partnership Project* (3GPP) popisující fungování mobilních sítí, např. přístupové části mobilní sítě 4. generace (4G). Všechna tato doporučení, standardy a specifikace mají vliv na to, jak fungují sítě a komunikace mezi nimi (**Obr. 3-1**).

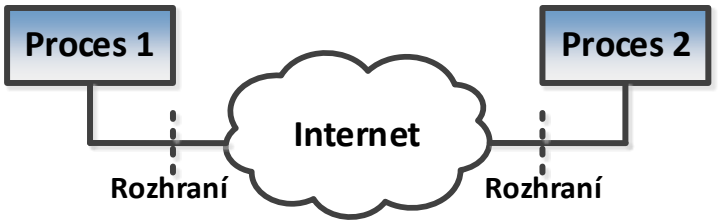


Obr. 3-1: Zjednodušené schéma Internetu s uvedením vybraných komponent a základního způsobu propojení základních typů sítí

Z jiného úhlu pohledu **můžeme Internet charakterizovat jako infrastrukturu poskytující služby distribuovaným aplikacím**, které běží na koncových zařízeních. Mezilehlá zařízení se o tyto aplikace do značné míry nestarají a pouze zprostředkovávají komunikaci mezi těmito koncovými zařízeními, na kterých aplikace běží. Tyto aplikace představují vždy určitý počítačový program, který je spuštěn v rámci nějakého operačního systému. Jedná se tedy o procesy běžící na různých koncových zařízeních, které využívají

Internet k tomu, aby spolu komunikovaly, jak zachycuje **Obr. 3-2**. Pozn.: Tento obrázek uvažuje pro zjednodušení pouze komunikaci typu bod-bod, což je jen jeden z možných případů.

Základním předpokladem pro komunikaci je definice rozhraní mezi procesem a sítí. Toto **rozhraní** musí definovat strukturu a formát předávaných uživatelských a řídicích dat (zpráv). Z programátorského hlediska je toto rozhraní označeno jako ***socket interface***.



Obr. 3-2: Zjednodušené schéma komunikace mezi procesy pracujícími na samostatných koncových zařízeních propojených obecnou sítí (Internetem)

3.2 Protokoly obecně

Protokoly jsou důležitou složkou komunikace v počítačových sítích. Protokoly popisují způsob interakce mezi komunikujícími stranami. Protokoly se v určité formě běžně využívají i v mezilidské komunikaci. Komunikaci obvykle zahajujeme pozdravem, který bývá opětován. Následuje např. vznesení nějakého požadavku (např. položení otázky) a v ideálním případě pak přímo odpověď. Na závěr lze očekávat rozloučení všech účastníků komunikace. Interakce může být samozřejmě narušena různými faktory (hluk, neznalost jazyka, neporozumění, …) a lidé jsou schopni na tyto situace nějakým způsobem reagovat.

Obdobně u protokolů v počítačových sítích se velice často potkáme s definovaným systémem interakce mezi komunikujícími stranami, stejně tak jako způsoby reakce na nestandardní stavy (např. že druhá strana nereaguje). Rozdílem také je, že u protokolů v počítačových sítích spolu komunikují hardwarové a softwarové komponenty, což umožňuje použít velmi různorodá schémata komunikace, stejně tak jako výrazně rychlejší výměnu informací.

Běžně se v počítačových sítích používají desítky různých protokolů různých účelů. Řadu těchto protokolů je nutné znát pro pochopení způsobu fungování počítačových sítí. Celkově však protokolů existují tisíce. Některé z nich jsou velmi jednoduché, některé naopak velmi komplexní. Většina z nich je používána jen ve velmi specifických prostředích či scénářích a není nutné se jim věnovat za účelem dosažení obecné znalosti fungování komunikace v počítačových sítí.

Aby komunikace daným protokolem fungovala, musí ho vždy znát a dodržovat každá z komunikujících stran. Obecně řečeno tedy protokol definuje formát a pořadí zpráv vyměňovaných mezi komunikujícími stranami, stejně tak jako způsob reakce na odeslání/přijetí zprávy či jinou událost.

3.3 Přístupové sítě (technologie)

Přístupové sítě představují důležitou část infrastruktury, která připojuje koncová zařízení, případně koncové sítě k prvnímu směrovači sítě poskytovatele připojení a tedy pak i dál do Internetu. Běžně se v této souvislosti používají termíny *edge network* (koncová síť), *access network* (přístupová síť) a *edge router* (první směrovač u ISP).

Možnými přístupovými technologiemi domácích i podnikových sítí jsou:

- **xDSL** (Digital Subscriber Line) – rozšíření infrastruktury klasického telekomunikačního operátora za účelem umožnění přístupu k Internetu. Písmeno „x" specifikuje konkrétní skupinu standardů, např. VDSL (Very High Speed DSL). Běžné přenosové rychlosti se u xDSL v současnosti pohybují od jednotek Mb/s až přes stovky Mb/s v závislosti na technických i obchodních faktorech. Většinou je přenosová rychlost výrazně nesymetrická, tj. v jednom směru je přenos možný výrazně rychleji než ve druhém (*download* vyšší než *upload*). Výhodou této technologie je využití stávající v řadě zemí poměrně rozsáhlé telekomunikační infrastruktury.
- **Kabelové sítě** – tyto technologie využívají existující infrastrukturu kabelových televizních operátorů. Pro přenos dat v těchto sítích je důležitá specifikace DOCSIS (*Data Over Cable Service Interface Specification*), která definuje způsob a i rychlost komunikace v těchto sítích. Teoretické přenosové rychlosti jsou spíše o něco vyšší než u xDSL technologií (desítky až tisíce Mb/s) a taktéž platí, že *download* je zpravidla vyšší než *upload*.
- **Wi-Fi a příbuzné bezdrátové sítě** – tyto technologie jsou často využívány v oblastech, kde není vybudována využitelná kabelová infrastruktura. Typickými vlastnosti těchto technologií je z důvodu použitého přenosového prostředí (vzduch) spíše nižší stabilita připojení (např. vliv počasí) a dosahované přenosové rychlosti jsou taktéž spíše nižší (desítky až stovky Mb/s).
- **Ethernet** – použití LAN (Local Area Network) technologie na připojení zákazníků. Častější spíše u větších či firemních zákazníků, či u metropolitních ISP. Přenosové rychlosti mohou být velmi vysoké, závisí pak často především na obchodních faktorech. Běžné jsou rychlosti 100 Mb/s a 1 Gb/s, ale možné jsou i výrazně vyšší rychlosti běžné spíše u velkých sítí. Velmi často je přenosová rychlost symetrická, tj. rychlosti v obou směrech přenosu jsou shodné.
- **FTTH** (Fiber to the home) – technologie založené na tom, že optické vlákno, jakožto nejlepší dosud existující přenosové médium, je dovedeno až k zákazníkovi. Přenosové rychlosti mohou z technického hlediska snadno přesahovat i 1 Gb/s a tyto sítě jsou často budovány jako tzv. PON (*Passive Optical Network*), o kterých bude ještě v tomto textu zmínka později.
- **3G/4G/LTE mobilní sítě** – běžně využívané sítě výhodné především vysokou mírou a snadností mobility koncového zařízení. V případě mobilních sítí 4G/LTE mohou být přenosové rychlosti srovnatelné se spíše pomalejšími xDSL či Wi-Fi technologiemi (vyšší desítky Mb/s pro *download*, nízké desítky Mb/s *upload*).
- **Satelitní sítě** – v hustě obydlených oblastech velmi málo využívaný typ připojení, pro který je specifická spíše nízká přenosová rychlost (přibližně do desítek, maximálně stovek Mb/s) a především velké zpoždění při přenosu paketů z důvodu velkých přenosových vzdáleností, které výrazně ovlivňuje některé typy komunikace či protokolů.

Výhodou je však dostupnost i mimo obydlené oblasti, např. v oceánech či polárních oblastech.

3.4 Transportní sítě (jádro sítě)

Jádro sítě má za úkol propojit vzájemně jednotlivé přístupové sítě, může se jednat např. o transportní síť ISP. Jádro je velmi různorodé propojení různých sítí a je často tvořeno velkým množstvím zařízení (směrovačů, ústředen, …) a přenosových tras. Transportní sítě mají zpravidla vysokou přenosovou kapacitu, přenosové trasy jsou často sdíleny a využívány pro přenos dat z více přístupových sítí a kumulativně jsou dosahovány často velmi vysoké přenosové rychlosti až v řádu násobků Tb/s. V transportních sítích se potkáme s různými technologiemi a způsoby přenosu, které se často označují jako způsoby komutace. Nejčastějším způsobem v současnosti je tzv. komutace paketů, setkat se však můžeme i s jinými typy přenosu, jako je komutace okruhů. Podrobněji problematiku komutací rozebírá kap. 4.1, kde je o dané problematice pojednáno obecněji, ne jen z pohledu transportních sítí.

3.5 Vzájemné propojení sítí

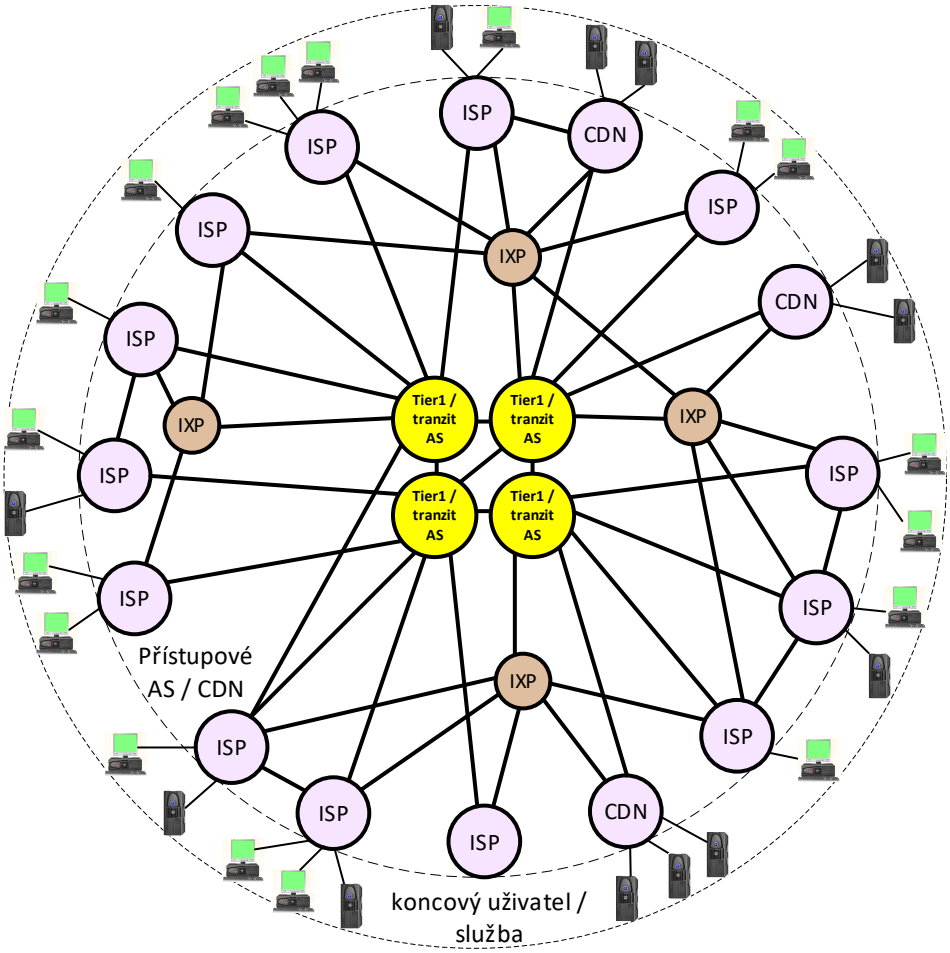
Transportní sítě jednotlivých ISP je třeba vzájemně propojit, aby byla možná komunikace koncových zařízení připojených do sítí různých ISP. Propojením jednotlivých ISP či dalších sítí vzniká síť sítí, což je vlastně Internet. Jednotlivým sítím z pohledu celého Internetu se říká **autonomní systémy (AS)** a existují jich aktuálně desítky tisíc. Vzhledem k celkovému počtu AS není možné přímé propojení každého AS se všemi ostatními AS. Způsob vzájemného propojení AS je velmi různorodý a ve většině případů nepřímý (zprostředkovaný). Někdy jsou AS propojeny pomocí tzv. **IXP** (*Internet Exchange Point*), což jsou uzly velkého významu, do kterých je připojeno velké množství často geograficky blízkých AS. Těchto IXP uzlů existují po celém světě nízké tisíce. V jiných případech pak komunikace probíhá přes tzv. **Tier1/tranzit AS**, které často nepřipojují přímo koncové zákazníky, ale prostřednictvím své globální sítě propojují jiné, geograficky vzdálené AS. Těchto AS je možné v Internetu najít několik stovek.

Způsoby propojení lze nalézt v grafickém znázornění i na **Obr. 3-3**. Zde jsou schematicky zaznačeny přístupové sítě (připojení koncových zařízení k síti ISP), transportní sítě (samotná síť ISP či CDN kteréhokoliv AS) a taktéž vzájemné propojení těchto AS (spojnice mezi ISP, CDN či Tier1/tranzit AS). Uvedené schéma představuje velmi zjednodušený pohled na vzájemné propojení AS, který však postačuje na to, aby bylo možné udělat si představu o principech globální topologie Internetu.

Jak již bylo uvedeno, z hlediska způsobu přenosu, se nejvíce v datových sítí využívá komutace paketů popsaná v kap. 4.1. Zařízení, která předávají pakety, se nejčastěji označují jako směrovače. Po trase mezi koncovými zařízeními, která spolu komunikují, mohou být pakety předávány přes nemalé množství směrovačů z různých AS (od jednotek až po nízké desítky mezilehlých zařízení[1]). Každý ze směrovačů musí rozhodnout, kam přijatý paket dále odešle, aby se dostal k adresátovi. K tomu se zpravidla využívají tzv. směrovací tabulky, které si vytváří každý směrovač (budeme se jim věnovat později). Samozřejmě platí, že výkonnost zařízení, jejich počet, délka a typ dílčích přenosových tras mají velký vliv na parametry dosahované při přenosu paketů, stejně tak jako aktuální zatížení těchto zařízení a zaplnění

[1] Množství těchto směrovačů lze často zjišťovat přes utilitu `traceroute`, která bude představena na cvičení předmětu.

kapacit přenosových tras. Velký vliv má také použitý komunikační protokol a další faktory, čemuž se budeme taktéž věnovat později.



Obr. 3-3: Zjednodušená struktura vzájemného propojení sítí (autonomních systémů) – struktura Internetu z globálnějšího pohledu

3.6 Základní parametry paketové sítě

Výkonnost a také efektivita sítě je měřena především prostřednictvím parametrů, které souvisí s tím, jak rychle jsme přes tuto síť schopni přenášet informace. Důležitá je především dostupná **šířka pásma** (*bandwidth*) nebo **datová rychlost** (*data rate*), případně **přenosová rychlost** (*bit rate*) a **propustnost** sítě (*throughput*), dále pak **zpoždění** (*delay, latency*). Pozn.: tyto pojmy jsou bohužel často používány špatně nebo zavádějícím způsobem.

Šířka pásma je tradičně spjata spíše s kmitočtovým rozmezím užitým např. při radiovém přenosu, jednotkou je tedy Hz. Avšak v souvislosti se sítěmi je nejčastěji zaměňována s teoretickou maximální dosažitelnou přenosovou rychlostí, která je uváděna v násobcích bitů za sekundu, (tj. anglicky b/s; bps = *bit per second,* česky bit/s), dnes typicky v Mbit/s (Mbps)[2] a Gbit/s (Gbps). Stejné jednotky jsou využívány i pro propustnost a datovou

[2] Je třeba mít na paměti, že mega (M) v případě přenosové rychlosti a mega z hlediska velikosti dat zpravidla není úplně to stejné. Mega totiž může dle souvislosti znamenat 10^6 nebo 2^{20}. V přenosových rychlostech se vychází z šířky pásma v MHz, kde mega je 10^6, a proto i zde platí, že mega značí 10^6. Naproti tomu u velikosti

nebo přenosovou rychlost**.** Všechny tyto parametry značí **počet bitů, které lze** daným kanálem (dle její technické specifikace) **přenést za sekundu**. Z pohledu uživatele jde spíše o teoretickou hodnotu, které není možné z důvodů různé (ale nevyhnutelné) neefektivity nebo redundance dosáhnout v koncové aplikaci. Platí, že např. síť s rychlostí 100 Mbit/s je schopna za 1 sekundu přenést 100 miliónů bitů, což znamená, že vyslání jednoho bitu trvá pouze 10 ns. Tyto hodnoty se v souvislosti s pokrokem v technologiích neustále zlepšují.

Naproti tomu **propustnost** je spíše měřená veličina, zpravidla již tedy v reálném nasazení a mezi koncovými systémy. Tato hodnota je proto vždy nižší než teoretická šířka pásma. Propustnost je mimo výše uvedené omezena také např. charakterem komunikace, formátem zpráv nebo konstrukcí a vytížením všech zařízení po trase na síti. Při výše uvažované šířce pásma 100 Mbit/s (standardní Fast Ethernet) může měřená propustnost dosahovat maximálně hodnoty okolo 95 Mbit/s. V případě přenosové trasy s více různými segmenty je propustnost vždy omezena nejpomalejším úsekem celé trasy (*bottleneck link*).

Dalším důležitým parametrem je **zpoždění** (*delay, latency*), který vyjadřuje, jak dlouho skutečně trvá přenos 1 bitu po dané trase. Zpoždění je vždy uvažováno jako čas (v sekundách) a je závislé především na délce trasy, velikosti přenášené zprávy, šířce pásma daného kanálu, mezilehlých zařízeních a samozřejmě také provozu (zatížení) na trase. Rychlost šíření signálu je fyzikálně limitována (3.10^8 m/s ve vakuu; $2{,}3.10^8$ m/s v měděném kabelu; 2.10^8 m/s v optickém vláknu).

Z hlediska zpoždění je velmi důležitým parametrem také **obousměrné koncové zpoždění**, tj. souhrnné zpoždění trasy tam a zpět mezi koncovými systémy. Anglicky je tento pojem označován jako *round-trip-time* (**RTT**). Hodnota RTT je rovna součtu zpoždění trasy v jednom a druhém směru[3].

Obousměrné koncové zpoždění při přenosu mezi stanicemi v jedné lokalitě se pohybuje okolo 1 ms, přenos mezi kontinenty pak může trvat až 100 ms, přenos země-družice-země pak i 300 ms. Toto zpoždění může mít na komunikaci jen malý nebo naopak zcela zásadní vliv. Pokud budeme přenášet pouze jeden bit (nebo jen velmi malou zprávu), nehraje velikost přenosové rychlosti daného kanálu prakticky žádnou roli. V této situaci je pro rychlost komunikace zásadní zpoždění přenosové trasy, jelikož i při velmi malé šířce pásma je doba vyslání informace velmi krátká (mikrosekundy). Naproti tomu pokud přenášíme velké objemy dat (očekáváme delší dobu přenosu, např. v řádu sekund), zpoždění trasy (v řádu ms) již nehraje takovou roli a roste význam šířky pásma. Obecně však vždy platí, že čím nižší zpoždění, tím pro komunikaci lépe.

Ztrátovost paketů je dalším základním parametrem. Uvádí se typicky v procentech a v relativní míře vyjadřuje, kolik paketů z celkového počtu nebylo v pořádku doručeno k adresátovi. Ztráty paketů mohou být způsobeny např. přetížením zařízení či poruchami na přenosové trase. V ideálním případě je ztrátovost nulová, v reálném případě pak často téměř nulová, popř. ve výši maximálně jednotek procent.

---

dat se vychází běžně z velikosti zabrané paměti, kde pracujeme běžně v mocninách dvou, proto zde 1 Mbit značí 2^{20} bitů, což je 1 048 576 bitů.

Obdobně výše uvedené platí i pro další používané jednotky, tj. zejména kbit, Gbit, případně Tbit.

Mimo to je vždy také třeba dát si pozor na pojmy „bit“ (b) a „byte“ (B). Platí, že 1 Byte = 8 bit.

Pozn.: V dalším textu bude z důvodu úspornosti uváděna anglická zkratka „B“ (byte) a „b“ (bit).

[3] RTT lze často zjistit pomocí utility ping, jejíž fungování bude objasněno detailněji v rámci cvičení a později v kapitole o síťové vrstvě.

3.7 Komunikační modely

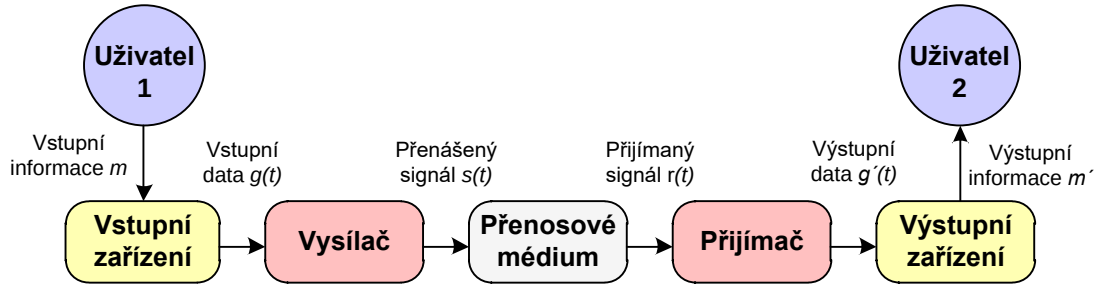
Základním účelem komunikace je vzájemná výměna informací mezi dvěma uživateli (procesy, tj. zdrojem a spotřebičem). Lze ji rozdělit na **dva typy komunikací**:

- **komunikace uvnitř sítí** včetně dohledu nad touto komunikací, tj. určitá servisní část komunikace v rámci síťové (internetové) infrastruktury,
- **komunikace mezi koncovými zařízeními**, tj. komunikace přes sítě, jak bylo naznačeno na **Obr. 3-2**.

Je třeba rozlišovat, že data a informace nejsou to stejné:

- **data** – reprezentace faktů, pojmů nebo instrukcí ve formalizované podobě vhodné pro komunikaci, interpretace (výklad) informace pro strojové zpracování.
- **informace** – význam, který mají data přiřazen, typicky pro uživatele.

Na obrázku **Obr. 3-4** je mezi dvěma uživateli (procesy) vyměňována informace nazvaná „*m*“. Informace *m* je pomocí vstupního zařízení reprezentována jako data *g(t),* ve formě časově proměnlivého signálu. V tomto okamžiku to ještě není signál vhodný pro vysílání a musí být „přeložen“ do podoby vhodné pro přenosové médium, tj. signálu *s(t),* což je úkol vysílače. Tento signál je již přenášen médiem a na jeho druhé straně se objeví jako signál *r(t)*, který může být odlišný od původního signálu *s(t)* následkem rušení či šumů v médiu. Signál *r(t)* je konvertován v přijímači zpět do tvaru výstupních dat g´(t), které mohou odpovídat vstupním datům přesně nebo přibližně. Nakonec je přes výstupní zařízení předána informace uživateli v podobě „*m´*“.



Obr. 3-4: Zjednodušené blokové schéma datové komunikace (pro jednoduchost přenos zaznačen pouze jednosměrně)

Tři základní úkoly vedoucí k realizaci přenosu informací spočívají v:

- vlastní přenos informace, tj. kódování dat a jejich přizpůsobení pro telekomunikační kanál,
- vyhledání cesty spojení dvou uživatelů v síti (tzv. směrování),
- použití vhodného způsobu komunikace, řízení výměny dat (tzv. protokoly).

Komunikační řetězec se stará zejména o:

- ***Řízení výměny informací,*** tj. způsob organizace přenosu dat mezi zdrojem a cílem informace.
- ***Definice rozhraní*** – zařízení musí mít definováno rozhraní s přenosovým systémem včetně tvaru a velikosti signálů.
- ***Synchronizaci***, tj. mezi přijímačem a vysílačem musí existovat určité formy časového sjednocení, tak, aby bylo možno rozeznat jednotlivé signálové elementy.
- ***Formátování zpráv***, tj. unifikace způsobu sestavení obsahu zprávy, tak aby si odpovídající části v komunikačním modelu „rozuměly".
- ***Adresování a směrování***, tj. jednoznačný způsob určení cíle a nalezení cesty k němu.

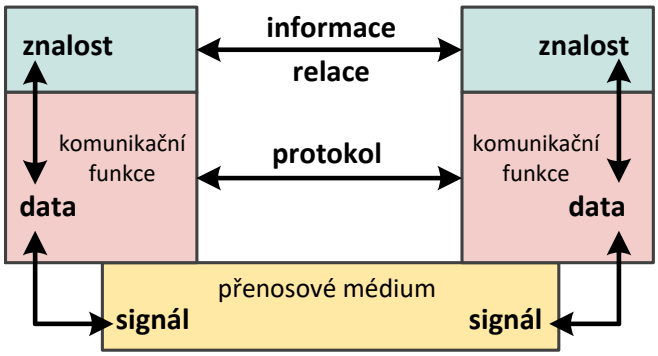
Komunikační řetězec zpravidla umožňuje:

- ***Vícenásobné využití přenosových systémů***, tj. komunikační řetězec je sdílen více uživateli, případně více procesy.
- ***Řízení systému,*** tj. konfigurace, dohled, reakce na chyby a přetížení, apod.
- ***Detekci a korekci chyb***, které mohou vznikat během přenosu.
- ***Zotavení se ze ztrát informací v komunikačním systému***, systém musí být schopen vrátit se minimálně do stavu, který byl před ztrátou informace.
- ***Řízení přenosu***, tj. zajištění, aby nedocházelo k zahlcení systému nadměrným množstvím přenášené informace.
- ***Ochranu zpráv***, tj. posílaná data může přijímat pouze zvolený příjemce.

3.8 Architektura komunikace systémů – vrstvy a protokoly

3.8.1 Horizontální a vertikální komunikace

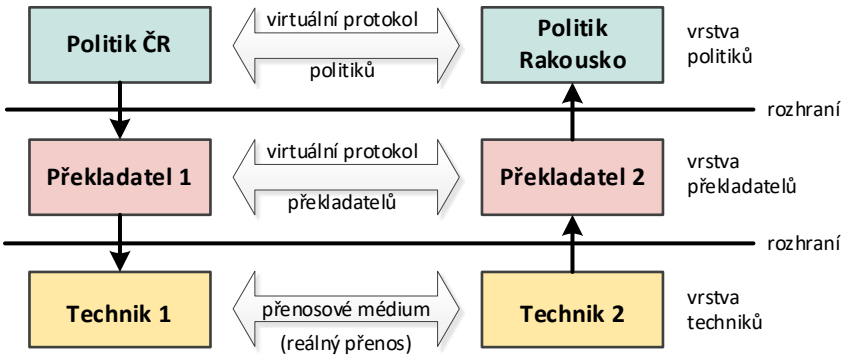
Z **Obr. 3-5** je patrné, že přenos informace mezi komunikujícími stranami se děje dle dohodnutých pravidel – **protokolu** (a samozřejmě přes přenosové médium).



Obr. 3-5: Základní model komunikace – principiální schéma

Aspekty komunikace, které je nutno v sítích při přenosu od jednoho uživatele k druhému řešit, lze shrnout do sedmi oblastí. K popisu využijeme zjednodušený nákres na **Obr. 3-6**, kde jsou pro prvotní přiblížení naznačeny pouze tři vrstvy. Představme si dva politiky, z nichž jeden je v ČR a druhý např. v Rakousku, přičemž první mluví pouze česky a druhý pouze německy – a je nutné, aby se nějakým způsobem domluvili a jeden druhému předal určitou informaci. Komunikace mezi nimi bude probíhat na celkem sedmi úrovních (vrstvách):

1) Politici jsou od sebe vzdáleni, musí mezi nimi tedy existovat komunikační systém, který jejich hovor přenese. Mezi politiky a komunikačním systémem musí existovat rozhraní, které přizpůsobí jejich řeč systému – vstup do komunikačního systému (tzv. aplikační vrstva).
2) V systému musí dále existovat zařízení umožňující překlad různých způsobů vyjadřování – jazyků, abeced (řešení problému vnitřní a vnější komunikace – tzv. prezentační vrstva).
3) Protože každý z politiků pracuje např. v nějakém institutu, musí existovat způsob, jak nasměrovat hovor na příslušného politika (tzv. relační vrstva).
4) Při navazování spojení si "spojovatelky" obou institutů potvrdí, že spolu komunikují, že mezi jejich instituty existuje koncové spojení v síti a toto udržují po dobu hovoru politiků (tzv. transportní vrstva).
5) Při spojení mezi instituty musí komunikační systém vyhledat nejvhodnější cestu hovoru mezilehlou sítí, tj. musí provést směrování v síti. Jednotlivé uzly sítě (spojovací body) spolu komunikují a vyhledávají optimální spojení (tzv. síťová vrstva).
6) V systému existují zařízení, která přenášený hovor vždy mezi uzly „očistí“ od šumů a přeslechů, a tak jej zbaví chyb vzniklých během přenosu mezi spojovacími uzly (tzv. spojová vrstva).
7) Systém musí obsahovat zařízení, které umí přizpůsobit „probíhající hovor“, přenosovému médiu, tj. zabezpečit fyzický přístup signálu na přenosové médium (tzv. fyzická vrstva).



Obr. 3-6: K vysvětlení průběhu komunikace mezi dvěma politiky

Na základě výše uvedeného popisu vrstev je možné vytvořit architekturu komunikace systémů a komunikační sítě. V každém komunikačním systému pak existují dva způsoby komunikace:

- ***Vertikální komunikace*** – probíhá formou požadavků od nejvyšší úrovně směrem k nejnižší a naopak. Komunikující strany zpravidla vnímají komunikaci horizontálně, ale ve skutečnosti vždy probíhá vertikálně (kromě nejnižší úrovně), přes jednotlivé úrovně systému. Např. politik, který chce komunikovat, aktivuje systém zadáním jména, adresy a jazyka druhého politika a tuto informaci předá do systému. Jakmile ji obdrží překladatel (nižší úroveň), převede informaci o jazyku do formy, kterou znají všichni překladatelé v systému a spolu s adresou a jménem tuto pošle dále. Princip komunikace je obdobný i níže. Obecně lze říci, že tento postup umožňuje připravovat znalosti tak, aby mohly být odeslány přes přenosové médium.
- ***Horizontální komunikace*** probíhá na dvou odpovídajících si úrovních, mezi nimiž musí existovat forma „společné řeči“. Např. politici mohou používat společné pojmy a zkratky, překladatelé se musí domluvit na jazyku, kterému oba rozumí, atd. Tato „společná řeč“ se nazývá protokol a musí existovat na každé dvojici vzájemně si odpovídajících úrovní komunikačního systému. Dále musí existovat na každé úrovni protokol domluvy na společném postupu, což spočívá v tom, že ke každé informaci převzaté od vyšší úrovně je přidána informace pro domluvu s protější úrovní. Opačně pak data, která přijdou od nižší úrovně, jsou očištěna od informací sloužících pro řízení této dané úrovně a až poté předána na úroveň vyšší, více viz dále. Horizontální komunikace je s výjimkou fyzické vrstvy vždy pouze virtuální.

3.8.2 Protokol, vrstvy a síťové modely

Protokol je množina pravidel určujících formát a význam rámců, paketů a zpráv vyměňovaných mezi partnerskými vrstvami. Komunikační protokol vytváří dojem, že spolu přímo komunikují entity na stejné vrstvě.

Klíčové prvky protokolu jsou

- ***syntax (skladba)*** – zahrnuje formáty dat a úrovně signálů, tj. jak mají signál či data vypadat,
- ***sémantika (≈ význam)*** – obsahuje řídicí informace pro spolupráci řízení vrstev a opravu chyb,
- ***časování*** – zahrnuje rychlost výměny dat, počet opakování a jejich posloupnosti (zpravidla velmi záleží na pořadí výměny zpráv).

Platí, že řídicí informace obsažené v protokolu jsou vždy primárně určené a srozumitelné pouze stejnolehlé vrstvě na vzdáleném systému. Z důvodu kompatibility je důležité, aby byl protokol detailně specifikován, jak bylo zmíněno již v kap. 3.1. Protokol zpravidla není v konkrétní vrstvě pouze jeden. **Soubor protokolů** (*protocol stack* nebo *protocol suite*) pak řeší celou komunikaci komplexně. Každý z protokolů ze sady řeší pouze určitý podproblém komunikace. Nižší vrstvy zpravidla poskytují **služby** vyšším vrstvám a mezi vrstvami dochází k vertikální komunikaci a předávání informací, aby bylo možné tyto služby realizovat.

Protokoly jsou obvykle implementovány v rámci jednoho počítače pomocí určitého počtu procesů, které spolu komunikují pomocí front a volání funkcí. Pro správnost funkce protokolu je nutné použít jeden nebo více časovačů. Protokoly používají rozhraní na operačním systému počítače. Nižší vrstvy se pak neobejdou bez hardwarových komponent (např. anténa), jak bude popsáno níže.

3.9 Základní popis referenčního modelu ISO/OSI

Síťová komunikace byla již od začátků svého budování založena na vrstvách, tak jak je popsáno výše. Z počátku však vznikaly různé vzájemně nekompatibilní a uzavřené architektury. Tyto sítě byly zpravidla využívány na úrovni jedné společnosti a nebylo možné je vzájemně propojit.

Postupem času rostl tlak na vznik otevřeného standardu, který by umožnil propojení sítí. Na základě výskytu dříve existujících okruhů problémů při síťové komunikaci byl tedy navržen v International Organization for Standardization (ISO) model OSI (*Open System Interconnection Reference Model*), který podchycuje všechny nezbytné aspekty komunikace. Stal se výchozím modelem komunikační architektury pro počítačově řízenou výměnu dat. Tento model položil *teoretický (a vědecký)* základ pro realizaci veřejných datových sítí.

Obr. 3-7 ukazuje řazení všech 7 vrstev, jejich popis je uveden dále. Norma ISO/OSI nespecifikuje přímo to, jak by měla implementace systémů vypadat. Uvádí všeobecné principy sedmivrstvé síťové architektury. Jedná se zejména o účel každé vrstvy, její funkci, služby poskytované vyšší vrstvě a také služby požadované od vrstvy nižší.

Počet vrstev v modelu (7) vznikl jako kompromis při jednáních o vzniku OSI. Vrstvy se číslují od nejnižší vzestupně. Jsou to:

1. **fyzická vrstva**
2. **spojová** (nesprávně, ale často i jako linková) **vrstva**
3. **síťová vrstva**
4. **transportní vrstva**
5. **relační vrstva**
6. **prezentační vrstva**
7. **aplikační vrstva**

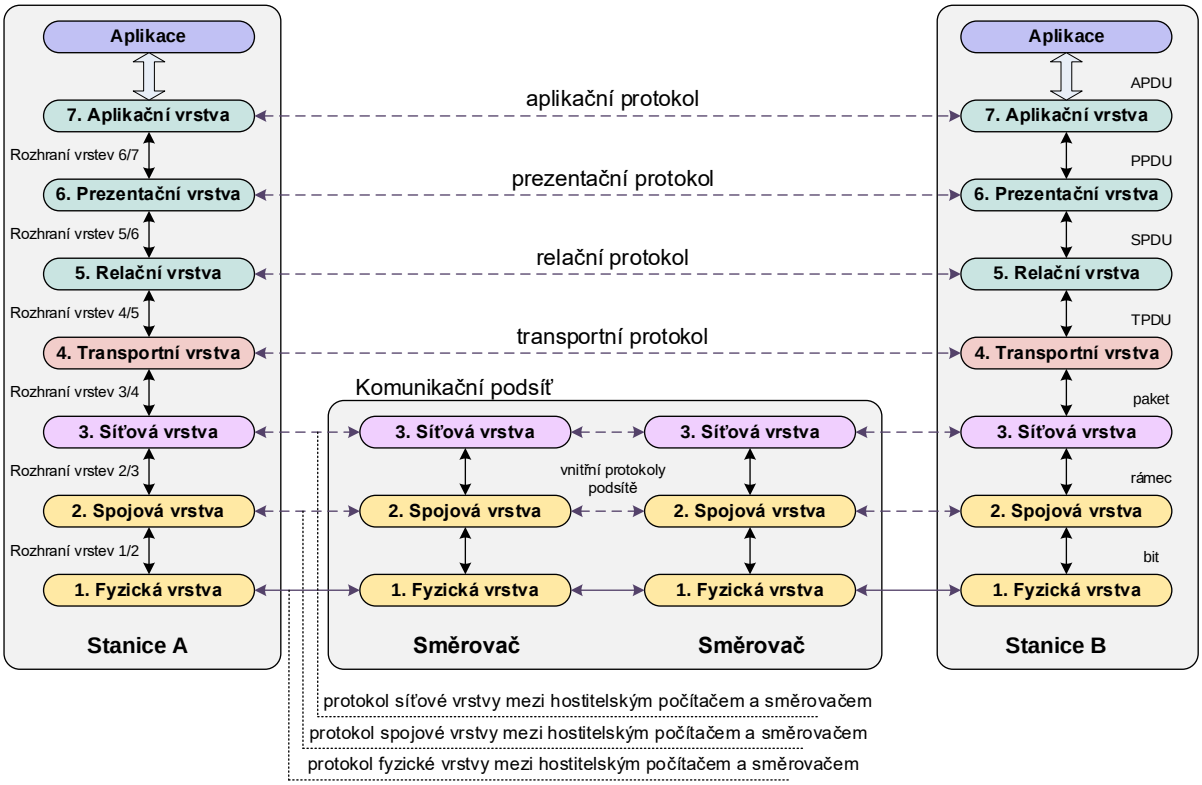
Implementace vrstev může být softwarová nebo hardwarová. Nejčastější je situace, kdy **nejnižší dvě vrstvy jsou zejména hardwarové, třetí vrstva může být dominantně hardwarová nebo softwarová** a všechny **vyšší vrstvy jsou již zejména softwarové**.

Vrstvy 1-4 lze shrnout do označení „**poskytovatelé transportní služby**“, vrstvy 5-7 jako „**uživatelé transportní služby**“.

Častější je však rozdělení na skupinu vrstev 1-3 a 4-7, tak jako na **Obr. 3-7**. Vrstvy 1-3 jsou označovány jako ***lokální***, protože se starají o komunikaci přes lokální síťovou infrastrukturu (fyzická vrstva, detekce a oprava chyb, směrování) – tvoří **komunikační podsíť**. Vrstvy 4-7 jsou ***koncové*** a umožňují vytvoření vazby komunikujících aplikací. Tyto vrstvy (4-7) se nachází především v koncových uzlech, zatímco vrstvy (1-3) jsou i ve většině mezilehlých prvků.

Význam zkratek použitých v **Obr. 3-7** :

- **APDU** (*Application Protocol Data Unit*), datová jednotka aplikačního protokolu,
- **PPDU** (*Presentation Protocol Data Unit*), datová jednotka prezentačního protokolu,
- **SPDU** (*Session Protocol Data Unit*), datová jednotka relačního protokolu,
- **TPDU** (*Transport Protocol Data Unit*), datová jednotka transportního protokolu.



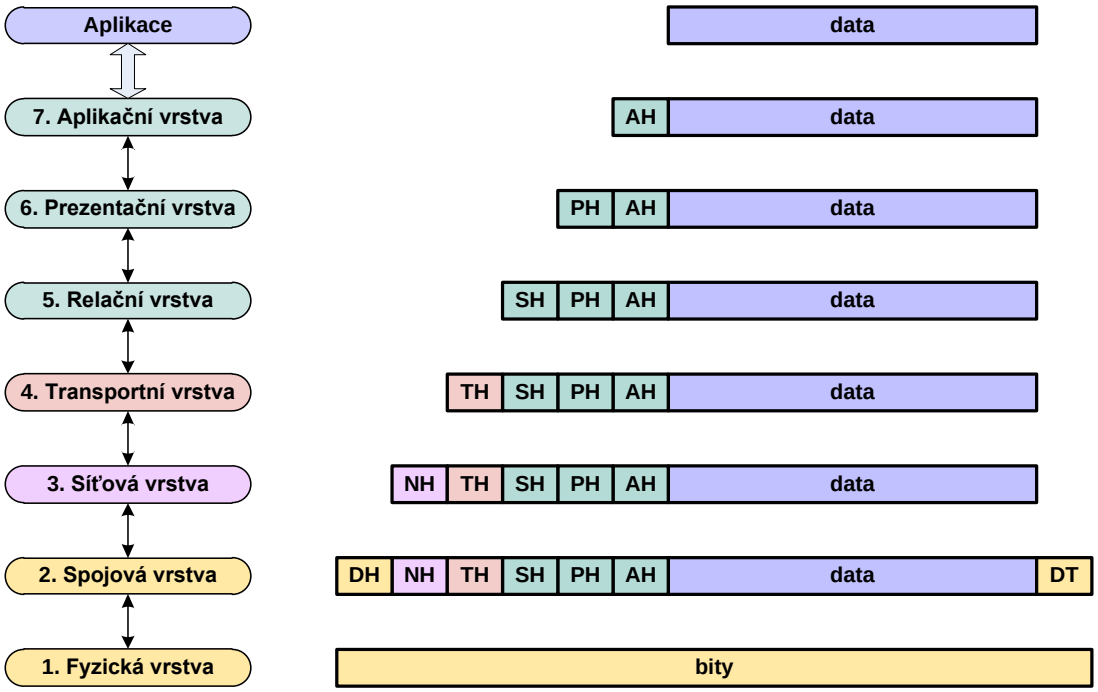
Obr. 3-7: Architektura sítě založené na modelu ISO/OSI

Z **Obr. 3-7** je patrné, že partnerská komunikace s jiným koncovým uživatelem je pouze zdání. Ve skutečnosti uživatelská data putují přes mnoho bodů, počínaje aplikační vrstvou na jednom konci a konče aplikační vrstvou na druhé koncové stanici. Při této cestě procházejí všemi nižšími vrstvami. Na mezilehlém prvku, který má oddělené části (rozhraní) pro každou stranu, s kterou komunikuje, dochází k průchodu až do síťové vrstvy, kde se provede rozhodnutí o dalším směrování dat a následně na výstupním rozhraní zpráva opět „sestoupí“ až na fyzickou vrstvu.

Z dalšího obrázku (**Obr. 3-8**) je zřejmé, že s průchodem od vyšší k nižší vrstvě se zvětšuje protokolová datová jednotka (PDU) o záhlaví jednotlivých vrstev. Tato operace se nazývá **zapouzdřování** (zabalení). V cílovém systému pak dochází postupně v jednotlivých vrstvách k **odpouzdřování** (rozbalení) zprávy.

Význam zkratek použitých v **Obr. 3-8**:

- AH (*Application Header*), záhlaví aplikační vrstvy,
- PH (*Presentation Header*), záhlaví prezentační vrstvy,
- SH (*Session Header*), záhlaví relační vrstvy,
- TH (*Transport Header*), záhlaví transportní vrstvy,
- NH (*Network Header*), záhlaví síťové vrstvy,
- DH (*Data-Link Header*), záhlaví spojové vrstvy,
- DT (*Data-Link Trailer*), zápatí spojové vrstvy.



Obr. 3-8: Znázornění tvorby PDU v jednotlivých vrstvách

3.9.1 Aplikace

Aplikace jsou koncové **procesy** (uživatelské úlohy) v referenčním modelu OSI, které jsou „rozptýleny“ po síti a mají potřebu komunikovat mezi sebou za účelem splnění úloh. Na tuto komunikaci lze klást další požadavky, jež přímo nebo nepřímo vyplývají ze samotné aplikace a týkají se spolehlivosti, reakční schopnosti a rozlišitelnost systémů. Uživatel si logické prostředí sítě často ani neuvědomuje, protože vlastní aplikace jej skrývá. Pro dnešní masové využití je to jistě nejvhodnější řešení.

3.9.2 Shrnutí hlavních úkolů jednotlivých vrstev ISO/OSI

Aplikační vrstva, jakožto nejvyšší vrstva, především zpřístupňuje informačním systémům celé prostředí ISO/OSI.

Prezentační vrstva koordinuje kódování a syntaxi vyměňovaných dat. Provádí zejména tyto úkony: transformace kódování, šifrování, komprese.

Relační vrstva především poskytuje informačním systémům nástroje pro řízení a synchronizaci jejich dialogu.

Transportní vrstva se stará o adresování konkrétní služby, segmentace a znovu-skládání dat, řízení spojení mezi komunikujícími aplikačními protokoly, řízení toku dat a řízení chybových stavů.

Úkoly **síťové vrstvy** lze shrnout zejména do logického adresování a směrování mezi jednotlivými sítěmi.

Spojová vrstva má na starost zejména vytváření rámců, adresování v rámci dané sítě, řízení toku dat, řízení chybových stavů a přístupové metody ke sdílenému médiu.

Nejníže je **fyzická vrstva**, která se zabývá těmito úkoly: fyzické charakteristiky médií a rozhraní, reprezentace bitů, přenosová rychlost, synchronizace mezi vysílačem a příjemcem, přizpůsobení se charakteru kanálu a topologii sítě (bod-bod, vícebodová) a přenosový režim z hlediska oboustrannosti komunikace.

Výše uvedené pojmy, úkoly a problémy u jednotlivých vrstev budou popsány blíže později v rámci podrobnějšího popisu jednotlivých vrstev ISO/OSI a to především v návaznosti na model TCP/IP.

3.10 Základní popis síťového modelu TCP/IP

3.10.1 Vazba mezi RM OSI a modelem TCP/IP

Soustava protokolů TCP/IP je původně „rivalem" obecného sedmivrstvého referenčního modelu ISO/OSI. TCP/IP je označení dvou přenosových protokolů, konkrétně protokolů TCP (*Transmission Control Protocol*) a IP (*Internet Protocol*). Ve skutečnosti ale zkratka **TCP/IP označuje celou soustavu protokolů** a ucelenou soustavu názorů o tom, jak by se počítačové sítě měly budovat, a jak by měly fungovat. Sada zahrnuje i vlastní představu o tom, jak by mělo být síťové programové vybavení členěno na jednotlivé vrstvy, jaké úkoly by tyto vrstvy měly plnit, a také jakým způsobem by je měly plnit – tedy jaké konkrétní protokoly by na jednotlivých úrovních měly být používány. TCP/IP je tedy stejně jako RM ISO/OSI síťovou architekturou, navíc, jak se ukázalo postupem času, velmi **vhodnou pro praktickou implementaci**.

Hlavní odlišnosti mezi modely ISO/OSI a TCP/IP vyplývají především z rozdílných výchozích předpokladů a postojů jejich tvůrců. Při koncipování referenčního modelu **ISO/OSI** měli hlavní slovo zástupci spojových organizací, kteří kladli **důraz na vlastnosti sítě** (především spojovaný a spolehlivý charakter služeb) s tím, že k síti **připojované hostitelské počítače** budou mít **relativně jednoduchou úlohu**. Později se ale ukázalo, že např. právě v otázce zajištění spolehlivosti to není nejšťastnější řešení – že vyšší vrstvy nemohou považovat spolehlivou komunikační síť za dostatečně spolehlivou pro své potřeby, a tak se snaží zajistit si požadovanou míru spolehlivosti vlastními silami. V důsledku toho se pak zajišťováním spolehlivosti do určité míry zabývá vlastně každá vrstva referenčního modelu ISO/OSI.

Tvůrci sady **TCP/IP** [4] naopak vycházeli z předpokladu, že **zajištění spolehlivosti je především problémem koncových účastníků** komunikace, a mělo by tedy být řešeno až na úrovni transportní vrstvy. V TCP/IP jsou tedy některé funkce přeneseny až na úroveň koncových stanic. Komunikační síť pak podle této představy nemusí ztrácet část své přenosové kapacity na zajišťování spolehlivosti (na potvrzování, opětné vysílání poškozených paketů atd.), a může ji naopak plně využít pro vlastní datový přenos. V komunikační síti může docházet ke ztrátám přenášených paketů, a to bez varování a bez snahy o nápravu. Komunikační síť by ovšem *neměla* zahazovat pakety bezdůvodně. Měla by naopak vyvíjet maximální snahu přenášené pakety doručit (v angličtině se v této souvislosti používá termín

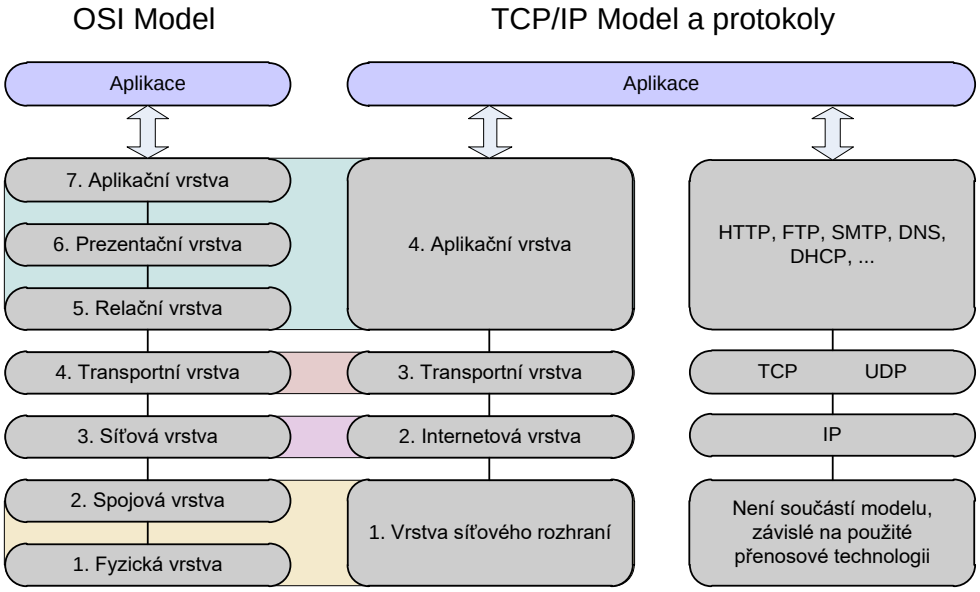
[4] Počátky TCP/IP spadají do konce 60. let a jsou úzce spojeny s činností agentury ARPA (*Advanced Research Projects Agency*), později s název DARPA (*Defence Advanced Research Projects Agency*), Ministerstva obrany USA, které si nové protokoly nechalo vyvinout pro svou počítačovou síť ARPANET. Na vývoji celé soustavy protokolů, financovaném prostřednictvím grantů Ministerstva obrany (účelových dotací na výzkum), se pak podílela počítačově orientovaná pracoviště předních univerzit USA. Svou dnešní podobu získaly nové protokoly zhruba v letech 1977-79, a brzy poté na ně začala postupně přecházet i vlastní síť ARPANET, která se posléze stala zárodkem a páteří celého konglomerátu sítí, dnes nazývaného *Internet*.

best effort), a zahazovat pakety až tehdy, když je skutečně nemůže doručit - tedy např. když dojde k jejich poškození při přenosu, když pro ně není dostatek místa ve vyrovnávací paměti pro dočasné uložení, v případě výpadku spojení apod. Na rozdíl od referenčního modelu ISO/OSI tedy **TCP/IP předpokládá jednoduchou a rychlou komunikační podsíť, ke které se připojují inteligentní hostitelské počítače.**

TCP/IP předpokládá nespojovaný charakter přenosu v komunikační síti – tedy jednoduchou datagramovou (nespojovanou) službu a obsahuje jen čtyři vrstvy (graficky **Obr. 3-9**), viz následující kapitoly.

3.10.2 Vrstva síťového rozhraní (Network Interface Layer)

Nejnižší vrstva, (dle OSI spojová vrstva a fyzická vrstva dohromady) má na starosti vše, co je spojeno s ovládáním konkrétní přenosové cesty resp. sítě, a s přímým vysíláním a příjmem datových paketů. V rámci soustavy TCP/IP není tato vrstva blíže specifikována, neboť je zcela závislá na použité přenosové technologii. Vrstvu síťového rozhraní může tvořit relativně jednoduchý ovladač (*device driver*), je-li daný uzel přímo připojen například k lokální síti. Často se i z pohledu TCP/IP bavíme o spojové a fyzické vrstvě dle modelu ISO/OSI.



Obr. 3-9: Srovnání modelu RM ISO/OSI a TCP/IP

3.10.3 Internetová vrstva (Internet Layer)

Vyšší vrstva, která již není závislá na konkrétní přenosové technologii, označována též jako IP vrstva (*IP Layer*) podle toho, že je realizována pomocí protokolu IP (IP verze 4 / IP verze 6). Funkčně odpovídá **síťové vrstvě** ISO/OSI a často je proto nazývána i jako vrstva síťová. Úkolem této vrstvy je, aby se jednotlivé pakety dostaly od odesílatele až ke svému skutečnému příjemci, zpravidla přes mezilehlá zařízení (směrovače). Vzhledem k nespojovanému charakteru přenosů v TCP/IP je na úrovni této vrstvy zajišťována jednoduchá (tj. nespolehlivá) **datagramová služba**. Síťová vrstva se však musí **vyrovnávat** i **s** konkrétními **odlišnostmi jednotlivých dílčích sítí** – například s odlišným charakterem

adres, s různou maximální velikostí přenášených paketů resp. rámců a jejich formátem a s odlišným charakterem nižší vrstvou poskytovaných přenosových služeb. Pro každou síť či každý přenosový kanál, na který je brána připojena, má samostatný ovladač na úrovni vrstvy síťového rozhraní.

3.10.4 Transportní vrstva (Transport Layer)

Třetí vrstva, též označována jako TCP vrstva (*TCP Layer*), neboť je nejčastěji realizována právě protokolem **TCP (*Transmission Control Protocol*)**. Hlavním úkolem této vrstvy je zajistit přenos mezi dvěma koncovými účastníky, kterými jsou v případě TCP/IP přímo aplikační programy (jako entity bezprostředně vyšší vrstvy). Podle jejich nároků a požadavků může transportní vrstva regulovat tok dat oběma směry, zajišťovat případně spolehlivost přenosu a také měnit nespojovaný charakter přenosu (v síťové vrstvě) na spojovaný. Nejčastěji využívaným protokolem transportní vrstvy TCP/IP je patrně protokol TCP. Dalším používaným protokolem na úrovni transportní vrstvy je pak zejména protokol **UDP (*User Datagram Protocol*)**, který na rozdíl od TCP nezajišťuje spolehlivost přenosu ani další sofistikovanější funkce známé z TCP. Samozřejmě je využíván pouze aplikacemi, které si spolehlivost na úrovni transportní vrstvy nepřejí nebo nemohou dovolit, či to není efektivní. To platí např. v případě služeb vyžadujících přenos pouze dvou zpráv (paketů), ve formě dotaz-odpověď.

3.10.5 Aplikační vrstva (Application Layer)

Nejvyšší vrstva, jejími entitami jsou jednotlivé aplikační programy, které na rozdíl od referenčního modelu ISO/OSI komunikují přímo s transportní vrstvou. Případné prezentační a relační služby, které v modelu ISO/OSI zajišťují samostatné vrstvy, si zde musí jednotlivé aplikace realizovat samy (pokud je vyžadují). Pokud aplikace prezentační nebo relační vrstvu nepotřebuje, nevzniká žádná (zbytečná) režie. Na této vrstvě se můžeme setkat s velkým množstvím protokolů, např.:

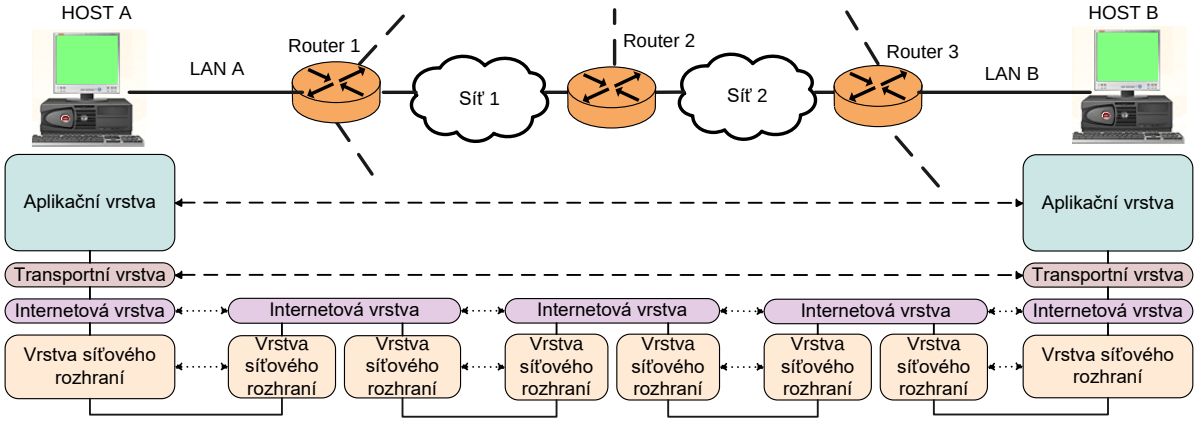
- **HTTP** (*Hypertext Transfer Protocol*), základní přenosový protokol ve WWW (*World Wide Web*) prostředí,
- **FTP** (*File Transfer Protocol*), protokol zejména pro přenos souborů,
- **SMTP** (*Simple Mail Transfer Protocol*), hlavní protokol pro přenos elektronické pošty,
- **DNS** (*Domain Name System*), protokol pro práci se jmennými názvy (adresami) v celém Internetu,
- **DHCP** (*Dynamic Host Configuration Protocol*), protokol pro centralizovanou správu IP adres na lokální síti.

3.10.6 Filozofie vzájemného propojování sítí pomocí TCP/IP

Řešení vzájemného propojování sítí je jedním z prvotních příčin vzniku celé soustavy protokolů TCP/IP. Filozofie TCP/IP od začátku usiluje o co **nejuniverzálnější propojení sítí různých typů** – od lokálních sítí typu Ethernet, Token Ring apod., přes veřejné datové sítě, až po rozlehlé sítě celosvětového dosahu. Klade si přitom za cíl umožnit každému uzlu komunikovat s kterýmkoli jiným uzlem, bez ohledu na to, zda mezi nimi existuje přímé spojení, nebo zda jsou například tyto uzly v různých sítích, které jsou vzájemně propojeny jednou nebo několika dalšími sítěmi. Výsledkem je pak jediná soustava vzájemně

propojených sítí, v terminologii TCP/IP označovaná obecně jako *Internetworking*. Z pohledu uživatele by vnitřní struktura této soustavy sítí měla být irelevantní - uživatelé, resp. jejich aplikační programy, se mohou na celý *Internetworking* dívat jako na jedinou velkou síť, ke které jsou připojeny jednotlivé koncové počítače - v terminologii TCP/IP označované jako hostitelské počítače (*host computers, hosts*).

Ve skutečnosti je výsledná **soustava** (*Internet*) tedy jen konglomerátem (dílčích) sítí stejného či různého typu, vzájemně **propojených na úrovni síťové vrstvy** pomocí zařízení označených někdy nesprávně jako brány (*gateway*), správně termínem IP **směrovač** (tj. *IP router*). Výhodou IP protokolu je, že **zavádí jednotný formát adres a způsob adresování i jednotný formát přenášených dat na úrovni síťové vrstvy**. Na **Obr. 3-10** je ukázka propojení dvou stanic přes Internet (Host A, resp. Host B, kteří sídlí na lokální síti LAN A, resp. LAN B). V dolní části je vidět na jakých vrstvách jednotlivé prvky pracují, spojitá čára značí reálný průchod dat sítí od Hosta A k Hostu B, přerušovaná čára značí virtuální spoje na jednotlivých vrstvách. (Přerušované čáry vedoucí ze směrovačů v horní části značí cesty k dalším částem Internetu, které v obrázku nejsou pro zjednodušení nakresleny.)



Obr. 3-10: Ukázka propojení sítí v rámci Internetu

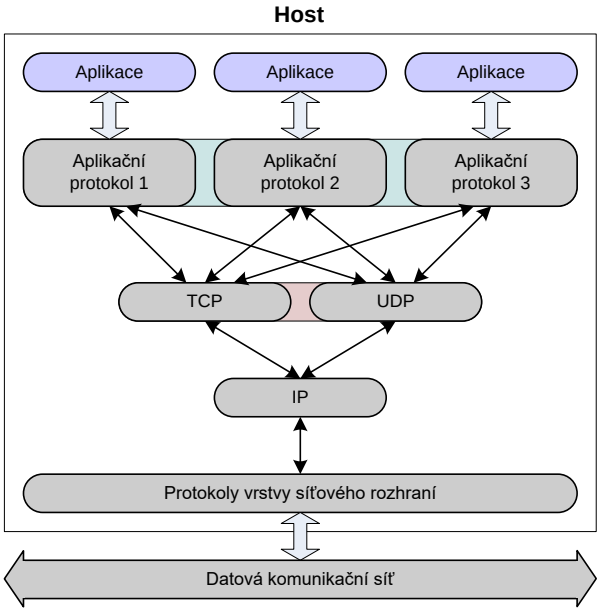
Z hlediska adresování jsou v TCP/IP využívány vždy dva základní typy adres uzlů. Každé zařízení má (zpravidla od výrobce) tzv. fyzickou adresu, která je důležitá pro adresování v rámci konkrétní sítě, v které se toto zařízení nachází. Tyto adresy jsou závislé na konkrétní technologii sítě a existuje proto více typů těchto adres. Fyzické adresy jsou spjaty s konkrétním hardware. Za pomoci těchto adres není možné komunikovat mezi různými sítěmi, a proto jsou v TCP/IP definovány vyšší (abstraktní) adresy, platné globálně a v libovolné síti, tzv. IP adresy. Tyto adresy jsou pouze logické, používané především na úrovni Internetové vrstvy a jsou spjaty především se softwarem.

V případě přenosu, který je naznačen na **Obr. 3-10**, se tak při komunikaci stanic Host A a Host B logické adresy na úrovni Internetové vrstvy nemění po celou dobu přenosu, adresátem je stále koncová stanice B, resp. její IP adresa. Jiná je ale situace v případě fyzických adres. Tyto adresy se v průběhu přenosu mezi stanicemi mění podle toho, přes kterou síť jednotky procházejí a jako cíl je vždy lokálně platná fyzická adresa.

V souvislosti s adresami je nutné ještě zmínit DNS jména, kterým se stejně jako výše uvedeným adresám budeme věnovat podrobněji později. Tato jména tvoří vlastně jmenný odkaz na určitý uzel, stanici či službu a bez jejich využití si lze jen obtížně představit dnešní sítě. Jako příklad DNS jména můžeme uvést např. seznam.cz.

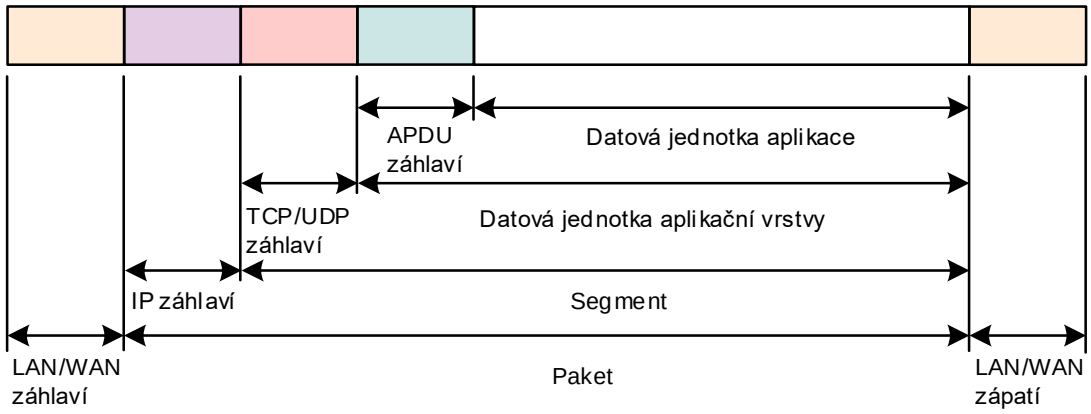
3.10.7 Souběh aplikací v rámci TCP/IP a zapouzdřování

Na **Obr. 3-11** jsou znázorněny tři současně běžící aplikace, např. www prohlížeč, emailový klient a ftp klient. Každá z těchto aplikací využívá příslušný **aplikační protokol**, při této volbě služeb jsou to např. HTTP, SMTP a FTP, které jsou schopny uživatelovy požadavky zformulovat do zprávy, kterou bude následně schopna druhá strana komunikace zpracovat, tzv. Application PDU (*Protocol Data Unit*), tedy protokolová datová jednotka aplikační vrstvy, jednoduše nazývána **data**. Tyto údaje (pocházející z libovolného z aplikačních protokolů), a s příslušným záhlavím, jsou předávány **transportní vrstvě**, při volbě protokolů HTTP, SMTP a FTP pak často protokolu TCP. Protokol UDP by byl typicky použit v případě volby jiných aplikací, avšak jeho použití zde není teoreticky vyloučeno. Přidáním záhlaví TCP vznikne TCP PDU, tedy protokolová datová jednotka transportní vrstvy, též nazývaná **segment**[5]. Transportní vrstva takto vzniklou jednotku předá **Internetové vrstvě**, v našem příkladu IP protokolu a přidáním IP záhlaví vznikne IP datagram, nejčastěji nazýván jako **paket**. Další postup závisí na použitém **síťovém rozhraní**, tedy v případě připojení stanice (hosta) na LAN se paketu přidá LAN záhlaví (*header*) a LAN zápatí (*trailer*) a vznikne **rámec**. Tomuto ději se říká zapouzdřování (*encapsulation*) a struktura výsledné jednotky (rámce) je patrná z **Obr. 3-12**.



Obr. 3-11: Znázornění souběžného fungování více aplikací v rámci TCP/IP a vazeb mezi vrstvami

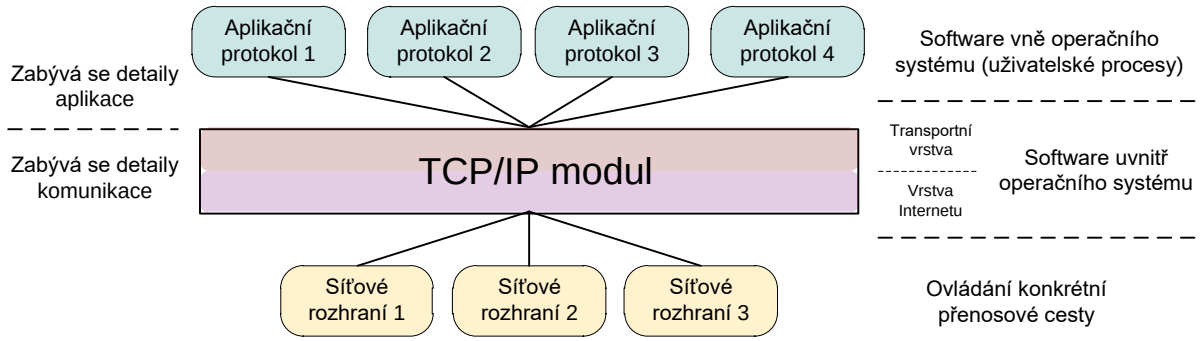
[5] Pojem segment je používán převážně v souvislosti s protokolem TCP. V případě použití druhého nejčastějšího transportního protokolu – UDP – se používá pro PDU název **datagram**, což je datová jednotka, se kterou se můžeme potkat následně i na Internetové vrstvě.



Obr. 3-12: Zapouzdřování (*encapsulation*) uživatelských dat jednotlivými vrstvami TCP//IP

3.10.8 Softwarový pohled na TCP/IP

V rámci operačního systému bývá protokolová sada TCP/IP implementována prostřednictvím tzv. **TCP/IP modulu.** Tento modul je pak schopný komunikovat s více vyššími protokoly a i s více fyzickými rozhraními konkrétního stroje. Tuto situaci lze znázornit prostřednictvím **Obr. 3-13**. V obrázku jsou naznačeny i hranice mezi jednotlivými vrstvami. **Nejvyšší (aplikační) vrstva je často implementována až v rámci konkrétního softwaru**, který ji využívá ke své funkci. Např. ftp server bude mít implementovaný ftp aplikační protokol, webový prohlížeč protokol http apod. **IP modul, který se skládá z transportní a síťové vrstvy, bývá běžně součástí operačního systému.** TCP/IP modul následně spolupracuje s konkrétními přenosovými cestami, které má daný stroj k dispozici – může jich být i více.

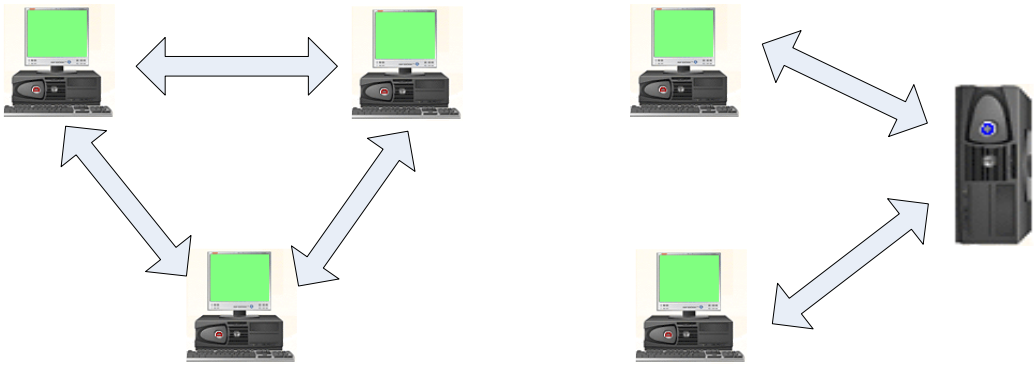


Obr. 3-13: Organizace softwarového vybavení na podporu sady TCP/IP

3.11 Základní způsoby komunikace z pohledu její organizace

Základní typy komunikačních systémů z pohledu vnitřní organizace komunikace jsou ***klient-server*** a ***klient-klient*** (*peer-to-peer* = P2P), viz **Obr. 3-14**.

Systémy **P2P** jsou založeny na skutečnosti, že všechny **stanice** v rámci systému **jsou si rovny**, zatímco model klient-server rozlišuje klientské stanice, které zasílají své požadavky na zpracování serverovým stanicím. Na P2P systém, se lze dívat jako na systém, jehož všechny stanice **obsahují jak klientskou, tak i serverovou část** (*servent*).



Obr. 3-14: a) Peer-to-Peer komunikace b) Klient-server komunikace

4 Principy komunikačních technik

4.1 Způsoby přenosu informace (dat)

Pro přenos informace mezi jejím zdrojem a cílem existuje několik základních způsobů přenosu, které jsou voleny zpravidla na základě povahy signálu. Např. hovorový signál má malé mezery mezi přenášenou informací, je velmi citlivý na různá zpoždění jednotlivých intervalů řeči a má vysokou nadbytečnost. Naproti tomu přenos dat mezi počítači se děje obvykle v dávkách, musí být velmi spolehlivý a zpoždění mezi jednotlivými částmi není až tak kritické. Pro tyto, ale i další, druhy signálu existují následující způsoby spojování:

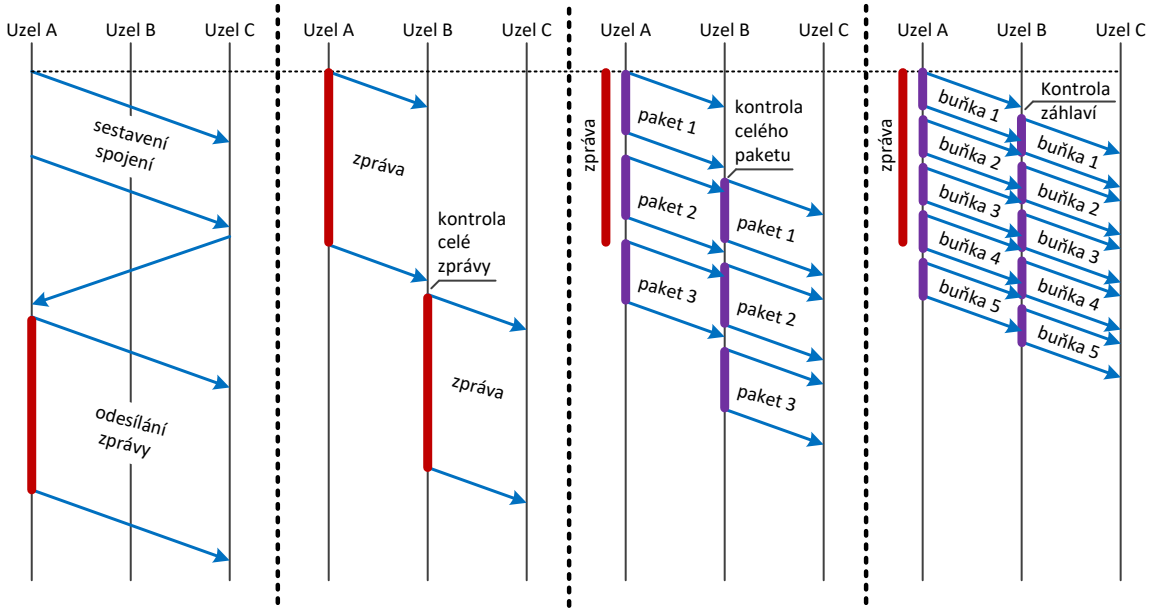
Komutace okruhů (circuit switching) **–** vytváří se fyzické spojení mezi koncovými účastníky (existuje mezi nimi dočasná přenosová cesta). Fyzické spojení je realizováno i uvnitř spojovacích uzlů. Důležitou vlastností komutace okruhů je nutnost sestavit spojení před vlastním přenosem informace, což představuje rezervaci prostředků a kapacit pro následný přenos. Tato operace posunuje dobu zahájení přenosu (může trvat cca 5 sekund). Z hlediska nákladů se jedná o „drahé“ spojení, neboť se platí za celou dobu sestaveného spojení, i když nedochází k přenosu informace po celou dobu. Tento způsob se využívá převážně pro přenos hovorových signálů (klasická pevná i mobilní telefonní síť), kde neexistoval dříve jiný způsob. Pro datové sítě se jedná o nepříliš obvyklý způsob přenosu.

Komutace zpráv (message switching) **–** nevytváří se fyzické spojení mezi přijímačem a vysílačem. Naproti tomu zdroj informace vyšle zprávu do prvního uzlu, kde se tato uloží, zkontroluje, a poté vyšle do dalšího uzlu směrem k příjemci dat. Tento způsob přenosu klade velké nároky na mezilehlé uzly, které musí být schopny celé zprávy uchovat ve svých pamětech (sít typu *store-and-forward*). Každá zpráva nese informaci o svém cíli. Výhodou metody je, že je vždy zatěžována pouze ta část sítě, kterou se právě daná zpráva přenáší. I tento způsob přenosu je pro datové sítě poměrně neobvyklý.

Komutace paketů (packet switching) **–** má obdobné vlastnosti jako komutace zpráv. Rozdílem je, že pokud je zpráva dlouhá, tak je rozdělena na bloky dat – pakety proměnné délky, s definovanou maximální délkou. Sítí jsou pak přenášeny jednotlivé pakety, obdobně jako v předcházejícím případě zprávy (a opět v režimu *store-and-forward*). V případě paketů však vznikají problémy s tím, že pořadí doručení paketů k cíli nemusí být dodrženo, a proto tato metoda vyžaduje dodatečné prostředky pro zajištění správnosti přenesení celé zprávy (pouhé zabezpečení proti chybám již nestačí). V současnosti nejčastější způsob přenosu v datových sítích.

Komutace buněk (cell switching) **–** zpravidla rozdělení na menší jednotky s přesně definovanou (fixní) délkou. Při přenosu se provádí kontrola pouze u záhlaví buňky (či rámce), čímž se předávání zrychlí. Proto dochází pouze k velmi malému zdržení přenášené jednotky v uzlu. Uživatel pak musí provádět veškeré kontroly přenesených dat samostatně. Tento způsob je možné využít k přenosu řečového signálu i klasických dat a využívá se např. u dnes již zastaralých ATM technologií. Výhodou oproti komutaci okruhů je úspora prostředků sítě, jelikož pro daný přenos je zpravidla blokována jen nezbytně nutná kapacita. Oproti komutaci zpráv a paketů je pak výhodou rychlejší odezva, velkou nevýhodou pak fixní velikost přenášené jednotky.

Pro lepší názornost si představme, že existuje telekomunikační nebo datová síť s jedním spojovacím (přepojovacím) uzlem a dvěma účastníky, viz **Obr. 4-1**. Uzel A reprezentuje generátor informací, uzel B spojovací mezilehlý bod a uzel C spotřebič informací. Na **Obr. 4-1** jsou pro každý výše uvedený způsob komutace naznačeny posloupnosti předávání informací, přičemž svislá osa představuje čas. Je patrné, že komutace okruhů (**Obr. 4-1**a) představuje nejpomalejší způsob přenosu informace a komutace zpráv (**Obr. 4-1**b) je o něco rychlejší, což však obecně platit nemusí. Rozdělením zprávy na pakety (**Obr. 4-1**c) je možné zkrátit dobu přenosu dat přes síť. Nejrychlejší metodou je v příkladu komutace buněk (**Obr. 4-1**d), síť v tomto případě nečeká na přijetí celé jednotky informace (buňky) a po kontrole záhlaví začne ihned aktuální buňku odesílat směrem k příjemci, avšak efektivita tohoto přenosu z jiných hledisek nemusí být vždy nejvyšší a velmi záleží na konkrétních podmínkách na přenosové trase.



a) komutace okruhů b) komutace zpráv c) komutace paketů d) komutace buněk

Obr. 4-1: Časové posloupnosti jednotlivých metod přenosu informace

4.2 Architektura a topologie sítí

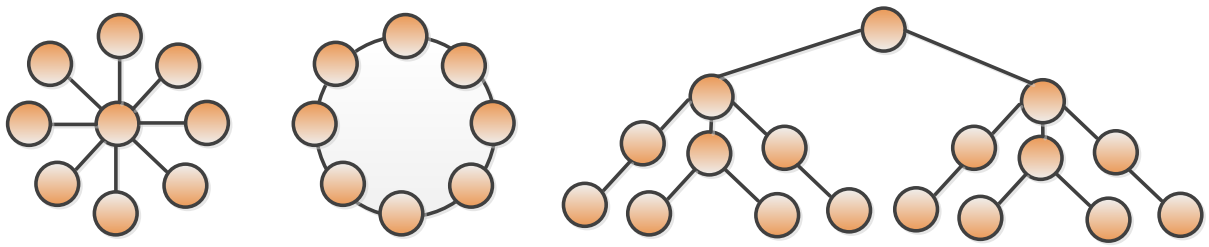
Z hlediska topologie sítí existují dva základní způsoby spojení uzlů sítě.

- **Dvoubodové spoje** (*point-to-point*), které jsou tvořeny řadou spojů, z nichž každý propojuje koncovou stanici s přepojovacím uzlem nebo tyto uzly navzájem. Informace je v tomto případě vyměňována nepřímo. Možné struktury sítě jsou schematicky uvedeny na **Obr. 4-2**.

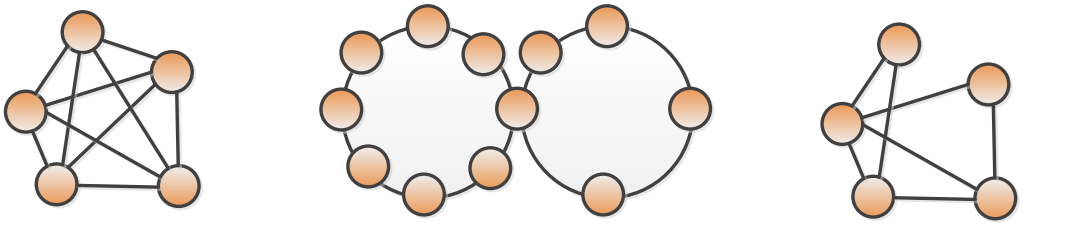
  Patří sem zejména topologie typu:

  - hvězda
  - kruh
  - strom
  - úplný polygon (úplné propojení, tj. každý přímo s každým dalším, *full mesh*)

- propojené kruhy
- obecná topologie (neúplný polygon, *partial mesh*).



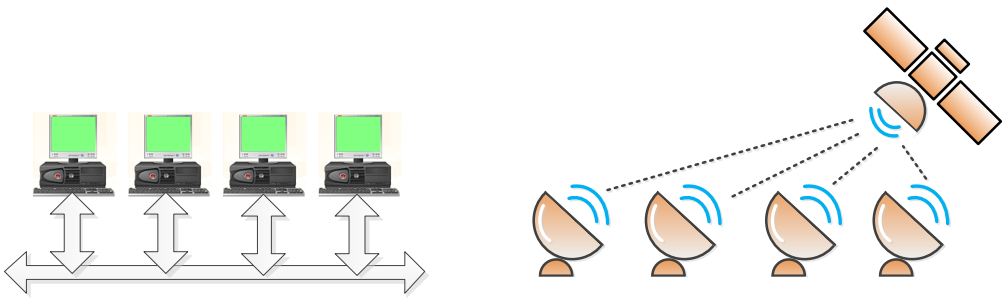
a) hvězda b) kruh c) strom



d) úplný polygon e) propojené kruhy e) obecná topologie

Obr. 4-2: Topologie sítí založených na dvoubodovém spojení

- **Kanály se všesměrovým vysíláním** (*broadcast*, případně *multipoint*). Multipoint představuje topologické uspořádání, na kterém může být vytvořeno více kanálů mezi různými dvěma místy. Broadcast pak je hromadný přenos z jednoho zdroje po společném kanálu do mnoha míst prostřednictvím buď rozvětveného vedení, nebo prostřednictvím všesměrového rádiového vysílání. Do této kategorie spadají mnohé lokální, metropolitní, rádiové či satelitní sítě, obecně často bezdrátové sítě. Systémy mají typicky jeden komunikační kanál, který je sdílen všemi uživateli sítě. Data vysílaná kterýmkoliv uživatelem jsou přijímána všemi ostatními a reaguje na ně obvykle pouze ten, jehož adresa je ve zprávě uvedena. Ostatní data ignorují. Systémy s všesměrovým vysíláním obecně rovněž umožňují současně adresovat data skupině či všem počítačům pomocí speciálních adres (např. tzv. skupinové adresování - *multicast*). Kanály s všesměrovým vysíláním vyžadují speciální rozhodovací mechanismus pro řešení konfliktů v případě, že na tomto společném kanále má zájem současně komunikovat více uzlů.



a) sběrnice (*multipoint*) b) rádiové spojení

Obr. 4-3: Topologie sítí založených na všesměrovém vysílání

4.3 Jiné členění sítí a technologií - dle velikosti

Sítě se dělí několika způsoby, z nichž nejběžnější je dělení dle velikosti, dosahu nebo rozlohy, na které se síť nachází. Toto členění zpravidla neposkytuje informaci z hlediska rychlostí těchto sítí. Ve všech kategoriích se můžeme potkat s velmi rychlými, ale i pomalejšími technologiemi. Některá řešení je velmi obtížně zařadit do jednoho konkrétního typu sítí a záleží zpravidla na konkrétním použití.

4.3.1 Personal Area Network (PAN)

Tzv. personální sítě představují sítě využívané pouze jednou osobou (nebo velmi nízkým počtem osob) a zpravidla se spíše nižšími přenosovými rychlostmi (jednotky Mbit/s). Setkáváme se zde se zařízeními, jako jsou chytré telefony, nositelná elektronika (*wearables*), tablety, ale i počítače nebo např. scannery či tiskárny. Typicky se jedná o bezdrátové (popř. i drátové) technologie s dosahem v řádu jednotek metrů či méně, jako jsou např. USB, Firewire, Bluetooth, NFC (*Near Field Communication*) nebo IrDA (*Infrared Data Association*). Příkladem je např. přenos dat mezi dvěma telefony nebo telefonem a počítačem.

4.3.2 Local Area Network (LAN)

Tzv. lokální sítě představují výkonný prostředek pro přenos informací v prostorově omezeném měřítku (typicky v rámci jedné budovy nebo maximálně v řádu kilometrů). Větších rozsahů se dosahuje propojením více LAN tzv. mosty nebo pomocí páteřních (*backbone*) sítí, např. MAN (*Metropolitan Area Network*), viz dále. Lokální sítě jsou dnes obvykle v provedení typu hvězda, případně v kombinaci s topologií typu strom, popř. sběrnicově a s rychlostmi 54 Mbit/s, 100 Mbit/s, 300 Mbit/s, 1 Gbit/s, ojediněle >10 Gbit/s. Dřívější LAN sítě využívají navíc také kruhové provedení. Běžné rychlosti dosahovaly 10 či 11 Mbit/s. LAN sítě (a i MAN sítě) jsou normalizovány rozsáhlou skupinou standardů IEEE 802. Počet uzlů je obvykle v řádu desítek či stovek, může být však i mnohem vyšší. Doba zpoždění přenosu mezi uzly je od 10 μs do 1 ms. Typicky se v tomto případě jedná o vnitřní instalace (domácnosti, firmy, celé budovy), tj. sítě ve vlastnictví a užívání jedné organizace nebo osoby, a technologie Fast Ethernet, Gigabit Ethernet, Wi-Fi (dříve Ethernet nebo Token Ring).

4.3.3 Metropolitan Area Network (MAN)

Tzv. metropolitní sítě jsou mezistupněm mezi lokálními sítěmi (LAN) a rozsáhlými sítěmi WAN, který zajišťuje především vysokorychlostní přenos dat mezi více lokálními sítěmi, případně mezi LAN a WAN. Rozsah těchto sítí je celoměstský a v současnosti se na principech sítí MAN budují i národní sítě. V těchto sítích se běžně pracuje s rychlostmi 1 Gbit/s a vyššími a poskytují tak prostředky pro přenos všech typů komunikace (telefonní služby, video, klasická data). Na úrovni MAN sítí se běžně potkáváme s optickými technologiemi anebo s rychlým Ethernetem, provozovaným přes optická vlákna. Dříve byly hojně využívány také technologie ATM (*Asynchronous Transfer Mode*) či FDDI (*Fiber Distributed Data Interface*). MAN síť je obvykle spravována jednou organizací, avšak její prostředky jsou využívány více subjekty. Zpoždění v těchto sítích je podobně jako u LAN sítí velmi nízké, přibližně na úrovni 100 μs až 10 ms.

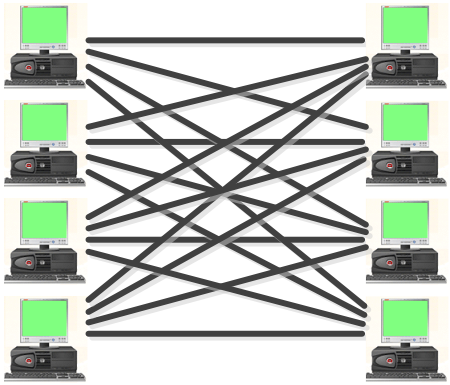
4.3.4 Wide Area Network (WAN)

Globální síť obvykle pokrývá rozlehlou oblast v řádu stovek i tisíců kilometrů. Typicky se jedná o sítě na úrovni jednotlivých států nebo kontinentů. Jejich hlavní úlohou je propojení jednotlivých geograficky rozprostřených LAN nebo MAN sítí. Jedna WAN síť může být vystavěna na různých technologiích a jednotlivé segmenty sítě mohou být vlastněny různými subjekty, přičemž provozovatel může mít některé části této sítě pouze v pronájmu (tzv. *leased lines*). Můžeme se setkat s přepínáním paketů, buněk, ale i okruhů, s technologiemi jako jsou POS (*Packet over SONET/SDH [Synchronous Optical Network/Synchronous Digital Hierarchy]*), MPLS (*Multiprotocol Label Switching*), dříve pak ATM (*Asynchronous Transfer Mode*), či FR (*Frame Relay*). V současnosti převládají optické technologie a z toho vyplývá, že i rychlosti těchto sítí jsou vysoké, avšak obecně lze říci, že WAN sítě jsou pomalejší, než MAN a LAN sítě. Topologie sítí WAN je obecná, požadavky na jednotlivé přenosové uzly jsou vysoké, jelikož do WAN sítě bývá připojeno větší množství subjektů. Zpoždění v těchto sítích je vzhledem k velkým vzdálenostem vyšší, řádově od jednotek ms až po stovky ms při využívání satelitních spojů nebo opravdu velkých vzdálenostech. Nejpoužívanější WAN sítí, resp. propojením většího množství WAN sítí, je tzv. Internetworking, zkráceně **Internet**. Jednotlivé WAN sítě můžeme označovat jako autonomní systémy (AS), které byly zmíněny již dříve.

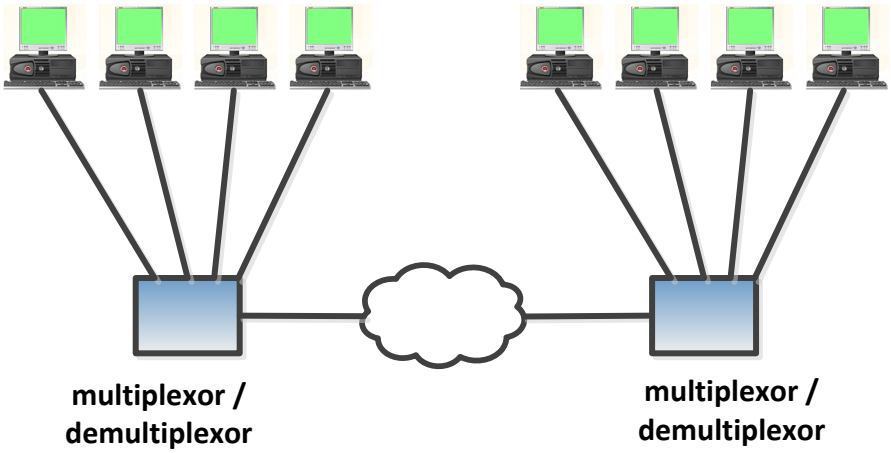
4.4 Vícenásobné využití přenosových cest

Jedním z úkolů komunikačních a telekomunikačních systémů je vhodně sdružovat různorodé signály před přenosem společnou přenosovou cestou a dále přizpůsobovat sdružené signály pro přenos touto cestou do k tomu účelu vhodného formátu.

Při vývoji přenosových systémů je patrná snaha o co nejefektivnější využití přenosového prostředí. Základem je, že nejlepšího ekonomického zhodnocení přenosových cest se dosáhne jejich vícenásobným využitím. Pro vícenásobné využití přenosového média se používají techniky multiplexování, kdy přes jedno médium je přenášeno více signálů (dat) od různých zdrojů k různým příjemcům, viz **Obr. 4-4**, z kterého je možné udělat srovnání počtu linek v případě, kdy není nebo naopak je využito multiplexování. Na obrázku je celkem 8 stanic (4 a 4), mezi kterými předpokládáme větší vzdálenost.



a)



b)

Obr. 4-4: Základní myšlenka vícenásobného využití přenosové trasy – rozdíl mezi (a) propojením bez multiplexování a (b) propojením s využitím multiplexu

Z hlediska využívání přenosových cest se postupně objevovaly následující principy vícenásobného přenosu:

- **Prostorové dělení** (prostorový multiplex), anglicky SDM (*Space-Division Multiplex*). Příkladem je více paralelních vedení, v rámci jednoho kabelu. Využíváno nejvíce v optice, kde je ekonomické, aby jeden optický kabel obsahoval více optických vláken. Tento způsob sám o sobě zpravidla není považován za pravé multiplexování.
- **Kmitočtové dělení** (frekvenční multiplex), anglicky FDM (*Frequency-Division Multiplex*), kdy se pro různé přenosy využívají různé kmitočty, resp. pásma kmitočtů v rámci dané trasy. Typickým příkladem je FM rádio, kde je možné na jednom místě na různém kmitočtu naladit různé stanice. Principiálně se jedná především o analogovou technologii, avšak neznamená to, že v jednotlivých pásmech nemohou být vysílány digitální signály. Z FDM vychází i velmi často využívané **OFDM** (*Orthogonal Frequency-Division Multiplex*). To je založeno na kódování digitálních dat na více nosných kmitočtů a je využíváno např. u **xDSL** (*Digital Subscriber Line*) technologií. Kmitočtové dělení je detailněji rozebráno v kap. 4.4.2.
- **Vlnové dělení** (vlnový multiplex), anglicky WDM (*Wavelength-Division Multiplex*). WDM představuje variantu kmitočtového dělení používanou v optice. Základní charakteristikou je, že se do jednoho optického vlákna multiplexuje více signálů, které jsou odlišeny svoji vlnovou délkou (tj. barvou).
- **Časové dělení** (časový multiplex), anglicky TDM (*Time-Division Multiplex*). TDM představuje zejména digitální technologie, kdy dochází k rychlému střídání účastníků v čase, čímž dochází ke sdílení přenosového pásma. Tomuto typu je více pozornosti věnováno v kap. 4.4.1
- **Kódové dělení** (kódový multiplex), anglicky CDM (*Code-Division Multiplex*). CDM, nebo též systémy s rozprostřeným spektrem, jsou nejsložitější a technicky patrně nejnáročnější způsob multiplexu. Jednotlivé přenosy jsou odlišeny speciální kódovou sekvencí. Bližší seznámení je nad rámec tohoto textu.

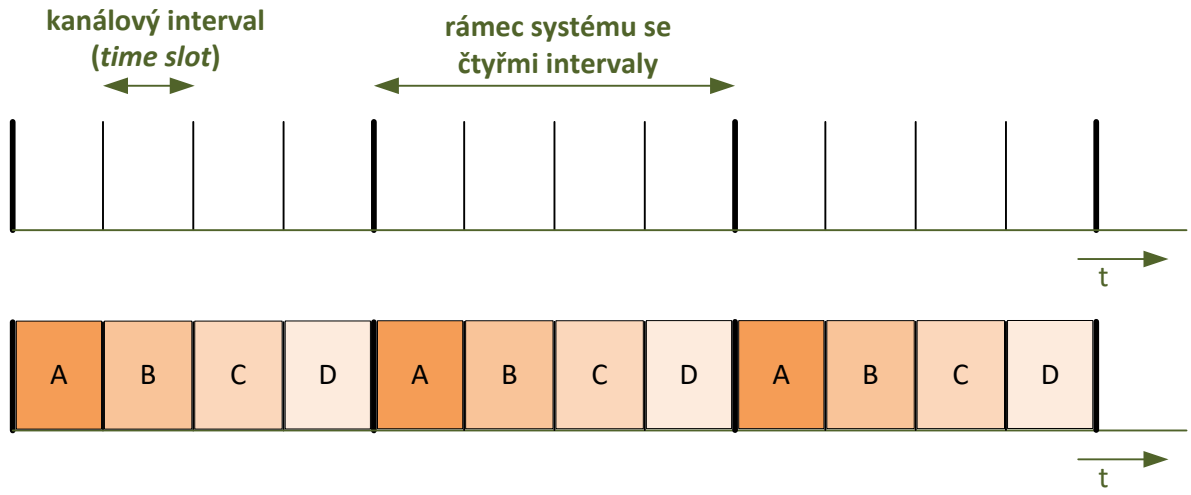
Tyto typy multiplexování jsou následně rozšířeny do tzv. přístupových metod, které umožňují v konkrétním nasazení několika vysílačům sdílet stejné přenosové médium jedním

z výše uvedených způsobů. FDM tak přechází do FDMA (*Frequency-Division Multiple Access*), TDM do TDMA (*Time-Division Multiple Access*) a CDM do CDMA (*Code-Division Multiple Access*). Tyto přístupové metody se běžně vzájemně kombinují, např. v systému GSM se v rádiové části využívá jak FDMA, tak TDMA.

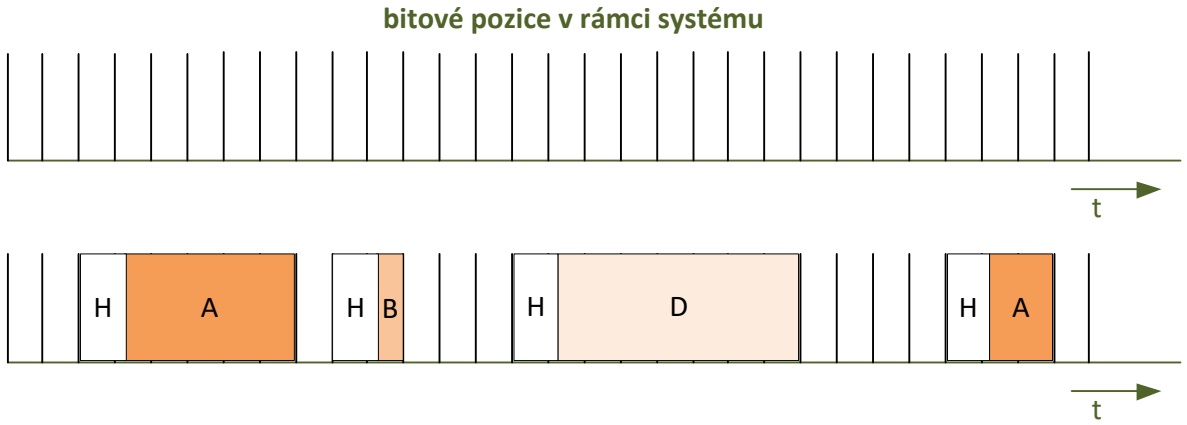
4.4.1 Časové dělení

Jak již bylo uvedeno, v případě časového dělení dochází ke střídání vysílajících stanic na sdíleném médiu. Představme si pro další popis situaci, že máme čtyři stanice, označené A až D, které mohou odesílat nějaká data systémem s časovým dělením. Existují tři základní přenosové režimy:

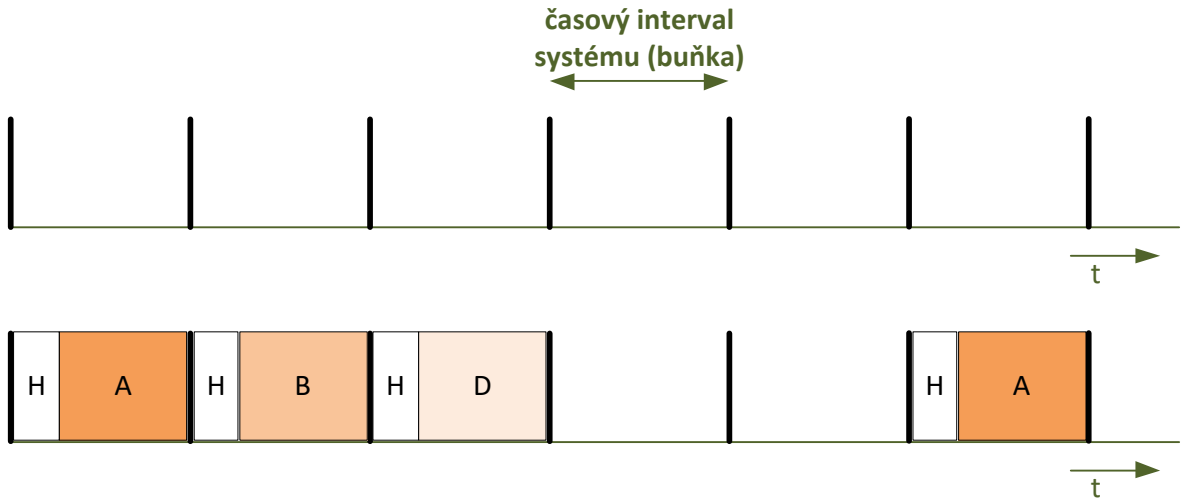
- **Synchronní přenosový mód** – u tohoto režimu platí, že stanice A až D se pravidelně střídají ve vysílání v předem daném pořadí, tj. každá má k dispozici čtvrtinu kapacity přenosového systému. Jednotlivým úsekům, kdy komunikuje jedna ze stanic, se říká kanálový interval (*time slot*) a těchto vždy stejně dlouhých intervalů může být až *n* v jednom rámci. Fakticky tento systém odpovídá komutaci okruhů tak, jak byl popsán na **Obr. 4-1**. Synchronní přenosový režim se využívá např. v přístupové části GSM síti, ale též v přenosovém systému PCM (*Pulse-Code Modulation*), kterému bude věnována pozornost dále. Výhodou tohoto přístupu je dostupnost konstantní rychlosti pro jednotlivé účastníky, avšak systém může z tohoto důvodu být značně neefektivní. V případě, kdy některá ze stanic A až D nebude aktuálně nic odesílat, totiž tato stanice blokuje ¼ kapacity systému, která by teoreticky mohla být využita pro zbývající stanice. Z pohledu vysílací strany je při tomto způsobu časového dělení třeba odesílaná data fragmentovat na přesně dané a stejně velké jednotky, které bude možné umístit do přiděleného kanálového intervalu. Graficky je situace znázorněna na **Obr. 4-5**.
- **Přenosový režim paketů** – tento režim odpovídá komutaci paketů z **Obr. 4-1** a připouští proměnnou délku zpráv, a tedy i nerovnoměrné rozdělení kapacity mezi vysílací stanice. Zprávy jsou odesílány tehdy, existuje-li k tomu požadavek. Každá zpráva musí obsahovat řídicí záhlaví, jelikož není předem dáno, komu patří. Tento režim je používán zcela běžně v současných datových sítích. Systém je flexibilnější než synchronní režim, avšak bez dalších mechanizmů nezajišťuje vysílacím stanicím žádnou přenosovou kapacitu, jelikož ta může být blokována jinými stanicemi. Graficky je situace znázorněna na **Obr. 4-6**.
- **Asynchronní přenosový režim** – v tomto systému existují buňky (elementární časové intervaly) s pevně danou délkou, do kterých lze vkládat rámce s přesně definovanou velikostí, avšak pouze v případě potřeby. Oproti synchronnímu přenosovému režimu je tedy v systému větší pružnost, jelikož jestliže např. stanice C nebude mít aktuálně nic k odeslání, kapacita může být využita ostatními stanicemi. Režie systému se však opět zvyšuje nutností přidávat řídicí záhlaví ke každé buňce. Konstantní délka jednotky může být limitující a může snižovat efektivitu systému. Přenosová kapacita pro jednotlivé uživatele může být různá a záleží především na nastavení systému (od nedefinované až po striktně vyhrazenou). Tento režim přenosu se typicky využívá v sítích ATM (*Asynchronous Transfer Mode*), viz kap. 6.4.4. Graficky je situace znázorněna na **Obr. 4-7**.



Obr. 4-5: Příklad synchronního přenosového módu – rovnoměrné rozdělení kapacity mezi čtyři stanice



Obr. 4-6: Příklad paketového přenosového módu – čtyři stanice (C nevysílá), různá velikost jednotek se záhlavím (značeno H), libovolná bitová pozice

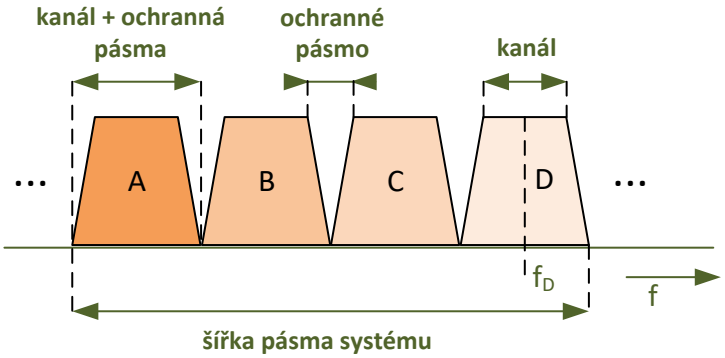


Obr. 4-7: Příklad asynchronního přenosového módu – čtyři stanice (C nevysílá), jednotky umístěné v libovolném intervalu, se záhlavím, a vždy se stejnou velikostí rámce

4.4.2 Kmitočtové dělení

Kmitočtové dělení spočívá v rozdělení kmitočtového spektra na jednotlivá pásma (rozsah kmitočtů, kanály). Jak již bylo uvedeno, jedná se o tradiční techniku, která je běžně využívána např. u FM rádia, kde na různém kmitočtu můžeme naladit jednotlivé stanice. Obdobně systém funguje i v jiných aplikacích, např. v systému GSM, kde kmitočtové dělení umožňuje, aby v jednom místě fungovalo více operátorů, kteří mají ke svému fungování přiděleny různé kanály. V dalších systémech může být kmitočtové dělení využito k odlišení směrů komunikace, kdy jedno pásmo je vyhrazeno pro komunikaci jedním směrem a druhé pro současně běžící komunikaci opačným směrem. Kmitočtové dělení je ilustrováno na **Obr. 4-8**. Z obrázku je patrné, že mezi jednotlivými kanály je nutné plánovat určité ochranné pásmo, aby nedocházelo k ovlivňování přenosů v navzájem sousedních kanálech.

Přenosový kanál je zpravidla definován středním kmitočtem [Hz] a šířkou pásma [Hz]. Na obrázku je jako ukázka naznačen střední kmitočet u kanálu D.



Obr. 4-8: Princip kmitočtového dělení do kanálů a ochranných pásem

4.5 Metody zajištění obousměrné komunikace

Existují celkem dva základní typy spojení či provozu z hlediska obousměrnosti. Jsou to simplexní spojení (*simplex*) a duplexní spojení (*duplex*). Výklad těchto pojmů se v literatuře různí[6], avšak my se budeme držet následujícího výkladu:

- **simplexní spojení** (*simplex*) – představuje řešení, kdy je možná obousměrná komunikace, avšak ne v jednom okamžiku zároveň. Tzn., že protistrany se musí nějakým způsobem dělit o přenosovou kapacitu, např. se střídat v čase. Klasickým případem jsou jednoduché vysílačky, kde nemohou oba účastníci hovořit zároveň a musí si předávat signál, že končí a dále může hovořit protistrana.
- **duplexní spojení** (*duplex*) – je systémem, kde technické prostředky umožňují současnou komunikaci oběma směry. Za plně duplexní lze považovat např. klasické

[6] V některých zdrojích se můžeme setkat s následující definicí simplexního spojení: Simplexní spojení je využito v případech, kdy obousměrnou komunikaci nepožadujeme, jedná se tedy vlastně o metodu jednosměrné komunikace. Systém se používá tam, kde postačuje přenos pouze jedním (předem daným) směrem a druhá strana nepotřebuje (a ani nemůže) žádným způsobem reagovat. Typickým příkladem je klasické rozhlasové a televizní vysílání, občas se tento přístup využívá také v signalizačních a senzorových systémech. Poté také tyto zdroje uvádějí zmínku o poloviční duplexním spojení (half-duplex), jehož popis odpovídá popisu simplexního spojení uvedeného v textu této kapitoly.

telefonní systémy, kde lze zároveň hovořit z obou stran (byť na komunikaci dvou osob to zpravidla nemá příznivý efekt). Tento způsob je využíván hojně v datových sítích, technických řešení pak existuje celá řada. V nejjednodušším případě existuje mezi oběma stanicemi dvojice kanálů, přičemž každý je vyhrazen pro komunikaci jedním směrem. V případě přenosu ve formě elektrického signálu (viz dále) existuje např. samostatná dvojice vodičů pro každý směr. U radiových přenosů se plný duplex běžně emuluje pomocí časového nebo frekvenčního dělení (viz kap. 4.4.1 a 4.4.2), což znamená, že přenos jedním směrem má vyhrazený jeden časový okamžik nebo kmitočet a přenos druhým směrem pak další časový okamžik nebo kmitočet.

5 Fyzická vrstva přenosových systémů

Jak již bylo dříve uvedeno, fyzická vrstva především přenáší tok bitů přenosovým médiem. Úkolem této vrstvy je tedy uzpůsobení dat získaných od spojové vrstvy do podoby bitového toku, což je často elektrický signál s měnícími se napěťovými úrovněmi. Zpravidla není možné bitový tok vyslat na médium přímo, ale musíme signál přizpůsobit nebo vhodně upravit. Fyzická vrstva tedy pracuje s **kódováním dat, modulacemi signálů a různými vlastnostmi přenosových médií**.

Na fyzické vrstvě se setkáváme se signály, nejčastěji elektrickými, které reprezentují bity zprávy. Důležitý je nejenom časový průběh signálu, ale i význam těchto signálů a jejich formát. **Nejdůležitějším úkolem fyzické vrstvy na straně vysílače je vytvořit na základě zadané bitové sekvence signál pro přenosové médium, na straně přijímače pak ze signálu na médiu rozpoznat odpovídající bitovou sekvenci.** Pro správné fungování musí být definována vzájemná návaznost řídicích a stavových signálů, to však zpravidla spadá do vyšší (spojové) vrstvy. **Do rámce fyzické vrstvy spadá dále také řešení problematiky přenosových tras a konektorů.**

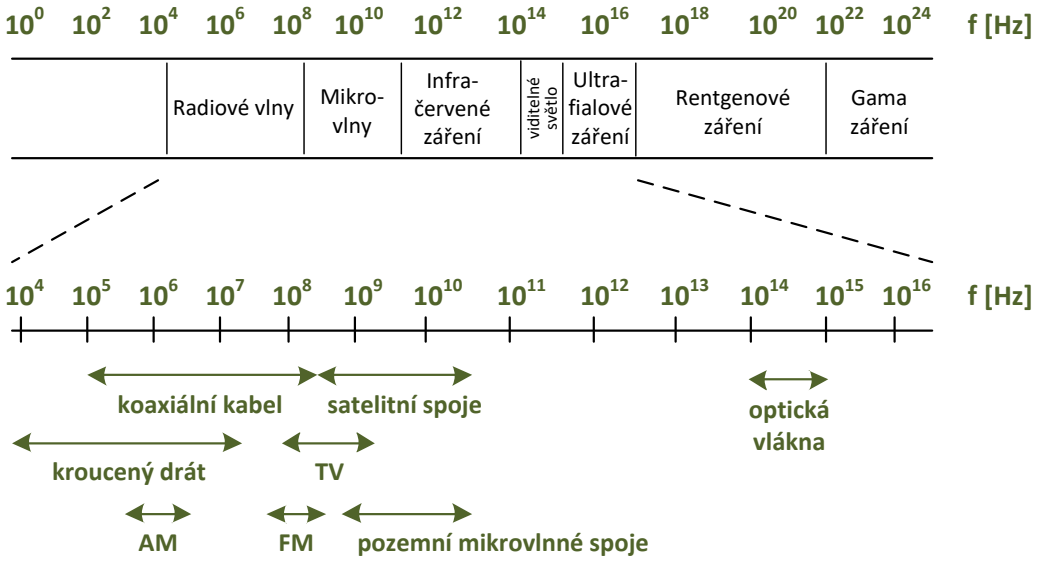
5.1 Úvod do problematiky přenosů na fyzické vrstvě

Přenosové médium představuje fyzické médium, kterým je přenášen signál od zdroje k cíli. Mezi nejběžnější přenosová média v datových sítích patří:

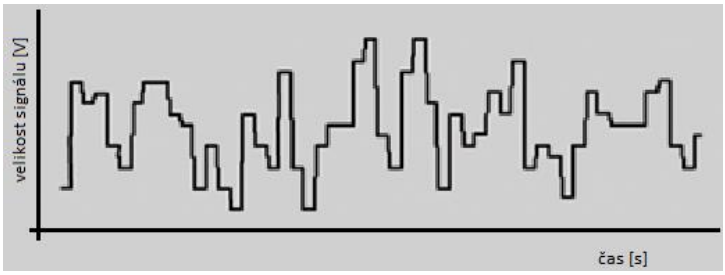
- **Elektrické vodiče** (obvykle měděné)
    - **Symetrický kabel**
    - **Koaxiální kabel**
- **Optická vlákna**
- **Volný prostor** (vzduch nebo vakuum)

Elektrickým i optickým přenosovým médiím je věnována větší pozornost v kap. 5.8. Ve všech výše uvedených případech je vlastní přenos realizován pomocí elektromagnetických vln, avšak v různých kmitočtových pásmech, jak naznačuje **Obr. 5-1**.

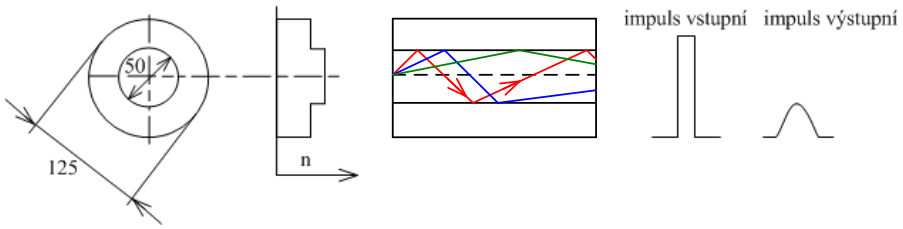
Podle typu média se zpravidla liší i to, jak vypadá přenášený signál. Ukázka signálu z elektrického vodiče je na **Obr. 5-2**, následuje ukázka z optického vlákna (**Obr. 5-3**) a v neposlední řadě jsou na **Obr. 5-4** ukázány i některé možnosti signálů vyslaných v bezdrátovém prostředí. Konkrétní techniky budou diskutovány v kap. 5.3.



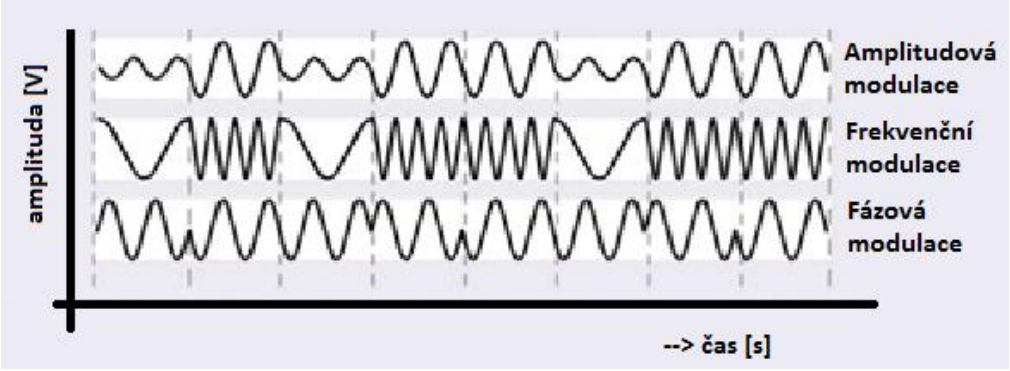
Obr. 5-1: Využitelnost spektra elektromagnetického záření (kmitočty uvedeny řádově)



Obr. 5-2: Ukázka elektrického signálu přenášeného na elektrickém vodiči



Obr. 5-3: Ukázka signálu na optickém vlákně, včetně několika parametrů (převzato)



Obr. 5-4: Ukázka možných signálů v bezdrátovém prostředí (jednotlivé modulace jsou vysvětleny v kap. 5.6)

5.2 Základní charakteristiky sledované u přenosových médií

Mezi základní parametry přenosového média (sledovaných především u metalických či optických vedení) patří šířka pásma, útlum, odolnost vůči elektromagnetickému rušení, impedance, přeslech mezi více vodiči a v neposlední řadě také cena.

- **Šířka pásma – závisí na fyzikálních vlastnostech daného přenosového média** a v konečném důsledku limituje množství dat, které je možné přenést daným médiem. Z pohledu přenosového média je šířka pásma vyjadřována buď v Hertzích, nebo nepřesně v bitech za sekundu, podle toho, zda jsou přenášeny analogové nebo digitální signály. Platí, že každý signál lze vyjádřit pomocí různých frekvenčních složek, jejichž přenos zpravidla omezuje právě dané přenosové médium. U medií, která mají "velkou" šířku pásma, je někdy šířka pásma záměrně rozdělena na části a to typicky u médií využívajících frekvenční dělení šířky pásma na jednotlivé kanály, viz kap. 4.4.2.
- **Útlum** – představuje zpravidla postupnou **ztrátu amplitudy** (velikosti) **signálu** na přenosovém médiu. Z toho jasně vyplývá, že útlum vždy závisí na délce média (resp. přenosové vzdálenosti). Základní jednotkou je decibel (dB), zejména u optických tras je však využívána i jednotka decibel na kilometr (dB/km). Zde však již hovoříme o měrném útlumu. Jsou definovány tři druhy útlumu, a to útlum napětí, proudu a výkonu. V případě útlumu výkonu je výpočet následující: A = 10 log (výstupní výkon / vstupní výkon). Z tohoto vzorce je snadno možno spočítat, že např. útlum 3 dB znamená snížení výkonu na 50 %.
- **Odolnost proti vnějšímu elektromagnetickému rušení** – odolnost vůči EMI (*ElectroMagnetic Interference*), které představuje zejména **energii z vnějších zdrojů** (v mnoha případech náhodnou) či energii ostatních signálů na stejném vedení, která může interferovat se signály na přenosovém médiu. Vlivem tohoto rušení může docházet ke zkreslení přenášeného signálu. Zdrojem rušení mohou být např. motory, lékařské přístroje, mobilní telefony, atd.
- **Impedance** – představuje velikost odporu (nejčastěji u vodiče) vůči střídavému elektrickému proudu. Impedanci dělíme na vstupní, výstupní a charakteristickou (vlnovou). Vlnová impedance má vliv na útlum média a je vyjadřována v jednotkách Ohm [Ω], přičemž platí, že pro velikost impedance je nejdůležitější indukční a kapacitní složka daného vedení.
- **Přeslech mezi vodiči** – představuje rušení signálem, který vzniká při vysílání po sousedním kanále či okruhu, či i z vodičů stejného vedení např. v kabelu. Existuje více druhů přeslechů, to je však nad rámec tohoto textu. Tento parametr je proto velmi **důležitý u paralelních vedení** a udává se v jednotkách dB.
- **Cena** – z ekonomického hlediska velice důležitý parametr. Vedení s kvalitnějšími parametry je obvykle dražší než vedení s méně kvalitními parametry. Velký vliv mají např. přídavné vrstvy stínění, které zvýší odolnost vůči rušení a sníží přeslechy, čímž se může např. zvýšit maximální možná přenosová rychlost.

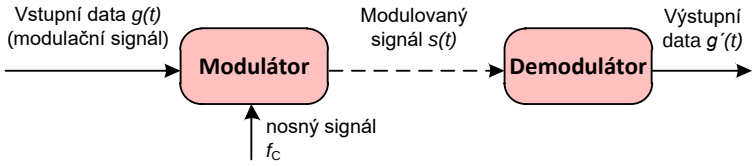
5.3 Úvod do přenosu digitálního signálu

Analogový signál představuje spojitý signál. Je to signál vyjadřující zprávu pomocí neomezeného počtu hodnot určité fyzikální veličiny (např. amplitudy, frekvence). Analogové signály se běžně vyskytují v přírodě. Naproti tomu digitální signály jsou výtvorem člověka. Digitální (číslicový) signál je ve své podstatě signál nespojitý v čase i amplitudě vyjadřující

zprávu pomocí omezeného počtu hodnot určité fyzikální veličiny (např. jen dvou hodnot, tj. jedniček a nul). Oba typy signálů se běžně používají např. v komunikačních technologiích, přičemž v současnosti dominují spíše digitální technologie.

Digitální signál je možné přenášet v:

- **základním pásmu** – tzv. ***kódování*** – úprava signálu pomocí tzv. linkových kódů. Principem přenosu v základním pásmu je, že se na médiu přenášejí pravoúhlé impulzy v původní frekvenční poloze. Rozdíly mezi jednotlivými kódy jsou v tom, jak je reprezentován určitý signálový prvek, či určitá sekvence signálových prvků. Více je tomuto tématu věnováno v kap. 5.5.
- **přeneseném (přeloženém) pásmu** – využití ***modulací*** (***klíčování***). Technika spočívá v přenesení signálu na určitý nosný kmitočet (f_C) tak, aby bylo možné ho vyslat na určité médium nebo přímo v konkrétním pásmu (kanálu). Následně dochází buď k řízení amplitudy, kmitočtu nebo fáze signálu v závislosti na okamžité hodnotě modulačního signálu (nebo kombinaci řízení více parametrů). Toto ovlivňování umožňuje reprezentovat původní posloupnost signálových prvků. Pro účely modulace se využívají dva základní bloky, a to *modulátor* a *demodulátor*, přičemž jak již vyplývá z názvů, první blok provádí modulaci (tj. úpravu signálu před vysláním na médium, nebo také přizpůsobení signálu přenosovému médiu a přeložení do přenosového pásma) a druhý demodulaci (úpravu přijatého signálu po jeho přijetí, navrácení signálu do základního pásma), princip je graficky znázorněn na **Obr. 5-5**. Této problematice je více věnováno v kap. 5.6.



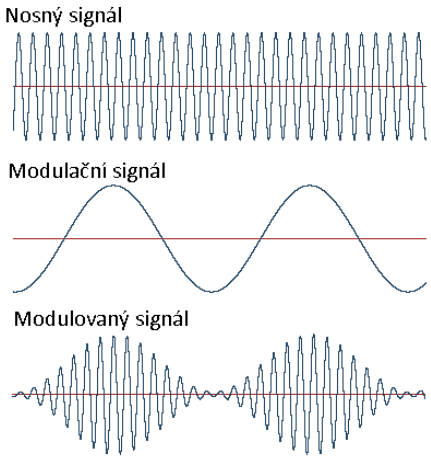
Obr. 5-5: Grafické znázornění modulace a demodulace a souvisejících pojmů

5.4 Analogové modulace

Základy modulací, tj. přenosu signálu v přeneseném pásmu, byly položeny u analogových signálů, a jelikož základní principy platí stejně i pro pokročilejší digitální modulace, budeme jim nyní věnovat pozornost. U analogových modulací dochází ke skládání vstupního analogového signálu se signálem nosné frekvence a to spojitě v čase. Výsledek je modulovaný signál, který je stále analogovým signálem, avšak typicky přeneseným na jiný kmitočet a mající určité vlastnosti. Šířka pásma výsledného signálu je poté centrována kolem frekvence nosného kmitočtu.

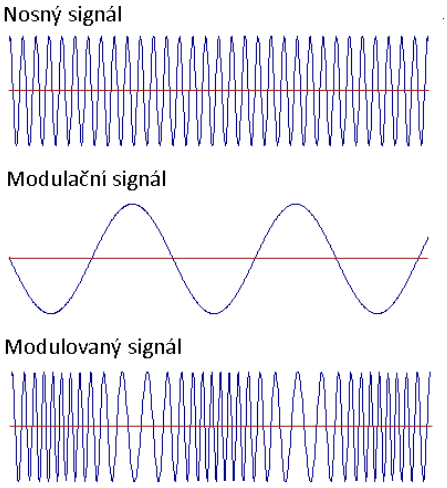
Existují tři základní typy analogových modulací:

- **Amplitudová modulace** (AM) – jednoduchý typ spojité analogové modulace. V závislosti na změně modulačního signálu se mění amplituda nosného signálu a ostatní parametry zůstávají nezměněny. Situace je znázorněna na **Obr. 5-6**. Z obrázku je zřejmé, že nosný signál má řádově vyšší kmitočet, než modulační signál. Technika AM je stále využívána např. při radiovém vysílání (např. rozhlas na dlouhých vlnách).



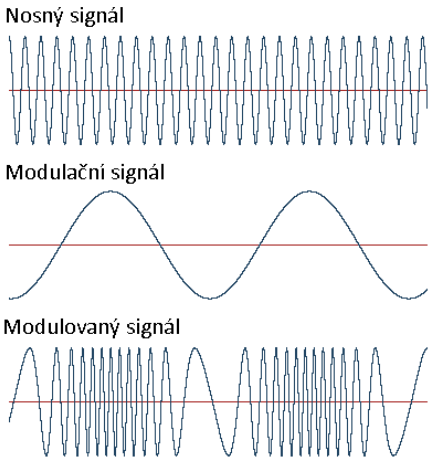
Obr. 5-6: Ukázka analogové amplitudové modulace (AM)

- **Kmitočtová modulace** (FM) – u frekvenční modulace je princip obdobný, avšak nedochází ke změně amplitudy, ale kmitočtu nosné vlny v závislosti na změně amplitudy modulačního signálu. **Obr. 5-7** demonstruje základní princip fungování. FM je běžně využíváno také při radiovém vysílání (např. FM rozhlas, velmi krátké vlny).



Obr. 5-7: Ukázka analogové frekvenční modulace (FM)

- **Fázová modulace** (PM) – u tohoto typu modulace dochází na základě modulačního signálu ke změně okamžité fáze nosného signálu. Pro složitost demodulace je využívána méně než předcházející dvě techniky. Formálně je tato modulace velmi příbuzná s kmitočtovou modulací. Ukázka PM je na **Obr. 5-8**.



Obr. 5-8: Ukázka analogové fázové modulace (PM)

5.5 Přenos digitálního signálu v základním pásmu

Přenos digitálního signálu v základním pásmu (tzv. kódování) je charakteristické umístěním v původní frekvenční poloze, tj. v pásmu začínajícím u frekvencí blízkých nule nebo obsahujících i stejnosměrnou složku.

Z pohledu stejnosměrné složky signálu můžeme rozlišovat dva typy přenosů:

- **Přenos se stejnosměrnou složkou** – příslušný kanál musí umět přenést i tuto stejnosměrnou složku, což vyžaduje galvanické spojení koncových zařízení.
- **Přenos bez stejnosměrné složky** – stejnosměrná složka je potlačena vhodným kódováním a příslušný kanál ji nemusí přenášet. Tento způsob je v reálných přenosových systémech využit častěji.

Jednotlivé typy linkových kódů můžeme **klasifikovat podle tří základních hledisek**:

- **Podle počtu úrovní definujeme signály**
  - **dvoustavové** – unipolární; existují dvě úrovně, jedna z nich je nulová a druhá nenulová.
  - **třístavové** – bipolární (pseudotrojkové), kromě nulové úrovně existují i další dvě stejné úrovně (v absolutní hodnotě), typicky s navzájem opačnou polaritou.
  - **vícestavové** – v kódu existuje větší množství úrovní, zpravidla rozložených do obou polarit.
- **Podle použité polarity signálových prvků**
  - **unipolární** (jedné polarity) – signálové prvky nabývají pouze jednu polaritu (ať už kladnou nebo zápornou).
  - **bipolární** (dvojí polarity) – signálové prvky mohou nabývat obě polarity.
- Podle toho, **zda se průběh vrací průběžně k nulové úrovni**, nebo přechází přímo k druhému stavu:
  - **signály s návratem k nulové úrovni** – *Return to Zero* (RZ).

- **signály bez návratu k nulové úrovni** – *Non-Return to Zero* (NRZ).

5.5.1 Význam linkových kódů

Jednotlivé linkové kódy se liší v přístupu k řešení následujících úkolů:

- **Stejnosměrná složka** – resp. její potlačení nebo alespoň snížení, které ne všechny kódy umí.
- **Synchronizace v přijímači** – v přijímači je nutná obnova synchronizace (časování), kterou některé linkové kódy usnadňují.
- **Detekce chyb** - některé kódy umožňují rozpoznat určitou míru chyb.
- **Šířka pásma** - vícestavové linkové kódy dokáží snížit nároky na potřebnou šířku pásma.
- **Odolnost vůči šumu** – některé kódy vykazují nižší chybovost při porovnatelném odstupu signálu od šumu.

5.5.2 Příklady jednoduchých linkových kódů

Existuje velké množství tzv. linkových kódů, mimo dále uvedené jsou to také např. HDB3, 2B1Q, 4B5B. Rozdíly mezi jednotlivými kódy jsou patrné z jejich popisu a také z názorného **Obr. 5-9**. Každý z kódů je vhodný pro jiné použití, detailnější popis je však nad rámec tohoto textu.

Unipolární a bipolární kód NRZ-L (*Non-Return to Zero Level*)

Jednoduchý kód bez návratu k nule. Logická nula je reprezentována jednou úrovní a logická jednička druhou, např. „1“ má vysokou napěťovou úroveň (H), zatímco „0“ pak nízkou úroveň (L). V některých aplikacích to může být i opačně. V případě unipolárního kódu bude jedna úroveň nenulová a druhá nulová, zatímco u bipolárního kódu budou obě úrovně nenulové. Zřejmým problémem tohoto kódu (zejména v jeho unipolární variantě) je stejnosměrná složka, která se zde vyskytuje a může způsobovat problémy na přenosové trase. Další nevýhodou kódu je situace, kdy se kóduje dlouhá sekvence stejné úrovně, v případě, že se v signálu vyskytuje delší řada „0“ nebo „1“ za sebou. To znesnadňuje synchronizaci mezi vysílací a přijímací stranou. Bipolární varianta NRZ-L je využita na rozhraní RS-232.

Unipolární a bipolární kód NRZ-I (*Non-Return to Zero Inverted*)

Tento kód se od předcházejícího liší v logice kódování. Platí, že se u něj kóduje změna (diferenční kód), tj. že např. hodnota „1“ znamená změnu úrovně signálového prvku, zatímco hodnota „0“ nezpůsobuje žádnou změnu. Opět zde tedy pracujeme s dvěma úrovněmi, avšak hodnota aktuálního signálového prvku je závislá i na tom předchozím. Bipolární varianta je v mírně vylepšené verzi využívána na sběrnici USB.

Kód AMI (*Alternate Mark Inversion*)

Bipolární kód, u kterého má hodnota „1“ za následek střídavě kladnou a zápornou změnu úrovně, zatímco hodnota „0“ nevytváří žádný signál. Tento kód nemá problém se stejnosměrnou složkou (ta je nulová), avšak problémem zůstává delší sekvence hodnoty „0“, kdy zůstává po delší čas stejná napěťová úroveň. Proto existují i další varianty tohoto kódu

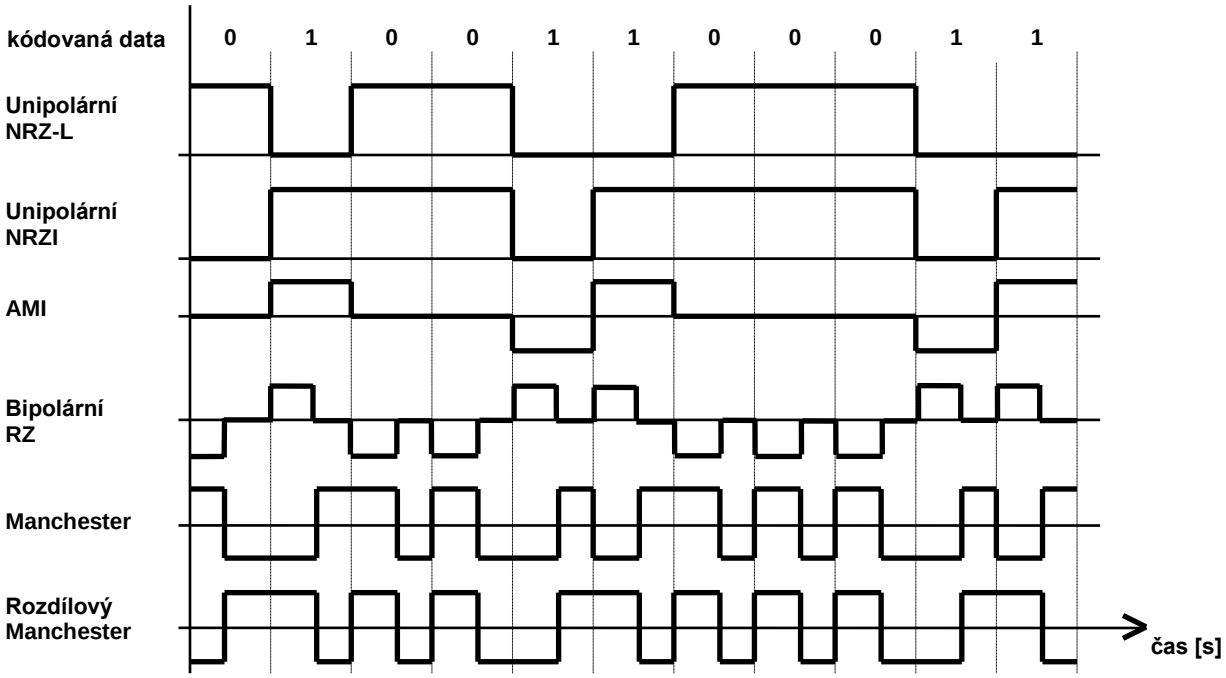
a i další příbuzné kódy, které tento problém řeší, avšak to je již nad rámec tohoto textu. Kód AMI je využíván ve starších telefonních systémech.

Kód RZ (*Return to Zero*)

Tento typ kódu může být principiálně unipolární i bipolární, avšak větší smysl má zcela určitě u bipolárního typu kódování, kterému je věnován další popis. Podobně jako u bipolárního NRZ-L je hodnota „1“ kódována jednou úrovní a hodnota „0“ pak druhou úrovní, avšak platí, že v polovině intervalu dojde k navrácení na nulovou úroveň napětí. Tento typ kódu mírně snižuje stejnosměrnou složku výsledného signálu a řeší problém se synchronizací, jelikož i v delší sekvenci signálů stejné úrovně se hodnota napětí pravidelně mění. Unipolární RZ v mírně modifikované variantě je využíván u infračervených optických přenosů na malou vzdálenost.

Kód Manchester

U tohoto kódu jsou jednotlivé bity reprezentovány přechodem úrovně uprostřed intervalu. Hodnota „0“ je typicky reprezentována přechodem z vysoké úrovně (H) na nízkou (L) a hodnota „1“ pak opačným přechodem, tj. změnou z „L“ na „H“. Opět platí, že logika může být i opačná. Kód je bipolární a proto vzhledem ke svému charakteru nemá žádnou stejnosměrnou složku. Výhodou je také samo-časovací vlastnost, jelikož kód vždy obsahuje pravidelně se opakující hrany. Tento kód je využíván u Ethernetu (10 Mbit/s standard).



Obr. 5-9: Ukázka vybraných linkových kódů

Rozdílový kód Manchester

Tento diferenční kód vychází ze standardního kódu Manchester a i u tohoto kódu dochází pravidelně ke změně úrovně uprostřed bitového intervalu. Avšak logická „0“

a logická „1“ jsou v tomto případě reprezentovány změnou na začátku intervalu. Jestliže kódujeme hodnotu „0“, na začátku intervalu dojde ke změně úrovně a jestliže hodnotu „1“, ke změně nedojde. Je zřejmé, že zda k této změně dojde nebo nedojde, určuje to, zda v půli intervalu bude sestupná nebo náběžná hrana. Tento kód je využíván u technologie lokálních sítí typu Token ring.

5.6 Přenos digitálního signálu v přeneseném pásmu

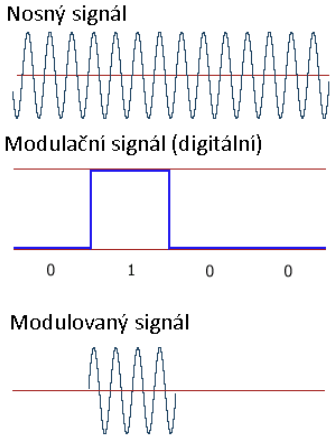
Přenos digitálního signálu v přeneseném pásmu probíhá za pomocí digitálních modulačních technik, tzv. ***klíčování*** (ne zcela správně nazývány také jako modulace). Principiálně jsou digitální modulace podobné těm analogovým. Hlavním rozdílem je, že modulační signál je diskrétní. Digitální klíčovací techniky jsou hojně využívány v přenosových systémech a zejména pak v bezdrátových přenosech (mobilní telefonní sítě, bezdrátové sítě Wi-Fi), ale např. i u ADSL (*Asymmetric Digital Subscriber Line*).

Vzhledem k tomu, že modulační signál je diskrétní, dochází u nosného signálu (který bývá harmonický) ke skokovým změnám. U tohoto signálu můžeme měnit jeho amplitudu, frekvenci, fázi (ukázky těchto modulací jsou zjednodušeně znázorněny také jako součást **Obr. 5-4**) anebo kombinaci některých z uvedených parametrů. Digitálních klíčovacích metod existuje obrovské množství, základní tři techniky tedy jsou:

- Amplitudové klíčování ASK (*Amplitude Shift Keying*)
- Frekvenční klíčování FSK (*Frequency Shift Keying*)
- Fázové klíčování PSK (*Phase Shift Keying*)

5.6.1 Amplitudové klíčování (ASK)

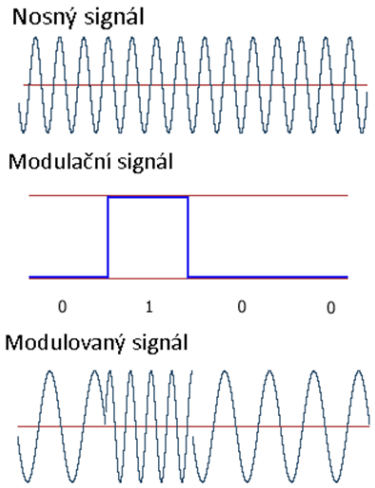
Tato velice jednoduchá technika spočívá v tom, že modulační signál střídavě spíná a vypíná nosný signál, podle toho, zda je právě modulována hodnota „1“ nebo hodnota „0“. Princip je ilustrován na **Obr. 5-10**. V této nejjednodušší podobě se ASK příliš nepoužívá, protože nemá moc výhodné vlastnosti, s výjimkou dobré citlivosti na náhlé změny signálu. Změna amplitudy je využívána zpravidla v kombinaci se změnou fáze a u pokročilých technik pak existuje i více definovaných úrovní než dvě (jak u amplitudy, tak i fáze). V případech, kdy jsou využity více než dvě úrovně, je možné do jedné napěťové úrovně „skrýt“ více než jeden bit.



Obr. 5-10: Základní princip amplitudového klíčování (ASK)

5.6.2 Frekvenční klíčování (FSK)

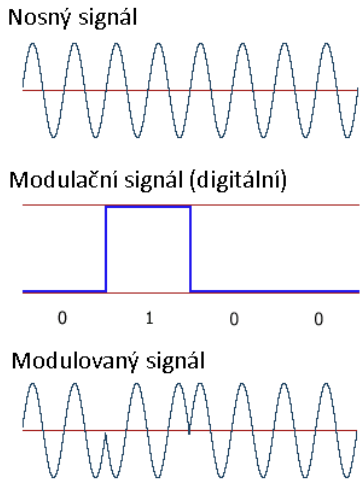
U frekvenčního klíčování se v závislosti na modulačním signálu skokově mění frekvence nosného signálu. V nejjednodušším případě potřebujeme dvě frekvence, mezi kterými se přepíná podle toho, zda je přenášena hodnota „0“ nebo „1“. Obě frekvence bývají umístěny blízko nosného kmitočtu. U frekvenčního klíčování bývá odolnost vůči chybám vyšší než u ASK a používá se hojně u radiových přenosů. Princip FSK je patrný také z **Obr. 5-11**. Jestliže se u FSK využijí více než dva kmitočty, je možné přenášet více bitů naráz, čehož je v praxi často využíváno.



Obr. 5-11: Základní princip frekvenčního klíčování (FSK)

5.6.3 Fázové klíčování (PSK)

Fázové klíčování spočívá v ovlivňování počáteční fáze v daném intervalu. V základním režimu platí, že hodnota „0“ je reprezentována jednou hodnotou počáteční fáze a hodnota „1“ fází opačnou (tj. o 180 stupňů posunutou). Základní princip je ukázán na **Obr. 5-12**. U pokročilejších technik jsou bity vyjádřeny větším množstvím fází anebo také určitou změnou fáze (diferenční klíčování). Techniky fázového klíčování jsou používány nejčastěji v kombinaci s amplitudovým klíčováním.



Obr. 5-12: Základní princip fázového klíčování (PSK)

5.6.4 Vícestavové klíčování

Předcházející popis amplitudového, frekvenčního a fázového klíčování byl zaměřen především na dvoustavové klíčování. To znamená, že každému jednomu bitu je přiřazen jeden stav nosného signálu (existují dvě amplitudy, dvě frekvence nebo dvě fáze). Tento způsob je z hlediska využití šířky pásma málo efektivní, proto se v modernějších systémech přistoupilo k využívání vícestavových digitálních modulačních technik. Jestliže chceme jedním stavem signálového prvku vyjádřit *n* bitů, potřebujeme 2^n stavů. Např. u 4-stavového klíčování přenášíme jedním signálovým prvkem 2 bity (dibit), u 256-stavového klíčování pak 8 bitů. Se zvyšujícím se počtem stavů výrazně roste efektivita přenosu, avšak zároveň se zvyšuje složitost přenosového systému jak na straně vysílače, tak na straně příjemce a také se snižuje odolnost přenosu vůči chybám, rušení apod. V názvech vícestavových modulací se objevuje počet stavu, se kterými technika pracuje, např. 8-PSK (nejčastěji jsou vícestavové modulace využívány právě u fázových modulací).

5.6.5 Kombinované fázové a amplitudové klíčování

Jestliže chceme dosáhnout co největšího počtu stavů (a tedy i přenášeného počtu bitů), je výhodné kombinovat více druhů klíčování. Nejčastěji se v tomto ohledu využívají kombinace fázového a amplitudového klíčování, kdy je modulačním signálem ovlivňována jak fáze, tak amplituda nosného signálu. Tento typ modulace se nazývá Kvadraturní amplitudová modulace QAM (*Quadrature Amplitude Modulation*). Běžně se využívají 8QAM, 16QAM, 32QAM, 64QAM, 128QAM a 256QAM, z čehož je zřejmý i celkový počet stavů nosného signálu.

5.7 Digitalizace řečového signálu

5.7.1 Základní postup při digitalizaci řeči

Převod mluveného slova do digitální podoby je základem moderních komunikačních a především telekomunikačních technik. Tato operace se provádí ve třech krocích, které na sebe přímo navazují:

- **vzorkování** – úkolem této etapy je ze spojitého signálu periodicky snímat aktuální hodnoty, a to vhodnou rychlostí, resp. s určitou frekvencí, která se nazývá vzorkovací kmitočet. Ze signálu se spojitým časem získáme signál, kde se vyskytují vzorky pouze v diskrétních hodnotách času.
- **kvantování** – signál se touto operací stává diskrétním, z neomezeného množství hladin se při kvantování vytvoří pouze např. 16 či 256 možných úrovní (hodnot). Zde dochází (zjednodušeně řečeno) k zaokrouhlení navzorkované hodnoty na nejbližší existující kvantovací úroveň, jelikož počet možných hodnot, kterých může digitální signál nabývat je omezen počtem bitů, které pro reprezentaci daného vzorku využíváme. Kvantování má za následek nevratné zkreslení původního signálu, které však při dostatečném počtu kvantovacích úrovní nemusí být pro člověka znatelné.
- **kódování** – v poslední etapě je stanovené hladině vzorku přiřazena určitá posloupnost, která danou hodnotu reprezentuje v použitém kódu. Zde může být principiálně použit některý typ kódu, o kterých bylo pojednáno v kap. 5.5, avšak existuje celá řada speciálních kódovacích technik, které slouží primárně k digitalizaci řeči. Liší se

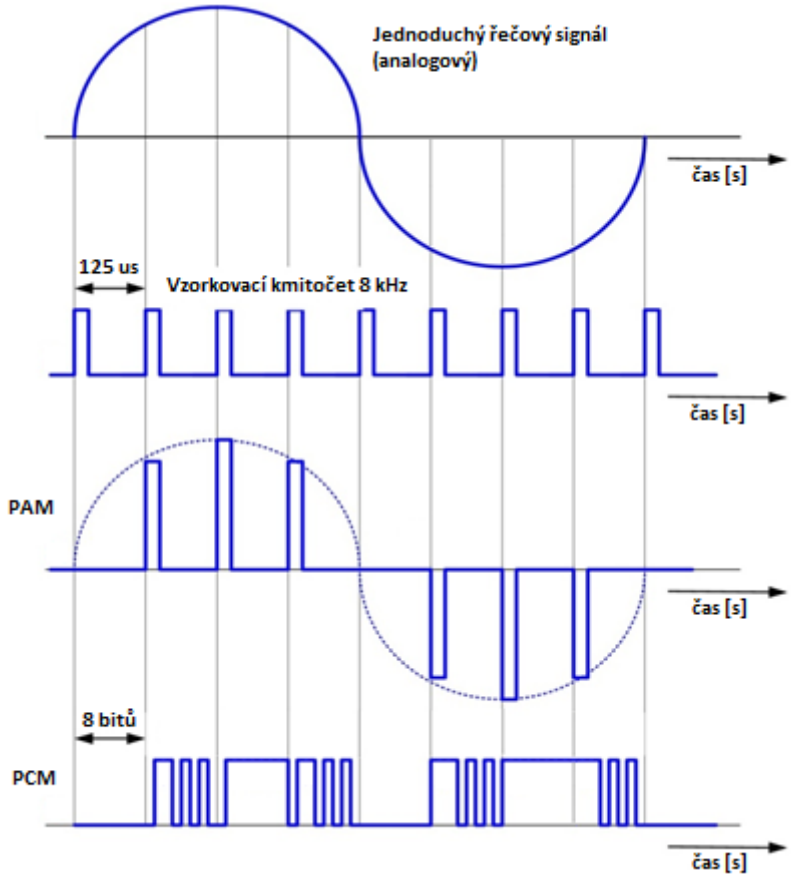
především v požadavcích na šířku pásma a dosahované kvalitě (věrnosti) reprezentace původního (analogového) signálu.

5.7.2 Příklad digitalizace řeči – systém PCM

Řečový signál je kmitočtově umístěn zejména v akustickém pásmu, od stovek Hz až po jednotky kHz. Pro účely přenosu řeči v telekomunikacích se běžně uvažuje omezené pásmo v rozsahu 300 až 3400 Hz, které plně postačuje k velmi věrné reprezentaci řeči. Podle Shannon – Kotelnikova (Nyquistova) teorému platí, že vzorkovací kmitočet musí být více než dvakrát větší než maximální frekvence vzorkovaného signálu, což lze zapsat jako:

$$f_{vz} > 2\, f_{max}\,.$$

S určitou rezervou je proto pro vzorkování řeči využíván kmitočet 8 kHz, což zajistí získání dostatečného množství vzorků. Vzorkování je často prováděno pomocí Pulzně amplitudové modulace[7] (PAM = Pulse Amplitude Modulation). V klasických telekomunikacích se poté pro kvantování a kódování používá tzv. pulzní kódová modulace (PCM = *Pulse Code Modulation*), případně její vylepšené obdoby. U základní PCM je využito celkem 256 kvantovacích hladin, což odpovídá 8 bitům na jeden vzorek. Těchto 8 bitů je pak následně kódováno jednoduchou kódovou sekvencí hodnot „1“ a „0“ (standardní binární vyjádření). Celá situace je ilustrována na **Obr. 5-13**.



Obr. 5-13: Ukázka principu digitalizace řečového signálu v systému PCM

[7] O tomto typu modulace není v tomto textu pojednáno. Popis či možné varianty jsou nad rámec tohoto textu.

Pro přenosovou rychlost digitalizovaného (pomocí PCM) řečového signálu tedy platí:

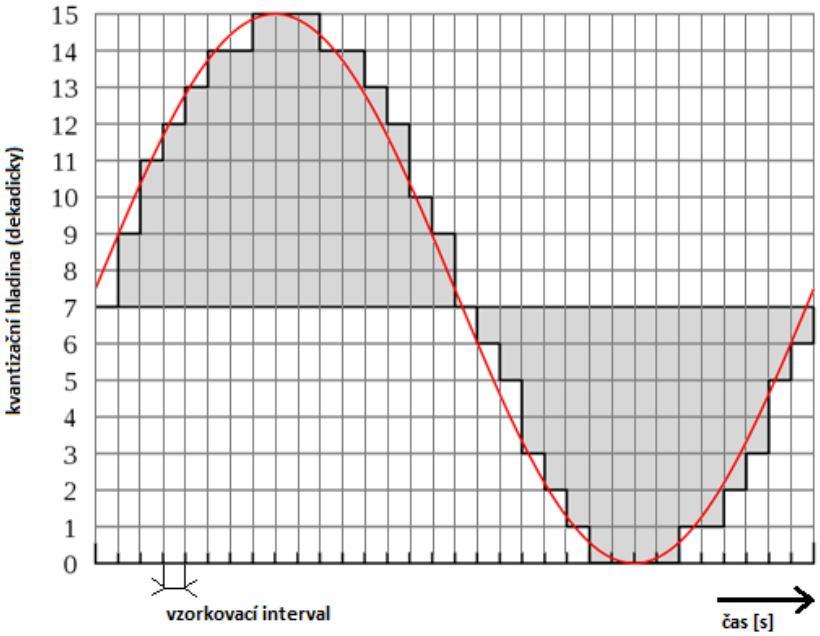
8000 [vzorků / sekunda] * 8 [bitů / vzorek] = 64 000 [bitů / sekunda] = 64 kbit/s.

Tato rychlost představuje základní jednotku v klasických telekomunikacích, např. v případě mobilních sítí se však můžeme setkat s nižšími požadavky na přenosovou rychlost, což však bývá zapříčiněno použitím pokročilejších technik (kodeků).

5.7.3 Kvantizační šum

Již v kap. 5.7.1 bylo zmíněno, že při kvantování vzorkovaného signálu dochází k nevratnému zkreslení původního signálu. Toto zkreslení je reprezentováno tzv. kvantizačním šumem. Jestliže nekonečný počet hladin omezíme na konečný, jedná se o proces, při kterém ztrácíme část informace. Ilustrační obrázek je možné nalézt na **Obr. 5-14**, kde je pro jednoduchost zachycena situace pro 16úrovňové kvantování. Červená čára je původní analogový signál, černá je pak signál po kvantování (a následné rekonstrukci bez dalších úprav). Kvantizační šum je dán rozdílem mezi hodnotou u původního signálu a tou novou. Z obrázku je patrné, že v krajním případě může být kvantizační šum u vzorku roven ½ rozdílu mezi kvantizačními hladinami. Je zřejmé, že s rostoucím počtem existujících hladin tento šum klesá.

Pozn.: ve skutečných systémech jsou kvantizační hladiny rozděleny nerovnoměrně, aby lépe reprezentovaly původní řečový signál vzhledem ke vlastnostem lidského ucha. V tomto ohledu se setkáváme s pojmy A-law a μ-law, které jsou však již nad rámec tohoto textu.



Obr. 5-14: Vysvětlení vzniku kvantizačního šumu

5.7.4 Digitální přenosové systémy na bázi PCM

V telekomunikačních sítích se setkáváme s potřebou přenášet větší množství řečových signálů (hovorů) na bázi PCM, tedy kanálů s rychlostí 64 kbit/s. V tomto ohledu existuje více standardů, přičemž z našeho pohledu nejdůležitější je přenosový systém E, který se používá v Evropě a přenosový systém T, který je užíván v Severní Americe. Základní jednotkou

systému E je multiplex E1 s rychlostí 2,048 Mbit/s, který sdružuje celkem 30 kanálů (30 x 64 kbit/s = 1920 kbit/s). Zbytek jeho přenosové kapacity je využit pro synchronizační a signalizační účely. Naproti tomu u systému T je základní jednotkou telekomunikační kanál T1 s rychlostí 1,544 Mbit/s, který sdružuje pouze 24 kanálů (24 x 64 kbit/s = 1536 kbit/s) a opět platí, že zbytek kapacity je využit pro synchronizační a signalizační účely. Existují však i multiplexy sdružující větší množství kanálů, jak ukazuje **Tab. 1**.

Pozn.: v této tabulce nejsou uvedeny další existující standardy používané např. v Japonsku.

Systém násobného skládání nižších multiplexů se nazývá **Plesiochronní digitální hierarchie (PDH)**. Detailní popis skládání multiplexů je nad rámec tohoto textu, avšak platí, že systém je z dnešního pohledu málo pružný a příliš komplikovaný. Z tohoto důvodu je v současnosti více využívána tzv. **Synchronní digitální hierarchie (SDH)**, která má oproti PHD četné výhody. U SDH se setkáváme s multiplexy STM-0 (51 Mbit/s), STM-1 (155 Mbit/s), STM-4 (622 Mbit/s), STM-16 (2,4 Gbit/s), ale i novějšími STM-64 (10 Gbit/s) a STM-256 (40 Gbit/s). Opět platí, že více detailů k této technologii je nad rámec tohoto textu.

Z hlediska zpětné kompatibility je důležité, že do multiplexů STM lze převést i multiplexy staršího PDH. Z hlediska kompatibility mezi jednotlivými systémy (např. Evropa a Amerika) je důležité, že existují postupy, jak převádět jednotlivé multiplexy v různých úrovních mezi jednotlivými systémy.

Tab. 1: Srovnání Evropských a Severoamerických přenosových systémů

Oblast	Typ multiplexu	Bitový tok (Mbit/s)	Počet datových/telekomunikačních kanálů
Evropa	E1	2	30
	E2	8	120 (4x E1)
	E3	34	480 (4x E2)
	E4	139	1920 (4x E3)
	E5	565	7680 (4x E4)
Severní Amerika	T1	1,5	24
	T2	6	96 (4x T1)
	T3	45	672 (7x T2)
	T4	274	4032 (6x T3)
	T5	400	5760 (60x T2)

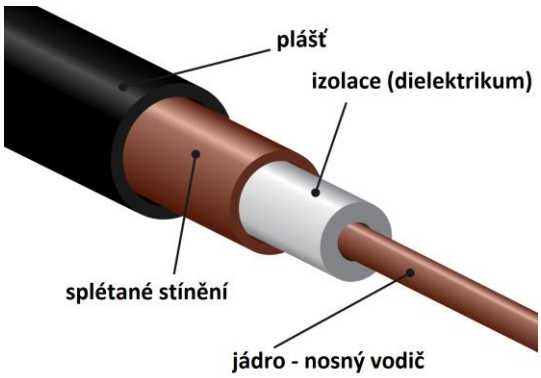
5.8 Základní typy telekomunikačních vedení a jejich charakteristika[8]

5.8.1 Koaxiální kabel

Základní obecná struktura koaxiálního kabelu je znázorněna na **Obr. 5-15**. Konkrétní struktura a použité materiály u koaxiálního kabelu ovlivňují řadu parametrů, např. frekvenční vlastnosti a měrný útlum, fyzické vlastnosti jako např. ohebnost a pevnost a také samozřejmě cenu. Využívána je elektrická vodivost koaxiálního kabelu (zpravidla je použita měď). Měrný útlum je daný ohmickými a induktivními vazbami a je velmi závislý na kmitočtu začíná na úrovni 3 dB/km.

Existuje velké množství různých typů kabelů, které se v mnoha ohledech liší. Obecně lze říci, že koaxiální kabel se používá pro digitální přenosy obvykle na kratší vzdálenosti do rychlostí 500 Mbit/s, pro přeložené pásmo (analogový přenos) pak do několika Gbit/s. Důležitým parametrem je tzv. charakteristická impedance, která je dána především induktivními a kapacitními vlastnostmi použitých materiálů. Pro přenos v základním pásmu se používá nejčastěji kabel s charakteristickou impedancí 50 Ω a pro přeložené pásmo (typicky televizní technika) s charakteristickou impedancí 75 Ω. Koaxiální kabel byl využíván i ve starších specifikacích Ethernetu.

Velkou výhodou koaxiálního kabelu je dobrá ochrana přenášeného signálu před vnějším elektromagnetickým rušením, která vyplývá z jeho konstrukce.



Obr. 5-15: Obecná struktura koaxiálního kabelu (převzato)

5.8.2 Symetrický kabel

Tento typ kabelu (slangově kroucená dvojlinka) se skládá z jednoho či více párů vodičů, které mohou být různým způsobem uspořádány. Každý pár (pokud je využit) tvoří jeden samostatný elektrický obvod. Vodiče jsou krouceny v párech z důvodu potlačení vnějšího elektromagnetického rušení. Kabely existují v několika provedeních, přičemž základní je tzv. nestíněná kroucená dvojlinka (UTP = *Unshielded Twisted Pair*). Další varianta, resp. skupina variant již pak obsahuje určitou formu stínění a nazývá se stíněná kroucená dvojlinka (STP = *Shielded Twisted Pair*), případně fóliově stíněná kroucená dvojlinka (FTP = *Foil-shielded Twisted Pair*). Grafické znázornění je ukázáno na **Obr. 5-16**.

[8] V této kapitole je záměrně vynecháno či opomíjeno jedno velice důležité přenosové médium, a to volný prostor (vzduch resp. vakuum). Problematika přenosů tímto médiem je velmi rozsáhlá a popis média relativně složitý. Popis je nad rámec tohoto textu.

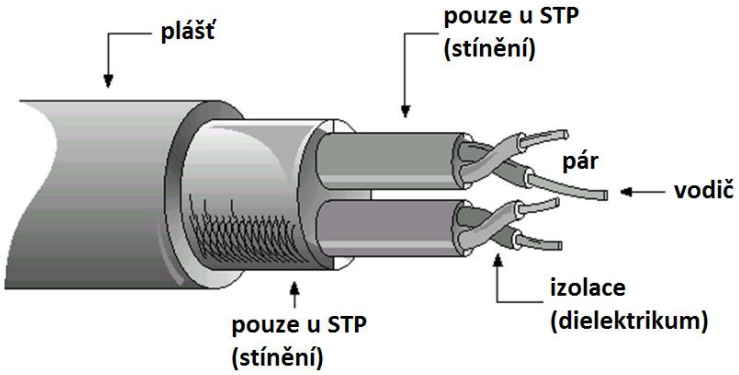
Jelikož vodiče jsou z mědi, využívána je taktéž elektrická vodivost. Měrný útlum daný ohmickými a induktivními ztrátami začíná od 0,5 dB/km a je závislý na frekvenci. Charakteristická impedance se pohybuje okolo hodnoty 100 Ω.

Nestíněný kabel se v telefonní síti využívá pro kmitočty do 3,4 kHz pro klasický telefon, při použití xDSL technologií pak až do několika MHz. V lokálních sítích při rychlostech nad 10 Mbit/s je maximální délka 100 m (bez regenerace). Nestíněná varianta má velkou výhodu v ceně a jednoduchosti instalaci, avšak je citlivější na šum a rušení (i než koaxiální kabel).

Stíněná varianta je dle předpokladů dražší, odolnější, ale také fyzicky silnější a náročnější pro instalace, avšak ochrana proti rušení je na velmi dobré úrovni. Stíněná varianta byla využívána v průmyslovém prostředí, v sítích Token ring a existují již standardy pro její použití v 10 Gbit/s Ethernetu.

U symetrického kabelu se můžeme potkat také s tzv. kategoriemi, které určují především požadavky na vnitřní uspořádání párů v rámci jednoho kabelu. V počítačových sítích je stále nejběžnější kategorie 5E (Cat5E), avšak je možné se již potkat i s kategorií 6 (Cat6). Rozdíl je pak především v možné maximální přenosové rychlosti přes tento kabel, což souvisí např. s útlumem, přeslechy, ale i dalšími parametry. Např. šířka pásma je u kategorie 5e minimálně 100 MHz, zatímco u kategorie 6 musí být alespoň 250 MHz.

U prakticky všech dnes používaných kroucených dvojlinek jsou k dispozici celkem 4 páry.



Obr. 5-16: Struktura symetrického kabelu (převzato)

5.8.3 Optický kabel

U optického kabelu je použito sklo (nebo někdy i plast s vhodnými parametry) k vedení světelných impulzů. Přenos je založen na úplném odrazu světla ve vlákně, odolnost proti elektromagnetickému rušení je enormní, neexistují přeslechy, šum je minimální a v případě některých kabelů je útlum velice malý, což umožňuje i přenosy na poměrně velké vzdálenosti (i stovky kilometrů bez jakýchkoliv úprav). Při srovnání s metalickými kabely je dále třeba zdůraznit vyšší cenu, ale zároveň vyšší teoretickou přenosovou rychlost na stejnou vzdálenost, potřebu speciálních zařízení (vysílače, přijímače) a složitější spojování, či oddělování vláken.

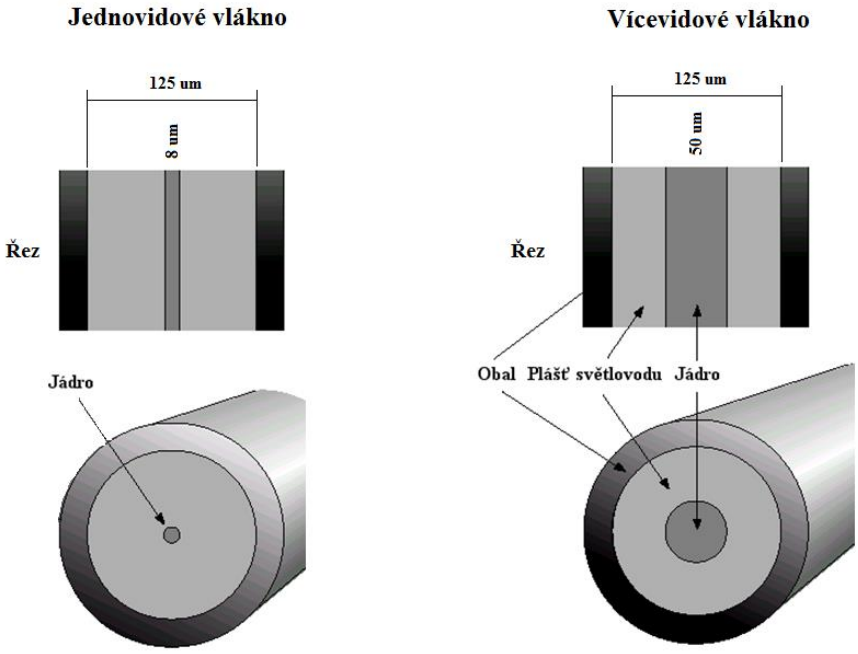
Optický kabel zpravidla obsahuje více optických vláken. Každé vlákno se pak skládá z jádra, pláště a dalších (ochranných) vrstev obalu. Jádrem prochází světelný signál, plášť slouží jako odrazová vrstva (s nižším indexem lomu, než má jádro). Jelikož jádro i plášť jsou velmi tenké (100 až 200 μm), jsou zapotřebí další vrstvy pro dosažení vhodné pevnosti (obal).

Jako vysílače jsou nejčastěji používány lasery anebo LED diody. Na přijímací straně se pak jedna zpravidla o fotodiody. U optických vláken lze díky vhodným multiplexním technikám dosáhnout přenosových rychlostí v řádu Gbit/s až Tbit/s.

Optická vlákna se dělí na dvě základní skupiny:

- **jednovidová** (*single-mode*)
  - jádro má průměr 8-10 μm, plášť ~ 100 μm
  - útlum může být i jen 0,15 dB/km
  - disperse[9] světla je menší než u vícevidového vlákna
  - vysílačem lasery
  - větší vzdálenosti (~ 100 km), vyšší cena
- **vícevidová** (*multi-mode*)
  - jádro má průměr 50-60 μm, plášť ~ 100 μm
  - útlum 0,5 – 2 dB/km
  - větší disperze
  - vysílačem LED diody
  - existují dva typy – s konstantním a s proměnným indexem lomu
  - menší vzdálenosti (~ 1 km), nižší cena

Optické kabely jsou v současné době masivně využívány jak v klasických telekomunikacích, tak v čistě datových sítích, a to především u páteřních tras. Ukázka vnitřních vrstev jednovidového i vícevidového vlákna je naznačena na **Obr. 5-17**.



Obr. 5-17: Vnitřní struktura jednovidového a vícevidového optického vlákna (převzato)

[9] Disperze je pro optické přenosy významný jev ovlivňující vlastnosti přenosové trasy (optického vlákna), kdy rychlost šíření světla (vlny) je závislá na jeho frekvenci (vlnové délce, resp. nepřesně řečeno na „barvě“).

5.9 Přístup koncových zařízení k fyzické vrstvě

Práci a úkony související s fyzickou vrstvou má u koncových zařízení na starost síťová karta nebo síťové rozhraní. V angličtině se potkáváme se zkratkou **NIC** (*Network Interface Card*). Síťová karta typicky pracuje nejen na fyzické úrovni, ale i na spojové.

Síťové karty se liší podle toho, do jaké infrastruktury (síťové architektury) stanici připojují, a to zejména podle toho, s jakým přenosovým médiem daná síť pracuje. Existují síťové karty pro všechny druhy médií uvedená v kap. 5.1 a 5.8. U koncových zařízení se můžeme setkat např. se **síťovými kartami pro** technologie využívající:

- **koaxiální kabely** – dnes však již považovány za zastaralé,
- **symetrické kabely** (kroucené dvojlinky) – v běžné praxi nejběžnější,
- **optická vlákna** – zejména u serverů s potřebou velmi rychlého připojení,
- **bezdrátové prostředí** – zejména u mobilních zařízení. V této oblasti existuje velké množství standardů.

Je zřejmé, že každá karta pak disponuje jiným typem konektoru, případně konektorů. Pro koaxiální kabel je tzv. BNC konektor, pro symetrický kabel velmi známý RJ-45 a u optických vláken se u konektorů nejčastěji setkáme s LC (případně SC) konektorem, resp. jejich párem (pro obousměrný přenos). U bezdrátových přenosů pak rozhraní nemá konektor, ale anténu (a často i několik).

5.10 Síťové prvky na fyzické vrstvě

Všechny existující síťové prvky se musejí vypořádávat s úkoly fyzické vrstvy. Avšak většina současných prvků pracuje i na vyšších vrstvách a podle té nejvyšší je také běžně řadíme. Existují však dva jednoduché síťové prvky, které pracují pouze na fyzické vrstvě a jsou to:

- **opakovač** (*repeater*) – zařízení s dvěma porty, jeden vstupní a druhý výstupní,
- **rozbočovač** (*hub*) – zařízení s více než dvěma porty.

Úkolem těchto síťových prvků je prostá obnova (regenerace) signálu po jeho přenosu a jeho opětovné vyslání na (další) médium. V případě rozbočovače pak i vytvoření kopií signálu na všechna připojená média (s výjimkou toho, po kterém signál přišel).

Opakovače ani rozbočovače nejsou schopny porozumět tomu, co je obsahem zpráv a do přenosového systému vnášejí jen velmi malé zpoždění. Ani jeden z těchto prvků nemá žádnou formu adresy a není možné ho v přenosové trase přímo rozpoznat.

Opakovače umožňují snadno „prodlužovat“ nebo spíše „nastavovat“ přenosové vedení. Tyto prvky jsou velmi často využívány u optických přenosů a díky nim je možné přenášet signály i na velmi dlouhé vzdálenosti, např. přes podmořské kabely. Rozbočovače pak byly využívány zejména v prvotních metalických sítích, ale pro svoji neefektivitu jsou dnes využívány pouze okrajově.

6 Spojová vrstva přenosových systémů

6.1 Úloha spojové vrstvy

Spojová vrstva je druhou vrstvou ISO/OSI modelu. Jak bylo uvedeno v kap. 3.9.2, tato vrstva mění prostý tok bitů na spolehlivou cestu přenosu datových bloků – rámců, popř. buněk či bloků. Spojová vrstva proto **zajišťuje vytvoření, udržení a uvolnění spojení** mezi entitami síťové vrstvy. Implementace spojové vrstvy je závislá na druhu sítě (rozdíly jsou především v technologiích a topologiích).

Jednou z hlavních funkcí spojové vrstvy je **odstraňování chyb**, ke kterým může dojít na fyzické úrovni. Pracujeme zde s tzv. bitovou chybovostí (BER = *Bit Error Rate*), která je vyjádřena jako poměr chybně přenesených bitů ku počtu všech přenášených bitů. Chybovost přenosů se může pohybovat řádově v rozmezí 10^{-3} až 10^{-14}. Je samozřejmé, že u dat požadujeme, aby byly přeneseny správně všechny bity. Chybovost je obecně vždy vyšší v bezdrátovém prostředí[10], nižší u metalických vedení a nejnižší u optických vedení. Vzhledem k charakteru chyb a vlastnostem konkrétní technologie se používá takový protokol, který umožní efektivní komunikaci na dané lince.

Pro efektivní komunikaci na lince **musí spojová vrstva zajistit**:

- **rámcovou synchronizaci** – data jsou vysílána v blocích nazývaných rámce. Rámec je základní jednotka na úrovni spojové vrstvy. Začátek a konec rámce musí být identifikovatelný.
- **řízení toku dat** – kdy vysílací stanice nesmí vysílat rámce rychleji, než je přijímací stanice schopna je přijímat, příp. obě stanice musí umět adaptivně reagovat na situaci.
- **řízení chybových stavů** – chyby vzniklé v přenosovém řetězci musí být rozpoznány a opraveny.
- **adresování** – uzly dané sítě musí mít nějakou jednoznačnou identifikaci. To má význam zejména na mnohabodovém spoji.
- **multiplexovaný provoz** – přenos dat i řídicích signálů procházející stejným kanálem. Přijímač musí umět rozlišit, která data jsou řídicí a která uživatelská.
- **řízení spoje** – tj. sestavení, udržování v chodu a ukončení spojení mezi stanicemi.

6.2 Podvrstvy spojové úrovně

Spojová vrstva se zpravidla rozčleňuje do dvou podvrstev – **podvrstvy řízení logického spoje** (**LLC** = *Logical Link Control*) **a podvrstvy řízení přístupu k přenosovému médiu** (**MAC** = *Media Access Control*).

[10] Příčiny této vyšší chybovosti jsou především: šum, interference, zkreslení, špatná synchronizace, útlum a efekty spojené s vícecestným šířením signálu. Tyto jevy mají v různých prostředích různý dopad na chybovost, avšak zpravidla právě v bezdrátovém prostředí je jejich vliv nejvýznamnější z důvodu otevřenosti tohoto prostředí.

6.2.1 Podvrstva LLC

Podvrstva LLC (*Logical Link Control*) poskytuje rozhraní mezi konkrétním přenosovým prostředkem a síťovou vrstvou. Tato podvrstva se stará o multiplexování požadavků síťové vrstvy, které mohou přicházet od různých protokolů třetí vrstvy (zejména IP, ojediněle také IPX nebo Appletalk). LLC umožňuje těmto protokolům koexistovat nad jednou infrastrukturou. Tato vrstva se dokáže postarat také o kontrolu toku dat a řízení chybových stavů mezi koncovými uzly (tyto funkce však v rámci modelu TCP/IP spadají do mnohem vyšší vrstvy – transportní).

6.2.2 Podvrstva MAC

Podvrstva MAC (*Media Access Control*) poskytuje služby specifické pro daný přenosový prostředek, což jsou zejména použité kódování a přenosové schéma, adresování či práce s rámcem. V případě sítí s mnohonásobným přístupem pak do podvrstvy MAC spadá řešení problematiky přístupu k médiu s ohledem na ostatní uzly sítě (sdílení kapacity, řešení kolizí).

6.3 Režimy komunikace v spojové vrstvě

V spojové vrstvě existují celkem dva základní typy komunikace dvou stran z hlediska obousměrnosti. Jsou to **simplexní spojení** (*simplex*), a **duplexní spojení** (*duplex*). Tyto dva způsoby byly již popsány v kap. 4.5, proto se k nim nebudeme znovu vracet.

6.4 Vytváření rámců

Rámec (spolu s buňkou a blokem) **představuje základní jednotku, se kterou pracuje spojová vrstva**. Protokoly spojové vrstvy potřebují v souvislosti s rámci ke své funkci obvykle tyto řídicí informace:

- které uzly spolu komunikují,
- kdy komunikace začíná a kdy končí,
- zda došlo při přenosu k chybám,
- kdo bude komunikovat jako další.

Rámec má obvykle **tři hlavní části**, a to:

- **datová část** – typicky paket, jehož tvar je nezávislý na přenosové technologii,
- **záhlaví** (*header*) – obsahující řídicí informace na začátku rámce, tvar závislý na konkrétní technologii. Záhlaví se obvykle skládá z více polí, z nichž nejdůležitější jsou:
  - **začátek rámce (preambule,** *flag***)** – slouží k identifikaci začátku celého rámce na médiu, předem daná sekvence jedniček a nul[11].
  - **adresy** – zdrojová a cílová, identifikace komunikujících uzlů.

[11] Pozn.: Toto pole se v některých zdrojích uvažuje samostatně, tj. nikoliv jako část záhlaví.

V záhlaví se mohou vyskytovat i další pole určené např. pro řízení toku dat, obsahující informace o protokolu vyšší vrstvy, zahlcení, délce datové části, či určená k řízení logických spojů.

- **zápatí** (*trailer*) – obsahující řídicí informace na konci rámce, taktéž závisle na použité technologii. Zápatí je obvykle použito k zjištění, zda není rámec poškozen (neobsahuje chyby) a také k identifikaci konce rámce. Setkáváme se s položkami:
  - **kontrolní sekvence rámce** (FCS = *Frame Check Sequence*), tj. pole sloužící k detekci chyb při přenosu[12]. K vytvoření kontrolní sekvence jsou velmi často využívány tzv. cyklické redundantní kódy (CRC = *cyclic redundancy check*).
  - **vlastní zápatí** – sloužící k identifikaci konce celého rámce, stejně jako v případě preambule se jedná o předem danou sekvenci jedniček a nul. Toto pole je zbytné v případech, kdy záhlaví rámce obsahuje informaci o délce datové části. V těchto případech je zřejmé, kde rámec končí, a není proto třeba k tomuto účelu využívat speciální sekvenci.

Záhlaví spolu se zápatím představuje nezbytnou režii přenosu (tato pole neobsahují žádná uživatelská data). Jelikož požadavky a vlastnosti v různých přenosových prostředích se liší, nelze vytvořit jeden univerzální tvar rámce. V prostředí náchylnějším na chyby je třeba větší režie přenosu k zajištění přijatelné úrovně kvality přenosu, zatímco ve spolehlivém prostředí si vystačíme s jednodušším tvarem. V době vývoje počítačových sítí v průběhu druhé poloviny 20. století se vyvinulo několik přístupů k problematice vytvoření rámců, z nichž tři jsou popsány v následujících podkapitolách.

6.4.1 Rámec protokolů Bisync a PPP

Nejstarší pohled na rámce je jako na řadu bajtů (oktetů) a tento přístup je využíván např. protokolem *Binary Synchronous Communication* (BSC, nebo Bisync) a novějším *Point-to-Point Protocol* (PPP).

U **Bisync protokolu** se konkrétně jedná o tzv. znakově orientovaný protokol, kdy existují speciální řídicí znaky (každý vyjádřen právě jedním bajtem). V protokolu Bisync existuje 5 možných formátů rámce, z nichž jeden je znázorněn na **Obr. 6-1**. Prakticky všechny ostatní v současné době využívané protokoly mají pouze jeden hlavní formát rámce. Vysvětlení řídicích znaků následuje níže.

8 b	8 b	8 b		8 b		8 b	16 b
SYN	SYN	SOH	Záhlaví	STX	Tělo (data)	ETX	CRC

Obr. 6-1: Jedna z možných struktur rámce u Bisync protokolu

Vybrané řídicí znaky u protokolu Bisync:

- **SYN** (*Synchronization*) – označení začátku rámce,
- **SOH** (*Start of Header*) – značka za kterou následuje záhlaví,
- **STX** (*Start of Text*) – označení začátku těla zprávy,

[12] Pozn.: V některých zdrojích může být FCS uváděno samostatně, nikoliv jako část zápatí.

- **ETX** (*End of Text*) – označení konce těla zprávy,
- **CRC** (*Cyclic Redundancy Check*) – není přímo řídicí znak, ale pole sloužící k detekci chyb při přenosu.

Problémem tohoto přístupu je, že v datové části se může objevit bajt, který je zároveň použit pro vyjádření hodnoty ETX. Špatná detekce konce zprávy není žádoucí, a proto musí protokol na výskyt této řídicí sekvence v datové části reagovat, zpravidla přidáním dalšího speciálního symbolu. Tento způsob vkládání dodatečných bajtů do dat se nazývá *stuffing* a je možné se s ním potkat i v jiných systémech.

Protokol Bisync je primárně určen pro poloduplexní znakový přenos dat, při kterém se střídá dotaz a odpověď. V současné době se lze s tímto protokolem potkat jen ojediněle, avšak principy v něm obsažené je možné nalézt i v modernějších řešeních.

Naproti tomu **protokol PPP** je stále hojně využíván na různých typech spojů bod-bod. Formát rámce protokolu PPP, ve verzi vycházející z HDLC (popsáno v kap. 6.4.2), je ukázán na **Obr. 6-2**. Pole Flag slouží jako návěst (značka) začátku a také konce celého rámce. Pole adresa a řízení obsahují zpravidla fixní hodnoty, a proto nejsou až tak důležitá (jsou obsažena pro budoucí nebo speciální využití). Pole protokol obsahuje informaci o protokolu vyšší vrstvy, který je obsažen v datové části. Pole CRC má stejný účel jako u všech ostatních spojových protokolů. Zajímavostí protokolu PPP je, že pořadí polí je sice pevně dáno, avšak jejich délka může být mezi komunikujícími stranami upravena.

Protokol PPP umožňuje ve spolupráci s dalšími protokoly i autentizaci, šifrování či kompresi přenášených dat. Přenos je u protokolu PPP duplexní. PPP protokol má dvě podvrstvy:

- **LCP** (*Link Control Protocol*) – správa spojení, tj. např. dojednání parametrů přenosu,
- **NCP** (*Network Control Protocol*) – vlastní přenos dat.

8 b	8 b	8 b	8 b		16 b	8 b
Flag	Adresa	Řízení	Protokol	Tělo (data)	CRC	Flag

Obr. 6-2: Struktura rámce u PPP protokolu

6.4.2 Rámec protokolu HDLC

HDLC (*High-Level Data Link Control*) je taktéž spojovým protokolem, avšak s rámcem pracuje jako s tokem bitů. Základní struktura rámce je velice podobná protokolu PPP, (resp. platí, že záhlaví PPP protokolu vychází z rámce HDLC). Struktura rámce je pevná, pro názornost je možné nahlédnout na **Obr. 6-3**.

Návěst začátku a konce rámce je „01111110“ a je zřejmé, že protokol musí zajistit, aby se tato sekvence nevyskytla v datové části, jelikož by to bylo chybně rozpoznáno jako konec rámce. To se řeší opět stuffingem (tzv. *bit stuffing*), což znamená vkládání bitů v případě potřeby tak, aby nebyla v datech přítomna návěst (*flag*). Je třeba si uvědomit, že tato vlastnost protokolů v praxi znamená, že přesná velikost rámce je dána jeho datovým obsahem a že není možné, aby byly rámce vždy stejně dlouhé.

HDLC může být používáno i na spojích bod – více bodů, avšak platí, že nyní je protokol využíván zejména na propojení dvou zařízení. HDLC je protokol vyvinut organizací ISO, avšak existují i jeho rozšíření, které vytvořila např. firma Cisco.

8 b	8 b	8 b		16 b	8 b
Flag	Adresa	Řízení	Tělo (data)	CRC	Flag

Obr. 6-3: Struktura rámce u standardního HDLC protokolu

6.4.3 Rámec standardu Ethernet

Protokol Ethernet je to nejpravděpodobnější, s čím se u lokální sítě můžeme potkat. Jeho specifikace je poměrně široká a neustále se vyvíjejí nové a modernější standardy. Ethernet je technologie pro sítě s vícenásobným přístupem, z čehož vyplývá i formát rámce, na který se v této kapitole zaměříme.

U síti typu Ethernet se ve skutečnosti můžeme setkat s několika základními formáty rámce, které se však liší jen poměrně málo. Dva nejdůležitější a patrně nejčastěji používané jsou:

- **IEEE 802.3 Ethernet**

8 B	6 B	6 B	2 B	46 až 1500 B	4 B
Preambule	Cílová adresa	Zdrojová adresa	Délka	Data	FCS

Obr. 6-4: Formát rámce u Ethernetu IEEE 802.3

- **Ethernet II**

8 B	6 B	6 B	2 B	46 až 1500 B	4 B
Preambule	Cílová adresa	Zdrojová adresa	Typ	Data	FCS

Obr. 6-5: Formát rámce Ethernet II

S oběma rámci se lze dnes běžně potkat v rámci jedné sítě, přestože běžnější je spíše rámec Ethernet II[13]. Rozdíl mezi rámci je především u pole, které je v obrázcích zaznačeno zelenou barvou – tj. „délka“ u IEEE 802.3 a „typ“ u Ethernet II. Pole „délka“ podle očekávání informuje o tom, jak dlouhá je datová část rámce (rozsah je proměnný). Pole „typ“ naproti tomu informuje zařízení o typu vyššího protokolu, který je v rámci rámce přenášen (typicky IP paket). Rozlišení rámců probíhá na základě hodnoty, kterou nalezneme v tomto poli. Pokud je hodnota (po převedení na dekadickou hodnotu) nižší než 1500, pole obsahuje délku dat

[13] V datové části rámce IEEE 802.3 mohou být ukryta další linková záhlaví, sloužící k větší míře řízení komunikace na této vrstvě (LLC = *Logical Link Control*). Další podverze (802.3 SNAP) je pak často využívána u proprietárních linkových protokolů sloužících speciálním účelům.
Naproti tomu u rámce Ethernet II se již další linková záhlaví neočekávají.

(1500 je maximální délka dat), pokud je hodnota vyšší, je v této hodnotě uložen kód typu protokolu v datové části. Pozn.: u rámce Ethernet II je rozpoznání konce postaveno pouze na kódování na fyzické vrstvě, kde je použit speciální kód umožňující zasílat mimo dat i určité řídicí sekvence.

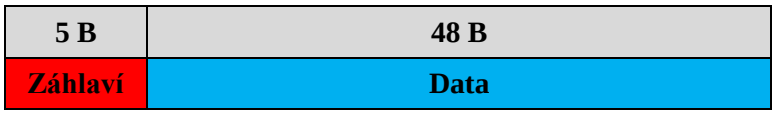
Ostatní pole z obrázků mají následující význam:

- **preambule** – obsahuje i pole SFD (*Start Frame Delimiter*), určení začátku rámce, synchronizace příjemce, oddělení hlavní části záhlaví
- **cílová a zdrojová adresa** – pole obsahují adresy komunikujících uzlů. Cílová adresa je důležitá pro konkrétní rámec, aby mohl být doručen k adresátovi, zdrojová adresa má význam pro případnou odpověď.
- **datová část** – obsahuje typicky další záhlaví (vyšší protokoly) a vlastní data.
- **FCS** (*Frame Check Sequence*) – kontrolní sekvence rámce, který je vytvořen z celého rámce s výjimkou preambule a SFD.

6.4.4 Rámec technologie ATM

Jak již bylo uvedeno, ATM (*Asynchronous Transfer Mode*) je technologie založená na komutaci rámců (buněk), viz kap. 3.2, a využívající tzv. asynchronní přenosový režim, viz kap. 4.4.1. Tato technologie se tedy snaží kombinovat výhody komutace okruhů (tj. garance kapacity přenosového kanálu a konstantní zpoždění přenosu) s vybranými vlastnosti komutace paketů (tzn. flexibilita a efektivita). V současné době je tato technologie na ústupu, avšak z důvodu představení dalšího možného přístupu ke stanovení struktury rámce je zahrnuta v tomto textu. Výraznou charakteristikou ATM je systém čtyř tříd přenosu, resp. možných kvalit služby, které definují požadavky na parametry přenosu u konkrétního datového toku. Jedná se o režimy CBR (*Constant Bit Rate*), VBR (*Variable Bit Rate*), ABR (*Available Bit Rate*) a UBR (*Unspecified Bit Rate*). Bližší vysvětlení tříd přenosu je však již nad rámec tohoto textu.

Buňka této spojově orientované technologie má **fixní** délku 53 B. Buňka se skládá ze dvou základních částí. Je to záhlaví o délce 5 B a datovou část o délce 48 B, viz **Obr. 6-6**. Je tedy zřejmé, že se jedná o velmi krátké buňky, teoreticky velmi vhodné např. pro přenos digitalizovaného hlasového signálu. Pozn.: Téměř 10 % přenosové kapacity každé buňky je vyhrazeno pro servisní informace dané komunikace (záhlaví), což je ve srovnání s dalšími zde uvedenými technologiemi nejvíce.



Obr. 6-6: Základní formát rámce technologie ATM

Záhlaví se běžně skládá z celkem šesti položek různé délky. Jsou to:

- **GFC** (*Generic Flow Control*), 4 bity – určité lokální funkce a řízení, např. sdílení ATM rozhraní, nicméně typicky nepoužíváno.
- **VPI** (*Virtual Path Identifier*), 8 bitů – spolu s VCI definuje přenosovou trasu pro danou buňku.

- **VCI** (*Virtual Channel Identifier*), 16 bitů – spolu s VPI definuje přenosovou trasu pro danou buňku, VCI reprezentuje konkrétní kanál v rámci dané trasy.
- **PT** (*Payload Type*), 3 bity – rozlišení uživatelských či řídicích dat, indikace dalších stavů souvisejících s přenosem dat.
- **CLP** (*Cell Loss Priority*), 1 bit – identifikátor zda má být buňka zahozena v případě extrémního zahlcení sítě.
- **HEC** (*Header Error Control*), 8 bitů – kontrolní kód se schopností detekce bitových chyb ve zbytku záhlaví a i schopností opravy v případě pouze jedné bitové chyby.

6.4.5 Rámec technologie Frame Relay

Frame Relay je WAN technologie založená na komutaci paketů, viz kap. 3.2, avšak pracující ve spojovaném režimu (spojově orientovaný přenos), tzn. mající některé vlastnosti podobné jako komutace okruhů. Spojení je u Frame Relay označováno jako virtuální okruh (VC = virtual circuit) a je reprezentován identifikátorem označovaným jako DLCI (*Data Link Connection Identifier*). Existují dva typy VC: PVC (*Permanent Virtual Circuit*) jsou natrvalo zřízeny v dané WAN síti a SVC (*Switched Virtual Circuit*), které se automaticky sestavují při požadavku na přenos dat. Technologie Frame Relay je stejně jako ATM na ústupu a můžeme se s ní setkat v praxi pouze ojediněle. Nicméně opět kvůli dalšímu možnému typu rámce je zařazena do tohoto textu.

Rámec Frame Relay má proměnnou délku. To přináší flexibilitu obdobně jako např. u Ethernetu. Základní struktura je naznačena na **Obr. 6-7**.

8 b	16 b	až 1600 B	16 b	8 b
Flag	Adresa	Data	FCS	Flag

Obr. 6-7: Základní formát rámce technologie Frame Relay

Popis jednotlivých položek rámce:

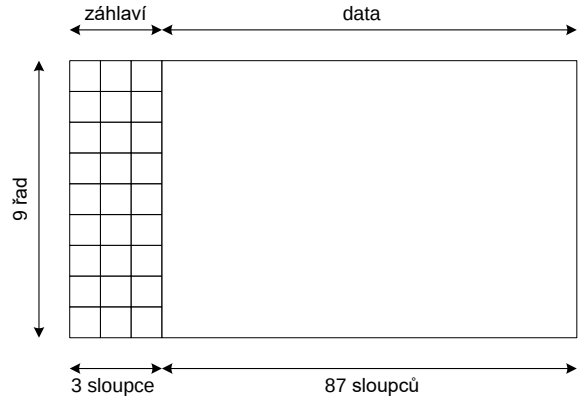
- **Flag** (návěští), 8 bitů – návěští začátku a konce rámce, fixní sekvence „01111110“.
- **Adresa**, nejčastěji 16 bitů – dělí se na další podčásti:
    - **DLCI** (*Data Link Connection Identifier*), nejčastěji 10 bitů – identifikátor virtuálního okruhu, kterým bude rámec přenášen.
    - **EA** (*Extended Address*), nejčastěji 2 bity – indikace použití rozšířeného formátu DLCI.
    - **C/R** (*Command response bit*), 1 bit – nepoužíváno.
    - **Congestion Control**, 3 bity – notifikace případného zahlcení Frame Relay sítě včetně uvedení typu zahlcení.
- **Data** – přenášená data, pole proměnné délky, komunikující strany si mohou domluvit konkrétní hodnotu maxima, nicméně standardně je doporučeno maximum 1600 B.
- **FCS** (*Frame Check Sequence*), 16 bitů – CRC kód umožňující detekci bitových chyb v rámci.

6.4.6 Vytváření rámců založené na časové synchronizaci (SDH, SONET, GPON)

SDH bylo krátce zmíněno v kapitole 5.7.4, SONET (*Synchronous Optical Network*) je jeho obdobou na Severoamerickém kontinentě. U těchto technologií je přístup k vytváření rámců založen na přesné časové synchronizaci (*clock-based framing*). Obě tyto technologie jsou primárně navrženy pro využití v telefonních společnostech a pro přenos většího množství 64 kbit/s telefonních kanálů, avšak jejich použití je nyní širší, např. pro přenos rámců jiné technologie, např. Ethernetu, IP paketů nebo ATM, jak bylo zmíněno i v 4.3.4.

Nejnižší jednotka, se kterou se můžeme u těchto systémů potkat je STM-0 (STS-1), které se týká následující popis.

U STM-0 je linková přenosová rychlost 51,84 Mbit/s a kapacita pro přenos dat pak 50,112 Mbit/s. Rozdíl mezi těmito dvěma hodnotami je určen pro záhlaví rámce, které se u těchto technologií nazývá *overhead*. Rámec má zde pevnou délku 810 B, z čehož 27 B je záhlaví. Oproti ostatním technologiím je unikátní to, že záhlaví je rozprostřeno v rámci celého rámce. Přijímač se snaží detekovat očekávanou posloupnost, která tvoří významnou část záhlaví a pokud se mu to povede několikrát po sobě v intervalu 810 B, má vyhráno (je synchronizován a rozpoznal začátky rámců). Časově je jednomu rámci vyhrazeno 125 µs. Z výše uvedeného popisu je patrné, že u SDH ani u SONETu se nevyužívá žádná forma *stuffingu*, jelikož by pak nebylo možné zajistit vždy stejnou délku rámců. Schéma rámce u STM-0 je naznačeno na **Obr. 6-8**. Přenos tohoto rámce pak probíhá po řádcích. Jelikož systém pracuje s pevnou délkou rámce, není vždy ideálně efektivní, jelikož ne vždy jsou data k přenosu dlouhá právě tak, jaká je kapacita rámce.



Obr. 6-8: Základní struktura rámce STM-0

Další technologie, založená na přesné časové synchronizaci je GPON (*Gigabit Passive Optical Network*), existující v současnosti v mnoha různých verzích s různými přenosovými rychlostmi a i různými dalšími vlastnostmi. Tato technologie založená na přenosech po optickém vlákně je řazena do kategorie přístupových technologií, tj. technologií pro přenos dat mezi poskytovatelem připojení a koncovým zákazníkem. Specifikace technologií GPON, popř. NG-PON (*Next-generation PON*), jsou velmi rozsáhlé, proto se v rámci této kapitoly zaměříme pouze na zjednodušený popis rámce, který je v této technologii využíván pro přenos dat. Konkrétně si popíšeme pouze rámec v sestupném směru přenosu (*download,* tj. přenos směrem k uživateli). Pozn: rámec ve vzestupném směru se v určitých ohledech liší, avšak jeho popis je již nad rámec tohoto textu.

Základní charakteristikou GPON je, že rámec má **fixní** délku v čase, konkrétně 125 µs. Z toho lze následně při uvažování konkrétní přenosové rychlosti odvodit, kolik je bitová délka jednoho rámce. Např. při jedné z možných přenosových rychlostí, 2,48832 Gb/s na fyzické vrstvě, je bitová délka jednoho rámce 38880 B (převedeno na bajty). Rámce jsou v systému řazeny hned za sebou, což principiálně odpovídá časovému dělení v synchronnímu přenosovému módu (viz kap. 4.4.1), avšak protože je možné i dynamicky měnit počet přidělených slotů jednotlivým účastníkům komunikace a rámce vždy obsahují záhlaví nesoucí řídící a dodatečné informace, princip organizace komunikace více odpovídá časovému dělení v asynchronním přenosovém módu (opět viz kap. 4.4.1).

Základní struktura rámce GPON v sestupném směru je znázorněna na **Obr. 6-9**. Tento rámec má po převedení do kódu na fyzické vrstvě délku právě 125 µs.

<table>
<tr><td>Proměnná délka</td><td colspan="4">Proměnná délka</td><td>4 B</td></tr>
<tr><td rowspan="2">Záhlaví FS</td><td colspan="4">Datová část FS</td><td rowspan="2">FCS (BIP)</td></tr>
<tr><td>GEM rámec</td><td>GEM rámec</td><td>…</td><td>GEM rámec</td></tr>
</table>

Obr. 6-9: Základní formát rámce technologie GPON v sestupném směru

Záhlaví FS (*Frame Sublayer*) má poměrně dosti složitou strukturu, jejíž detailní popis je nad rámec tohoto textu. Zjednodušeně řečeno, obsahuje řadu polí týkajících se řízení přístupu k médiu z hlediska vzestupného směru, rozdělení kapacity kanálu mezi jednotlivé účastníky, které může být rovnoměrné i nerovnoměrné a řadu dalších identifikátorů. V datové části v sestupném směru jsou naskládány tzv. GEM (*GPON Encapsulation Method*) rámce. Tyto rámce mají opět proměnnou velikost, avšak jejich velikost v součtu musí být nastavena tak, aby vyplnily celou datovou část uvedenou v obrázku, jejíž délka je závislá na konkrétní přenosové rychlosti, jak již bylo vysvětleno výše. Tyto GEM rámce mají vlastní záhlaví (fixní délky 8 B), které sdružuje různé identifikátory a řídící informace, následované datovou částí proměnné délky, běžně v délce odpovídající Ethernetovému rámci bez preambule (až 1518 B). Konečně pole FCS (*Frame Check Sequence*), u GPON označované jako BIP (*Bit Interleaved Parity*), představuje jeden ze základních způsobů detekce případných bitových chyb při přenosu (viz kap. 6.6.3), kterých je však u GPON rámce definovaných více.

Celkově lze GPON rámec označit za rámec založený na časovém dělení s poměrně velkou režií přenosu (záhlaví FS může mít délku až 2078 B, další bajty jsou u záhlaví každého z GEM rámců), relativně velkou variabilitou přenosových režimů, s čímž souvisí i relativní složitost dané technologie. Dalším důvodem, proč byl GPON rámec zařazen do tohoto textu, je ukázka možnosti sestavení větších rámců za pomocí kratších rámců, což bývá u technologií založených na časovém dělení poměrně časté.

6.4.7 Umístění rámce na sdílené médium

Jestliže je přenosové médium sdíleno více uzly, není zpravidla možné, aby všechny vysílaly současně (pokud není použita některá z metod vícenásobného přístupu, o kterých bylo pojednáno v kap. 4.1, které nyní neuvažujeme). Bez jakékoliv kontroly by se však velice často stávalo, že by se několik zařízení pokoušelo vyslat svoje rámce na médium. Metody starající se o řízené umisťování rámců na společné médium jsou známy jako metody řízení přístupu na médium (*media access control*). Protokolů existuje více a základní odlišnosti jsou dány především tím, s jakým médiem pracují. Základním úkolem těchto protokolů je

definovat způsob, jak umožnit uzlům médium sdílet. Bez těchto pravidel by nebyla možná žádná komunikace, velice často by docházelo ke kolizím (poškození rámců). K pochopení významu je možné si představit paralelu, jak by asi vypadala doprava na běžných silnicích bez jakýchkoliv pravidel, semaforů, značení atd.

Metody přístupu na médium jsou závislé na:

- způsobu, jak je médium sdíleno mezi uzly sítě,
- topologii sítě na úrovni spojové vrstvy.

Metody přístupu na médium jsou děleny na:

- **metody s vysokým stupněm kontroly** – maximální snaha o vyhýbání se kolizím. Tyto metody samy o sobě však zpravidla způsobují velké zpomalení komunikace a přináší nezanedbatelnou režii.
- **metody s nízkým stupněm kontroly** – i zde je snaha o vyhýbání se kolizím, avšak určitá míra chyb je tolerována (a následně řešena). Režie komunikace je nižší, což v běžných aplikacích vede zpravidla ke zrychlení.

Dělení dle náhodnosti metody (související s předcházejícím) je na:

- **deterministické metody** (plně řízené) – u těchto metod je typické, že v daný okamžik přenáší data pouze jedna stanice a každá stanice před zahájením přenosu musí čekat, až přijde na řadu. Stanice se tedy v možnosti vysílání nějakým způsobem střídají. Každá stanice má tedy zaručenu určitou přenosovou kapacitu, a dokud neskončí přenos rámce stanice, která byla na řadě, k žádnému dalšímu přenosu nedochází. Systém je velmi dobře předvídatelný, avšak poměrně neefektivní. S deterministickými přístupy se můžeme setkat u kruhových topologií (Token Ring, FDDI).
- **nedeterministické metody** (založené na vzájemném soutěžení uzlů) – zjednodušeně platí, že stanice mohou zkusit vysílat v libovolný okamžik, v systému tedy existují kolize. Metody proto musí obsahovat mechanizmy, jak tyto kolize řešit. Aby se množství kolizí v systému snížilo, tyto metody typicky obsahují i mechanizmy, jak detekovat již probíhající přenos – *Carrier Sense Multiple Access* (CSMA). Pokud nějaký přenos již probíhá, snaha o další vysílání by vedla ke kolizi, což není žádoucí. V případě detekce přenosu tedy stanice krátký časový interval počká a poté pokus opakuje.

  Ke skutečné kolizi dojde tehdy, jestliže je médium prázdné a dvě nebo více stanic začnou vysílat ve stejný časový okamžik. Pokud se tato situace opakuje často, výrazně to snižuje přenosovou kapacitu, v extrémním případě může dojít až k zablokování sítě. Metody fungují poměrně efektivně zejména v případech, kdy intenzita přenosů na sdíleném množství není přehnaně vysoká. S nedeterministickými přístupy se můžeme setkat u Ethernetových i Wi-Fi sítí.

  U standardního Ethernetu se potkáváme s variantou CSMA/CD (*Collision Detection*), která principiálně odpovídá výše uvedenému popisu. U sítí Wi-Fi (standardy 802.11) je využívána přístupová metoda CSMA/CA (*Collision Avoidance*), která spočívá v tom, že uzly se snaží kolizím vyhýbat. Základní rozdíl oproti CSMA/CD je, že uzel po zjištění, že je médium prázdné, předtím než zahájí vlastní přenos dat, informuje ostatní stanice

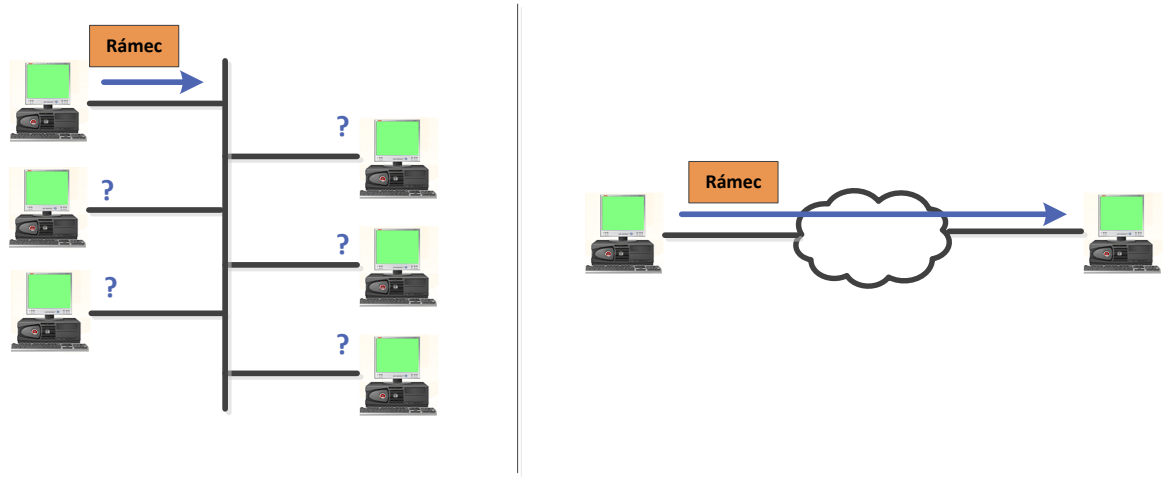
o tom, že bude vysílat. To množství kolizí výrazně sníží. Je však zřejmé, že metoda obnáší určité snížení přenosových rychlostí.

6.5 Adresace spojové vrstvy

Adresy na úrovni spojové vrstvy jsou často používány při transportu rámců po lokálním médiu (typicky sdíleném). Adresy na této úrovni jsou někdy označovány jako **fyzické adresy** nebo také **hardwarové adresy**, s těmito adresami se však vždy pracuje až na úrovni spojové vrstvy. Jak je patrné z popisu různých typů rámců (viz kap. 6.4), adresa je uložena v záhlaví rámce, přičemž specifikuje cíl rámce na lokální síti. Je velmi vhodné do záhlaví umístit i adresu odesílatele, aby bylo možné na danou zprávu snadno odpovědět.

Spojové (linkové) adresy jsou používány pouze pro lokální adresování v rámci dané sítě, za hranicemi této sítě nemají žádný význam. Jestliže daný uzel přesuneme do jiné sítě, stále bude mít stejnou linkovou adresu, avšak to bude opět platná pouze lokálně. Rámec ve své původní podobě neopustí nikdy danou síť, a pokud jsou data v něm obsažená určená uzlu mimo uvažovanou síť, musí být na hranici sítě vytvořen rámec nový. K tomu dochází tak, že je z rámce vytažena datová část a ta je pak zapouzdřena do rámce nového, jehož formát záleží na technologii, která je v následující síti použita.

Adresování má velký význam na topologiích s vícenásobným přístupem, kde více uzlů sdílí společné médium. Jednotlivá zařízení tak snadno poznají, zda je rámec určen pro ně nebo jiný uzel a zároveň platí, že pokud v síti existují nějaká propojující zařízení, mohou být linkové adresy využity k tomu, aby rámce směřovaly směrem ke svému cíli vhodnou cestou. Situace je graficky demonstrována na **Obr. 6-10**a.



a) b)

Obr. 6-10: a) K vysvětlení potřeby adresování na síti s vícenásobným přístupem, b) situace na síti bod-bod

Naproti tomu u topologií bod-bod, kde je síť tvořena pouze přímým propojením dvou uzlů je z principu adresování zbytečné (viz **Obr. 6-10**b). Nicméně i v těchto případech se můžeme setkat s rámci, které adresování obsahují (viz kap. 6.4), např. z důvodu větší univerzálnosti daného protokolu.

6.5.1 Základy adresace u technologie Ethernet a 802.11 (Wi-Fi)

Nejčastější technologie, s kterými se dnes můžeme na úrovni lokálních sítí setkat, jsou bezesporu Ethernet a také sítě 802.11 (Wi-Fi). Oba protokoly existují v několika verzích, Ethernet může fungovat jako síť s vícenásobným přístupem, standard 802.11 přímo reprezentuje síť tohoto typu, z čehož vyplývá potřeba adresování. Způsob adresace je u obou protokolů velice podobný a není závislý na verzi protokolu.

Každý uzel této sítě disponuje unikátní adresou, která souvisí se síťovým rozhraním, které uzel do této sítě připojuje, a běžně je zapsána v jeho paměti. Adresa je přednastavena výrobcem síťového rozhraní a je celosvětově unikátní. Délka adresy je 48 bitů (6 bajtů), z čehož vyplývá, že počet možných adres je $2^{48} \approx 3.10^{14}$. Adresa je běžně označována jako **MAC adresa**, linková adresa či fyzická adresa, občas i nesprávně jako logická nebo síťová adresa. Adresy je možné zapisovat binárně, běžně se však využívá pouze hexadecimální zápis, který je výrazně stručnější. Např. adresa v binárním formátu:

01010000 11100101 01001001 00111000 10011101 10001111

Je pro člověka snáze zapsatelná a čitelná jako:

50:E5:49:38:9D:8F

Jak je patrné z příkladu, běžně je adresa zapisována po bajtech, tj. hexadecimální zápis tvoří šest částí, oddělených dvojtečkou (někdy pomlčkou).

Unikátnost adresy na lokální síti je zajištěna i v případě, kdy jsou zde použity síťové rozhraní od různých výrobců. Jednotliví výrobci mají totiž přiděleny určité rozsahy, které se nepřekrývají. MAC adresa je proto dělena na dvě části, kdy prvních 24 bitů reprezentuje kód výrobce a dalších 24 bitů pak kód konkrétní karty. Pokud známe MAC adresu, lze z ní zpětně zjistit, kdo je výrobcem rozhraní.

6.6 Techniky detekce chyb

6.6.1 Míra chybovosti a její vliv na přenos

Při přenosu rámců po médiu může docházet k chybám. Jak bylo uvedeno v kap. 6.1, chybovost se pohybuje řádově v rozmezí 10^{-3} až 10^{-14}. To znamená, že na jeden špatně přenesený bit připadá 10^3 až 10^{14} bitů celkově přenesených bitů[14]. Za bitovou chybu považujeme záměnu logické „1“ za logickou „0“ nebo obráceně. Tyto chyby mohou nastávat ojediněle nebo shlukově, což má různé následky. Existují však i další typy chyb, které mohou např. způsobit, že bude špatně detekován konec rámce, což vyplývá z popisů rámců uvedených v kap. 6.4.

Z výše uvedených hodnot se zdá, že v případech, kde je chybovost velmi malá (10^{-14}), je pravděpodobnost chyby téměř nulová a nemělo by být nutné nějak se jí speciálně zabývat. Je však zřejmé, že i jeden změněný bit může mít na přenášená data fatální dopady, a proto je určitá forma obrany proti chybám přítomna prakticky u všech linkových rámců (viz kap. 6.4). Uveďme si dva číselné příklady. U velmi rychlých optických linek je přenosová rychlost v řádu desítek Gbit/s a chybovost v řádu 10^{-14}, což obnáší průměrně 0,0004 chyb za sekundu (při rychlosti 40 Gbit/s). Druhým příkladem by byl bezdrátový kanál o rychlosti 1 Mbit/s

[14] Chybovost na úrovni 10^{-3} až 10^{-6} je z praktického hlediska velmi vysoká a může mít na přenos fatální následky.

s extrémní chybovostí 10^{-5}. Snadno lze spočítat, že průměrně bude za jednu vteřinu 10 bitových chyb.

V praxi je však důležitější počet při přenosu poškozených rámců, než perioda bitové chybovosti. Pro jednoduchost výpočtu vezměme jako příklad velikost rámce 1250 B, což je 10 000 bitů. Uvažujme např. chybovost 10^{-5} a 10^{-14}. V prvním případě je pravděpodobnost poškození alespoň jednoho bitu v rámci rovna 10^4 x 10^{-5} = 10^{-1}, tedy 10 %. To je velmi vysoké číslo, a proto je u reálných přenosových systémů snaha dosáhnout chybovosti alespoň na úrovni 10^{-6}, což pro výše uvedené hodnoty znamená příznivější 1 % pravděpodobnost chyby v rámci. V druhém případě je pravděpodobnost poškození rámce rovna 10^4 x 10^{-14} = 10^{-10}, tedy velmi nízkých 0,00000001 %. Pozn.: Tyto výpočty nejsou zcela přesné, k přesným výsledkům je třeba využít aparát složené pravděpodobnosti, přesné výsledky jsou však velmi podobné.

6.6.2 Základní přístupy k detekci chyb při přenosu

Jedno z možných řešení detekce chyb by bylo, že by se každý rámec přenášel dvakrát. Příjemce by rámce porovnal, a pokud budou stejné, je relativně vysoká pravděpodobnost, že k žádné chybě při přenosu nedošlo. Pokud budou rámce různé, k chybě došlo zcela určitě, avšak není možné rozpoznat, zda je alespoň jeden z rámců v pořádku, případně který. Velkou nevýhodou tohoto systému mimo vysokou neefektivitu je i špatná odolnost vůči periodickým chybám, které poškodí oba přenášené rámce stejným způsobem a tyto se pak jeví jako stejné, avšak ve skutečnosti jsou oba poškozené. Systém může být nastaven i tak, že je přenášeno a porovnáváno větší množství kopií stejného rámce, což však ještě zvyšuje neefektivitu a neodstraňuje problém s periodickými chybami.

V případě, že detekuje příjemce chybu, musí se s tím nějak vypořádat. Existují dva základní přístupy. První z nich je, že příjemce po detekci chyby požádá odesílatele o opakované zaslání toho stejného rámce a původní zahodí. Pokud je pravděpodobnost chyb nízká, je vysoká pravděpodobnost, že opakovaný přenos bude úspěšný a také že celkový počet opakovaných přenosů bude velmi nízký. Druhou metodou je, že obsah rámce je nastaven tak, že je možné (při nízké úrovni chyb) opravit chyby na straně příjemce automaticky[15].

Pozn.: Z výše uvedeného popisu vyplývá, že opakované přenášení rámců snižuje reálnou propustnost sítě, což je taktéž jednou z příčin, proč není propustnost sítě rovna (teoretické) přenosové rychlosti.

Základní myšlenkou všech mechanizmů detekce a i oprav chyb je přidání určité redundantní informace do rámce. Tato informace umožňuje spojové vrstvě na straně příjemce rozpoznat, zda při přenosu došlo nebo nedošlo k chybě.

6.6.3 Metody zabezpečení proti chybám při přenosu

Nejjednodušším způsobem zabezpečení jsou tzv. **paritní bity**. Parita funguje tak, že se vezme 7 bitů zprávy a spočítá se počet jedniček ve zprávě. Následně pak je k řetězci původních 7-mi bitů přidán bit paritní, který reprezentuje informaci o počtu jedniček ve zprávě (zda je sudý nebo lichý, podle typu parity, která je použita). Např. sekvence „1010001“ (se třemi jedničkami) bude v případě sudé parity doplněna o další jedničku tak, aby počet jedniček byl sudý, tj. výsledkem bude posloupnost „10100011“. Pokud je parita

[15] Zde se používají tzv. dopředné korekční kódy, avšak tato problematika je nad rámec tohoto textu. Tyto kódy jsou využívány např. u xDSL, GPON či WiMAX technologie.

lichá, bude stejná posloupnost doplněna na „10100010“, čímž celkový počet jedniček zůstane lichý. Je zřejmé, že parita není příliš spolehlivý prostředek detekce chyb, postačuje, aby došlo v rámci jednoho bajtu ke dvěma chybám, a tato změna zůstane nedetekována. Parita je proto využívána spíše doplňkově.

Pozn.: Existují i složitější mechanizmy využívající paritu, základní myšlenka však zůstává stejná.

Kontrolní součty představují o něco vyšší míru zabezpečení. Jak vyplývá z názvu, provádí se součet celého rámce (např. po bajtech) a výsledek je uložen jako kontrolní hodnota za rámec (kontrolní bajt). Tento mechanizmus bohužel není odolný vůči záměně pořadí bitů, jelikož součet se změnou pořadí bitů nezmění.

Lepším řešením jsou proto tzv. **cyklické redundantní kontroly** (CRC = *cyclic redundancy check*), které jsou používány u prakticky všech spojových protokolů, jak je patrné z kap. 6.4. Tyto kontroly (často nazývány ne zcela správně jako kontrolní součty) umožňují velice dobrou detekci chyb po přenosu rámce. Základní matematický aparát, který je u CRC použit, je dělení polynomu polynomem a do pole FCS (*Frame Check Sequence*) je pak ukládán zbytek po tomto dělení. Z tohoto důvodu potřebuje v případě CRC pouze velmi malou redundanci (např. 4 bajty (tj. 32 bitů) CRC kódu (označováno jako CRC-32) na 1500 bajtů dat v rámci). Navíc platí, že tyto kódy jsou postaveny tak, že pravděpodobnost detekce chyby je velmi vysoká, téměř 100 %.

Fungování protokolů využívajících CRC je takové, že jak vysílač, tak příjemce znají používaný algoritmus. Vysílač před odesláním rámce spočítá kontrolní sekvenci a přidá ji k rámci jako pole FCS. Příjemce pak u přijatého rámce taktéž spočítá kontrolní sekvenci, a pokud obdrží stejný výsledek, rámec je považován za neporušený. Pokud jsou kontrolní součty různé, je třeba spustit nápravné mechanizmy, které, jak již bylo uvedeno, spočívají zejména v opakování přenosu.

6.7 Spolehlivý přenos

6.7.1 Řízení chybových stavů

Jestliže se na problematiku chyb při přenosu podíváme z mírně vyššího pohledu, nemusí docházet pouze k chybám uvnitř rámce, jak bylo popisováno v předchozích kapitolách. Rozlišujeme proto:

- **poškození rámce** – uvnitř rámce došlo k bitové chybě a tato chyba je rozpoznána (např. díky CRC),
- **ztráta celého rámce** – rámec vůbec není detekován na straně příjemce nebo jej není možné rozpoznat.

Z předcházejících kapitol je zřejmé, jak se vypořádat s poškozeným rámcem. Avšak jak postupovat pokud není žádný rámec obdržen? Pokud příjemce žádný rámec neobdrží, zpravidla ani neví, že ho měl očekávat. Musí proto existovat nějaký nadřazený mechanizmus, který rozhoduje o případném opakování vysílání i v těchto případech a který umožní spojové vrstvě se se ztrátami rámců vypořádat.

Tyto techniky velice často souvisí i s regulací toku dat a v praxi se s nimi setkáváme velice často až na transportní vrstvě. Mohou se však vyskytovat i na vrstvě spojové, a proto se jimi budeme nyní dále zabývat.

Ne všechny spojové protokoly poskytují spolehlivý přenos. Některé z nich ponechávají detekce ztracených rámců (a tedy i dat) na vyšších vrstvách, typicky na transportní úrovni (je možné se s těmito mechanizmy setkat i na aplikační úrovni). To pak spojovou vrstvu zjednodušuje.

Základní předpoklady a fakta pro řízení chybových stavů a toku dat jsou:

- **Velikost vyrovnávací paměti příjemce není neomezená** (počet rámců, které je možné vyslat pro určitého příjemce v rámci jednotky času, je omezen),
- **Delší rámec = větší pravděpodobnost výskytu chyb**, riziko opakovaného přenosu rámce roste,
- **Kratší rámec = rychlejší detekce chyb**, což také souvisí s tím, že omezená velikost rámců je výhodná i z pohledu opakovaného přenosu vyslaných dat (v případě opakování se přenáší méně dat),
- **Stanice nemůže blokovat médium na neomezeně dlouhou dobu** (pokud je využíváno sdílené médium). To má příznivý vliv i na zpoždění celé komunikace,
- Pokud přenášíme velké objemy dat, vždy je **snaha dosáhnout co nejmenší chybovosti** (alespoň 10^{-9} až 10^{-10}), což zpravidla vyžaduje pevné spoje.

Možné systémy detekce ztracených rámců jsou:

- **Kladná potvrzení** – zasílání kladných potvrzení pro bezchybně přijaté rámce. Přijímač potvrzuje přijaté rámce. Ty, které nejsou potvrzeny (byly ztraceny nebo poškozeny), jsou ze strany vysílače přeneseny opakovaně. K opakování dochází až po vypršení určitého časového intervalu (časovač, *timeout*). U této metody je intenzita přenosů mezi příjemcem a vysílačem poměrně vysoká, což má za následek lepší vazbu mezi komunikujícími stranami, ale zároveň i vyšší zatížení přenosových kapacit. S tímto systémem se v komunikačních protokolech potkáme častěji.
- **Záporná potvrzení** – zasílání záporných potvrzení doplněných o žádost o opakované vysílání rámců. V tomto případě je vysílač kontaktován pouze v případě problémů. Tento způsob obnáší slabší vazbu mezi komunikujícími stranami. Je zřejmé, že pokud přijímač nereaguje, nemusí to být vždy bezchybný stav. Výhodou je nižší zatížení přenosových kapacit.

V souvislosti s metodami detekce ztracených rámců a řízení přenosu dat se setkáváme s technikami, jako jsou ARQ (*Automatic Repeat reQuest*), tj. automatická žádost o opakování a klouzavé (posuvné) okno (*Sliding Window*). Druhá z těchto technik souvisí úzce s ARQ, které navíc existuje v celkem třech variantách: *stop-and-wait ARQ (SW)*, *Go-Back-N ARQ* (*GBN*) a *Selective Repeat ARQ (SR)*. Tyto techniky popisují následující kapitoly.

6.7.2 Stop-and-wait ARQ (SW)

Tato metoda je tím nejjednodušším, co si lze v případě řízení toku dat představit. Vysílač a přijímač pracují sekvenčně. Vysílač odešle rámec a následně čeká, dokud nemá od

příjemce potvrzení o přijetí (*acknowledgment*). Pokud při přenosu nedošlo k žádné chybě, je toto potvrzení bráno zároveň jako signál, že příjemce je připraven přijmout další rámec. Pokud k chybě došlo, po upozornění od příjemce se vysílač pokusí rámec doručit znovu. Je zřejmé, že vysílač nemůže vysílat rámce libovolně, ale odesílání rámců je vlastně krokováno příjemcem, podle toho jak potvrzuje dříve odeslané rámce.

Systém umožňuje v jeden okamžik přenášet pouze jeden rámec, tento systém je tedy velice jednoduchý, avšak bohužel to obnáší velkou neefektivitu z hlediska využití přenosové kapacity. Obzvláště je to patrné na delších přenosových kanálech, kde hraje významnou roli doba průchodu signálu fyzickým médiem (tam a zpět, tj. projeví se dvakrát). Přesto je možné se s ním potkat u několika řešení a i na jiné než spojové vrstvě, např. u aplikačního protokolu TFTP (*Trivial File Transfer Protocol*).

Detekce chyb u přijatých rámců probíhá pomocí standardních mechanizmů popsaných v kap. 6.6.3, typicky CRC. Vysílač musí také pracovat s určitým časovačem (*timeout*), pokud dojde ke ztrátě celého rámce (ať už původního nebo potvrzujícího). Situace je demonstrována na **Obr. 6-11**, kde jsou ukázány jednak všechna zpoždění, se kterými při přenosu pracujeme (dohromady tvoří také hodnotu RTT, která byla zmíněna v kap. 3.6), zejména však situace se ztrátou přenášeného rámce při přenosu (levá část) a následně také ztrátou potvrzení o doručení (pravá část).

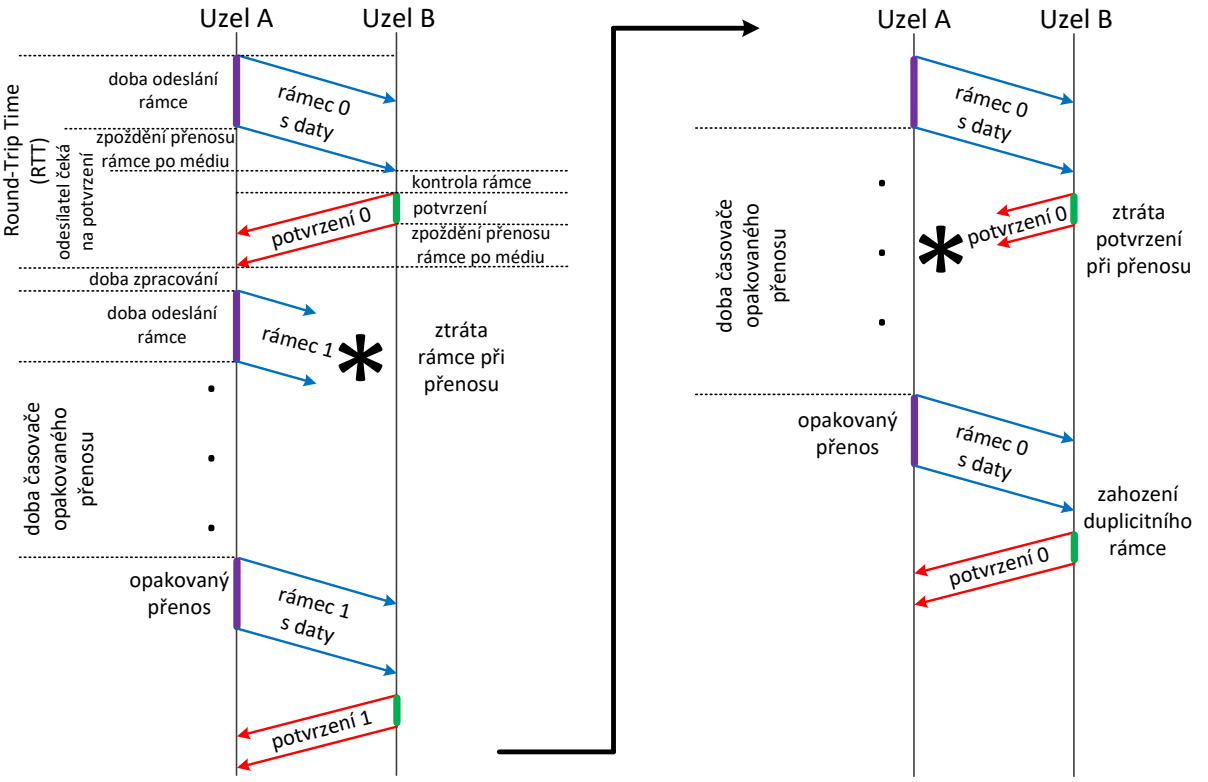
Z obrázku lze vyčíst, že vhodně zvolená doba časovače opakovaného přenosu je velmi důležitá. Pokud by byl časovač příliš krátký, bude se přenos zbytečně opakovat dřív, než bude schopen příjemce doručit potvrzení o přijetí a naopak, pokud by byl časovač příliš dlouhý, bude se zbytečně dlouho čekat a celý přenos tak bude brzděn.

V obrázku se u rámců objevují čísla „0“ a „1“. Mechanizmus *stop-and-wait* totiž musí nějakým způsobem rozlišovat sudé a liché rámce, zejména proto, aby příjemce byl schopen snadno rozpoznat duplicitní rámec (viz pravá část obrázku). V rámci záhlaví metody *stop-and-wait* je tedy nutné vyhradit jeden bit pro tento účel.

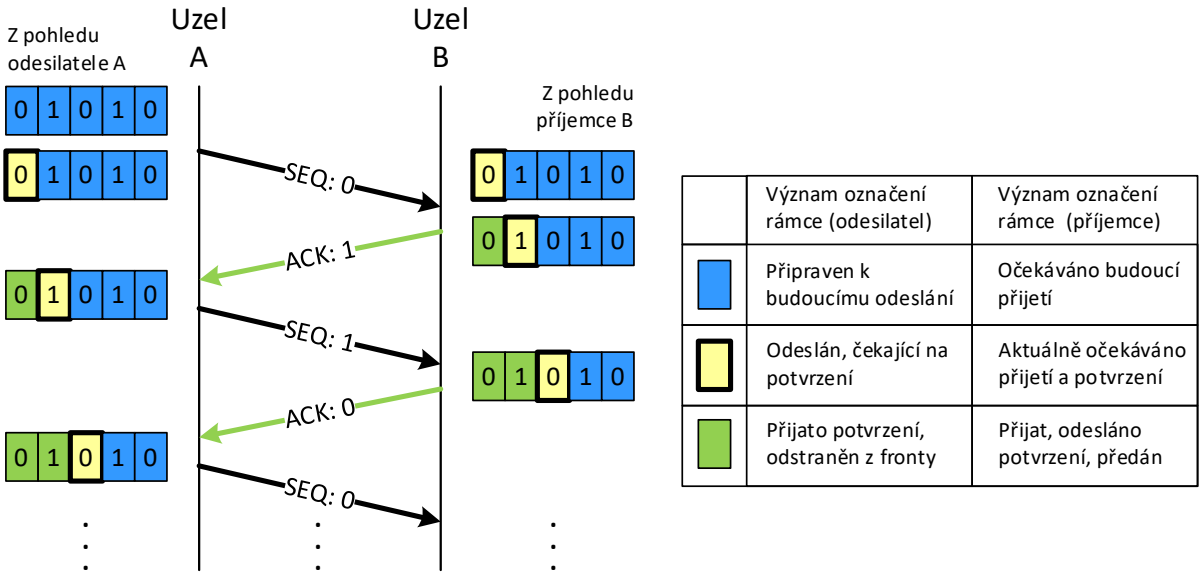
Situaci s přenosem několika rámců z pohledu řazení a stavu jednotlivých rámců na straně odesilatele a příjemce ukazuje **Obr. 6-12**. V ukázce je u odesilatele k odeslání nachystáno 5 rámců, přičemž přenos a potvrzení prvních dvou jsou v grafice zachyceny. Je důležité si uvědomit, že žádná z komunikujících stran nevidí, co se děje na přenosové trase nebo u protistrany. Stav a fungování komunikace vyhodnocuje pouze podle zpráv, které (ne)dorazí. Proto je barevné označení rámců rozlišeno na pohled odesilatele a příjemce. Rámec, který dosud nebyl potvrzen, je nutné si zachovat pro případný opakovaný přenos. Jakmile potvrzení (ACK, často také s číslem dalšího očekávaného rámce, jak je uvedeno v obrázku) dorazí, je možné rámec z fronty vyřadit a věnovat se dalšímu rámci.

Z obou obrázků je patrná vysoká neefektivita metody *stop-and-wait* ARQ, přenosové médium je dlouhé časové úseky nevytíženo. Jak již bylo uvedeno, největší problém to pak představuje na delších přenosových trasách, kde doba přenosu rámce, bude již velmi dlouhá (relativně vzhledem k ostatním časům v systému). Navíc je potřeba vzít v potaz, že tato doba se na přenosu jednoho rámce projeví vždy dvakrát, protože vždy čekáme na přenos potvrzení. Pozn.: Při měření RTT je uvažován rámec minimální možné délky, v tomto obrázku není délka rámce specifikována.

Uveďme si příklad. Mějme kanál o teoretické přenosové rychlosti 2 Mbit/s se zpožděním RTT 50 ms. Jestliže budeme používat rámce o délce 10000 bitů, maximální přenosová rychlost vzhledem k použité technologii *stop-and-wait* je rovna podílu počtu bitů v rámci a času potřebnému na přenos rámce, tedy pouhých (10000 / 0,05) = 200 kbit/s. Při těchto parametrech je tedy propustnost pouze $^1/_{10}$ přenosové rychlosti.



Obr. 6-11: Ukázka fungování mechanizmu *stop-and-wait* ARQ (pro jednoduchost pouze jednosměrný přenos dat)



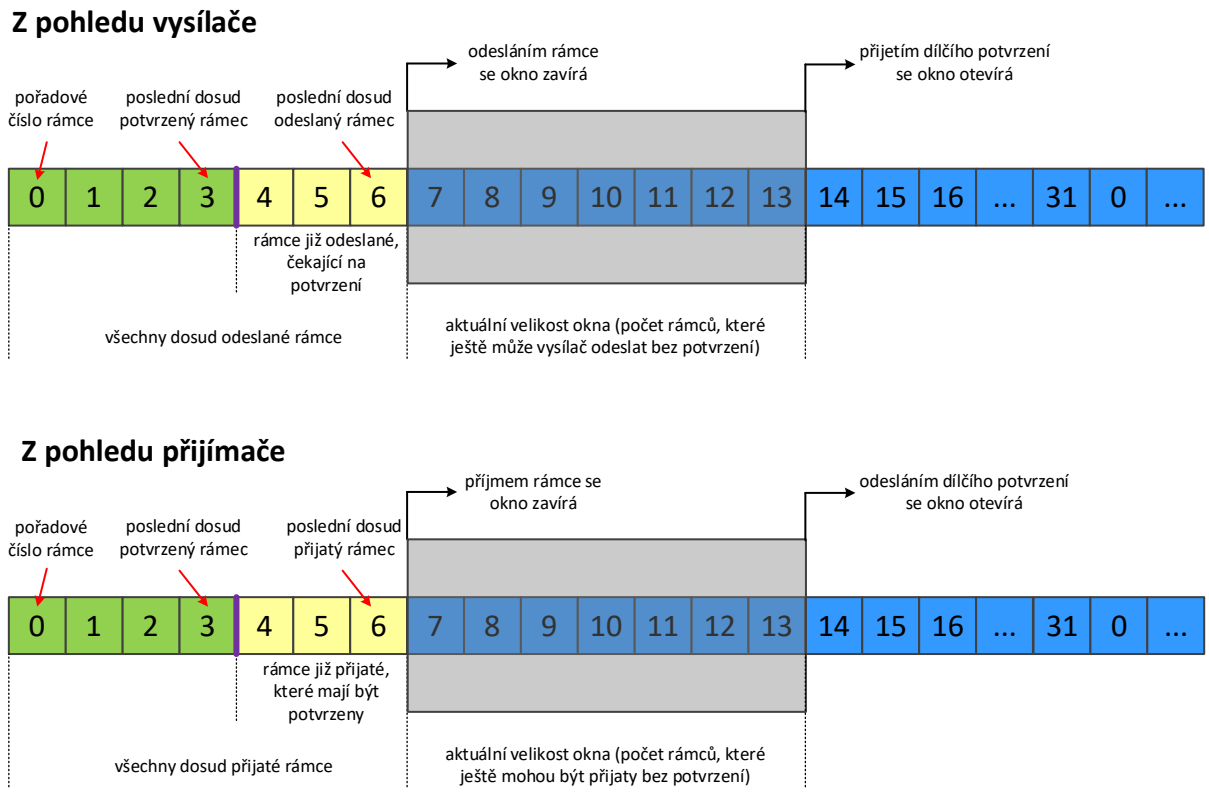
Obr. 6-12: Řazení a stav jednotlivých rámců u mechanizmu *stop-and-wait* ARQ z pohledu odesilatele a příjemce (částečná ukázka bezchybného přenosu několika rámců)

6.7.3 Technika klouzavého okna

Jestliže chceme využít kapacitu kanálu efektivněji, nesmí vysílač čekat na potvrzení každého rámce před tím, než začne vysílat další. Vysílač musí rámce odesílat hned za sebou a průběžně pak dostávat zpět i potvrzení o doručení (při plně duplexní komunikaci). Pokud chceme zachovat kontrolu nad fungováním přenosu, vysílač může v situaci, kdy nemá potvrzení o doručení, vyslat pouze omezený počet rámců a poté musí zastavit (pokud mezi tím žádné potvrzení nepřišlo)[16]. Tato hodnota je nazývána jako **velikost okna** (*window size*).

Pro určení ideálního počtu rámců vyslaných bez čekání na potvrzení jsou důležité dva následující výpočty. První je násobek teoretické přenosové rychlosti a zpoždění, tedy např. 2 Mbit/s . 50 ms = 100 kbit = 12,5 kB. Jestliže vezmeme tuto hodnotu a podělíme ji velikostí rámce (např. 10000 b = 1,25 kB), dostaneme počet rámců, které je třeba vyslat za sebou (bez čekání na potvrzení), aby byla využita celá kapacita kanálu: 12,5 kB / 1,25 kB = 10. Tato hodnota koresponduje i s výpočtem v předcházející kapitole, kde byla vypočtena propustnost při použití základní metody *stop-and-wait* na $^1/_{10}$ přenosové rychlosti.

V případě využití této techniky musí být rámce taktéž nějak číslovány a je zřejmé, že již nevystačíme pouze s jedním bitem (**Obr. 6-13**). Pole musí být vícebitové a např. pro výše uvedená čísla by bylo teoreticky potřeba minimálně čtyři bity[17].



Obr. 6-13: Postup komunikace při využití techniky klouzavého okna z pohledu vysílače i přijímače

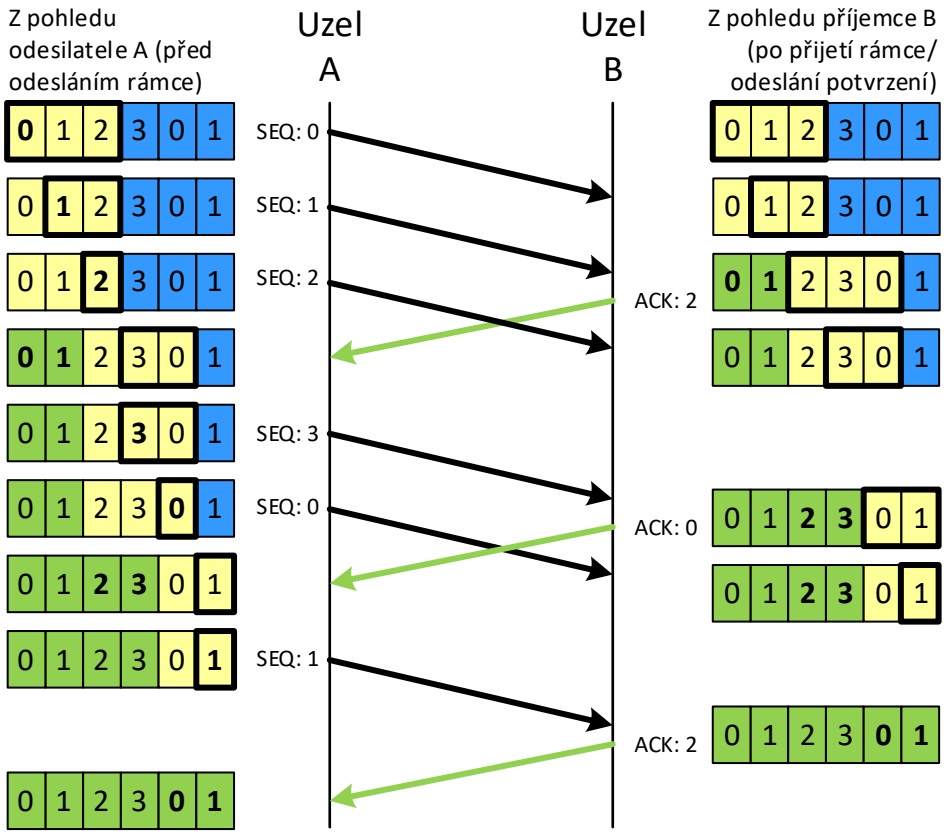
[16] U pokročilých systémů se tento počet může dynamicky měnit v závislosti na potřebě příjemce rámců.
[17] Ve skutečnosti bude potřeba více bitů, viz dále u konkrétních technik GBN a SR.

Vysílač musí vést evidenci o tom, které rámce odeslal, které jsou již potvrzeny a také průběžně sledovat kolik je aktuálně již vyslaných, avšak dosud nepotvrzených rámců. Pokud tento počet rámců dosáhne dohodnuté maximum, musí vysílání dalších rámců pozastavit, dokud neobdrží další potvrzení o úspěšném doručení. Přijímač musí být připraven dohodnutý počet rámců přijmout a každým potvrzením, které zašle, dává najevo připravenost přijímat další rámce. Příjemce může potvrzovat každý rámec zvlášť anebo může být systém nastaven tak, že potvrzuje kumulativně (2 a více rámců). Uveďme si příklad. Dejme tomu, že je maximální velikost okna 10 a byly již odeslány rámce číslované jako 0, 1, 2 a 3. Příjemce může potvrdit přijetí všech těchto rámců jednou zprávou, ve které bude uvedeno, že jako další rámec očekává 4 (ACK: 4). Tím dává najevo, že předchozí rámce přijal a že přenos proběhl bez detekce chyb.

V tomto případě byla uvažována maximální velikost okna právě 10. Veškerý popis je součástí **Obr. 6-13**. Pokud má systém spolehlivě fungovat i v případě ztrát rámců, musí být číslování nastaveno tak, aby velikost okna nebyla příliš veliká vzhledem k maximálnímu číslu rámce. Konkrétní způsob stanovení je uveden u jednotlivých metod uvedených dále.

Pozn.: V reálných systémech může být délka okna vyšší, běžně se pak můžeme setkat s velikostí odpovídající řádově několika desítkám až stovkám rámců.

Konkrétní příklad fungování mechanizmu klouzavého okna bez chyb při přenosu, na kterém je vidět i jednotlivé mezistavy, je uveden na **Obr. 6-14**, který používá stejné barevné rozlišení rámců jako předchozí obrázky.



Obr. 6-14: Konkrétní příklad fungování techniky klouzavého okna z pohledu vysílače i přijímače ve stavu bez chyb přenosu (6 rámců k přenosu, velikost okna rovna 3, příjemce potvrzuje kumulativně vždy 2 přijaté rámce)

Vysílač (**Obr. 6-14**) má 6 rámců k odeslání, velikost okna je 3 rámce. Příjemce nemusí nutně potvrzovat každý rámec zvlášť, může tak činit i kumulativně, v tomto příkladu potvrzuje vždy 2 přijaté rámce. Výraznější rámeček reprezentuje aktuální velikost okna, tučně označené rámce pak ty, které jsou aktuálně odesílány nebo potvrzovány. Uzel A může na začátku díky velikosti okna odeslat ihned za sebou 3 rámce a následně čeká na potvrzení. Doba čekání je ovlivněna způsobem, jakým příjemce potvrzuje (okamžitě, se zpožděním, jednotlivě, kumulativně) a také dobou trvání RTT. V dalším okamžiku vysílač příjme potvrzení ACK: 2, potvrzující přijetí rámců SEQ: 0 a SEQ: 1 ze strany příjemce. Díky tomu může tyto rámce již vyřadit z fronty (nebude potřeba opakovaný přenos) a posunout okno i na rámce SEQ: 3 a SEQ: 0, které vzápětí odešle. Tím dojde opět k vyčerpání velikosti okna a vysílač musí počkat na přijetí dalšího potvrzení. Zbytek komunikace z pohledu vysílače probíhá v obdobném duchu.

Z pohledu přijímače (**Obr. 6-14**) je výchozí stav takový, že je připraven přijmout až 3 rámce. Přijetím každého rámce se okno zavírá, dokud se příjemce nerozhodne odeslat potvrzení. To může udělat až tehdy, když je připraven přijmout další rámce. Jakmile odešle potvrzení rámců (jako první v příkladu jsou to potvrzení za SEQ: 0 a SEQ: 1), může tyto rámce předat vyšší vrstvě. Obdobně postupuje příjemce dále až do konce komunikace.

Výše uvedený popis se vztahuje pouze k ideální situaci, kdy v systému nedochází k chybám. Tyto situace mohou být řešeny různými způsoby, např. v dalších kapitolách popsanými mechanizmy *Go-back-N ARQ* nebo *Selective Repeat ARQ*, které oba používají techniku klouzavého okna. Mimo to je zřejmé, že v případě, kdy dojde k chybě, dojde ke snížení propustnosti přenosu.

6.7.4 Metoda Go-back-N ARQ (GBN)

Tato metoda využívá jako základní mechanizmus techniku klouzavého okna tak, jak byla popsána v předcházející kapitole. Jak vyplývá z názvu, metoda dále pracuje s návratem do určitého stavu. Jak bude vysvětleno dále, jedná se o návrat před stav, kdy došlo při přenosu k chybě. V praxi je možné se s tou metodou potkat velmi často.

Metoda primárně řeší situaci, kdy je přijat rámec s chybou a je třeba, aby příjemce dal najevo vysílači, že je třeba přenos rámce opakovat. Nicméně vzhledem k mechanizmu klouzavého okna je pravděpodobné, že vysílač mezitím vyslal další rámce, což je potřeba vzít v potaz. Jednoduchost metody je postavena na principu, že příjemce zahodí nejen rámec, který byl poškozený, ale i každý další obdržený (i bezchybný), dokud není opakovaně přenesen původně očekávaný rámec. Tento způsob redukuje požadavky na přijímač i vysílač. Přijímač si nemusí pamatovat rámce, které jsou mimo pořadí a následně pořadí rámců přeskládávat. Vysílač musí mít všechny rámce, které dosud nejsou potvrzeny tak jako tak uloženy. Celý systém je tedy nastaven tak, že pokud v některém rámci dojde k chybě, neřeší se problém selektivně, ale návratem o několik kroků zpět a přenos poté pokračuje ve stejném duchu. Je zřejmé, že sice dochází k opakovanému přenosu správně přenesených rámců, avšak v systému s malým množstvím chyb nebo malou velikostí okna to nehraje velkou roli.

Přijímač tedy používá dva druhy zpráv, případně příznaků ve zprávě jedné. Jeden je kladné potvrzení na principu ohlášení čísla dalšího očekávaného rámce (čímž dává najevo úspěšné přijetí těch předcházejících). Tato zpráva bývá někdy označována jako zpráva *Receive Ready* (RR) či *Acknowledgement* (ACK). Druhá je zpráva negativního potvrzení, často označována jako *Reject* (REJ) či *Negative Acknowledgement* (NACK). V této zprávě je opět číslo dalšího očekávaného rámce, který nebyl přijat nebo byl přijat s chybou, a proto je třeba, aby byl přenesen znovu.

Ukázka mechanizmu *Go-back-N ARQ* je na **Obr. 6-15**a. Z obrázku lze vyčíst, k jakým základním chybám může dojít a jak jsou řešeny. Obrázek je určen pouze pro ilustraci chybových stavů, ke kterým může dojít a způsobu jejich řešení. Uvedené schéma nespecifikuje velikost okna, která by teoreticky mohlo přenos rámců omezit. Předpokládáme tedy, že velikost okna je zde dostatečně velká.

Pro Go-back-N (GBN) platí, že velikosti okna (W) musí být menší nebo rovna $2^m - 1$, kde m je počet bitů použitých pro číslování rámců. Tedy $W \leq 2^m - 1$. Jako příklad uveďme, že kdyby byl počet bitů $m = 5$, pak $W \leq 31$. Z opačného pohledu pak platí, že pokud chceme používat např. velikost okna $W = 30$ jednotek (rámců), musíme použít číslování jednotlivých rámců za pomocí alespoň 5bitového číslování ($m = 5$).

Systém *Go-back-N* umožňuje jak potvrzování každého rámce zvlášť, tak kumulativní potvrzování, které je ukázáno v **Obr. 6-14** a i **Obr. 6-15**a. V případě, že bude potvrzován každý rámec zvlášť, bude mezi komunikujícími stranami silná zpětná vazba, která umožní rychlou reakci v případě chyb. Na druhou stranu však toto schéma povede k většímu zatížení zpětného kanálu.

Konkrétnější situaci s GBN ukazuje **Obr. 6-16**. Zde předpokládáme přenos šesti rámců (číslovaných 0 až 5), velikost okna je právě 3 rámce a příjemce potvrzuje každý přijatý rámec zvlášť. V obrázku je uveden i pohled odesilatele na práci s oknem a stavem jednotlivých rámců, stejně tak jako reakce na chybu, kterou je ztracení rámce SEQ: 2. Toto konkrétní schéma předpokládá, že časovač opakovaného přenosu nepotvrzeného rámce byl delší, než doba, za kterou se k vysílači A dostala informace o přijetí rámce mimo pořadí (zpráva REJ: 2). Vysílač se po přijetí této zprávy vrátí zpět k odeslání rámce SEQ: 2 a pokračuje od tohoto bodu dále, tj. rámce 3 a 4, které příjemce mezitím zahodil z důvodu přijetí mimo pořadí, jsou přeneseny znovu. Je třeba si uvědomit, že může dojít nejen ke ztrátě rámce, ale i potvrzení, jak bylo principiálně naznačeno na **Obr. 6-15**a, avšak způsob reakce vždy závisí na konkrétním stavu komunikace, nastavení mechanizmu a také zpoždění přenosu mezi protistranami (RTT).

6.7.5 Metoda Selective Repeat ARQ (SR)

Tato metoda řeší v principu stejné situace jako v předcházející kapitole popsaná *Go-back-N ARQ*. Základní mechanizmy jsou velice podobné, avšak jak již lze odvodit z názvu, dochází u této metody pouze k selektivnímu opakování přenosu. To znamená, že opakovaně jsou přenášeny pouze poškozené rámce a přenos rámců po chybě neprobíhá v původně plánovaném pořadí, pouze se mezi aktuálně odesílané rámce vřadí ten opakovaně přenášený. To dělá řízení komunikace složitější na straně vysílače.

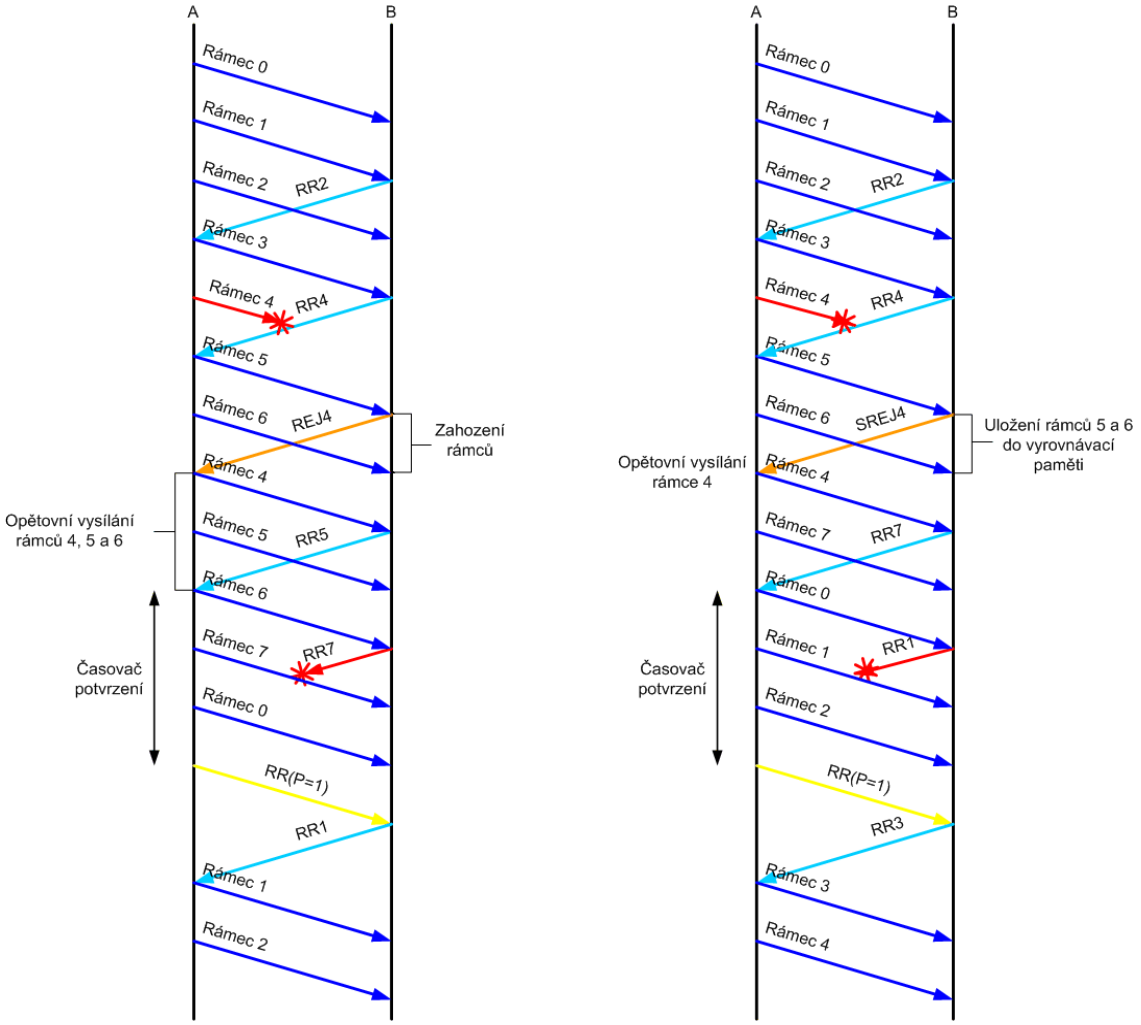
Metoda však komplikuje fungování především na straně příjemce. Spojová vrstva si musí totiž někam ukládat všechny rámce, které byly úspěšně přijaty po rámci s chybou a teprve poté, co dojde k úspěšnému opakování přenosu, předává všechny rámce k dalšímu zpracování vyšším vrstvám. Hlavní výhodou metody je úspora přenosové kapacity, proto se s touto metodu můžeme v praxi také často potkat.

Zpráva, kterou příjemce zasílá poté, co zjistí, že rámec je porušen, se obvykle nazývá *Selective Reject* (SREJ) nebo *Negative ACK* (NACK) a obsahuje číslo očekávaného rámce (tj. toho, co byl poškozen). Vysílač v tomto případě opakovaně přenáší pouze rámce, u kterých obdržel zprávu SREJ a také ty, u kterých případně vyprší hodnota časovače. Druhá situace může nastat např. tehdy, dojde-li ke ztrátě potvrzení. Možná komunikace mezi vysílačem a příjemcem je přehledně naznačena na **Obr. 6-15**b, který obsahuje i příslušný popis. Opět pro jednoduchost předpokládáme, že velikost okna nijak neomezuje probíhající přenos.

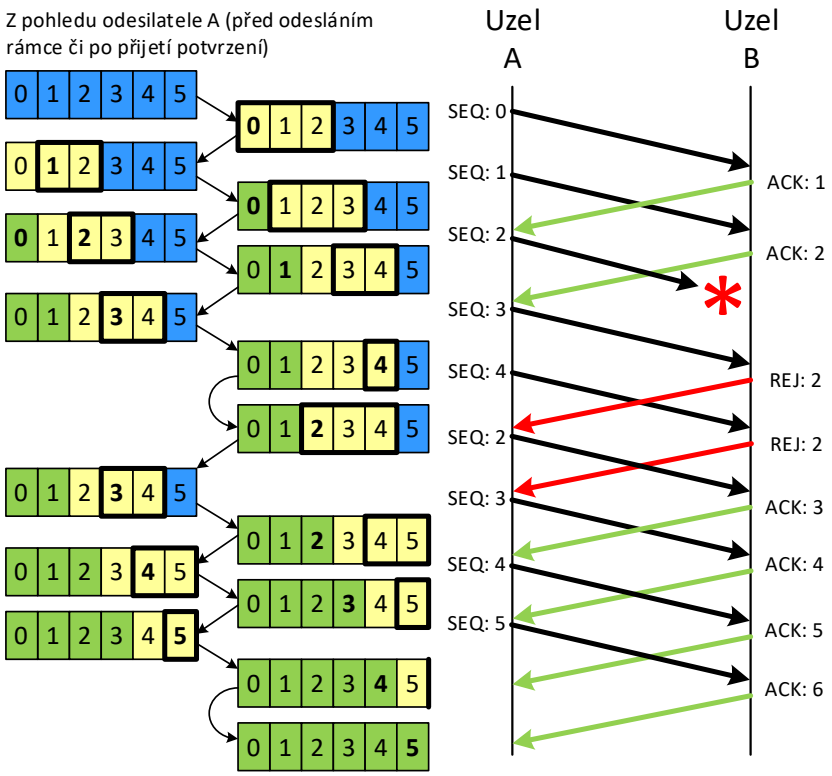
Pro Selective Repeat (SR) platí, že velikosti okna musí být menší nebo rovna 2^{m-1}, kde m je počet bitů použitých pro číslování rámců. Tedy $W \leq 2^{m-1}$. To je méně než u GBN a je to dáno především efektivitou SR, která je dosahována díky větší variabilitě přenosu oproti GBN. Jako příklad uveďme, že kdyby byl počet bitů $m = 5$, pak $W \leq 16$. Z opačného pohledu pak platí, že pokud chceme používat např. velikost okna $W = 30$ jednotek (rámců), musíme použít číslování jednotlivých rámců za pomocí alespoň 6bitového číslování ($m = 6$).

Mechanizmus SR nejlépe funguje v případě individuálního potvrzování rámců, přesto **Obr. 6-15**b ukazuje variantu, kde je používáno kumulativní potvrzování. V případě individuálního potvrzování je v systému silná vazba mezi komunikujícími stranami, která umožňuje dostatečně pružně a efektivně reagovat na případné chyby.

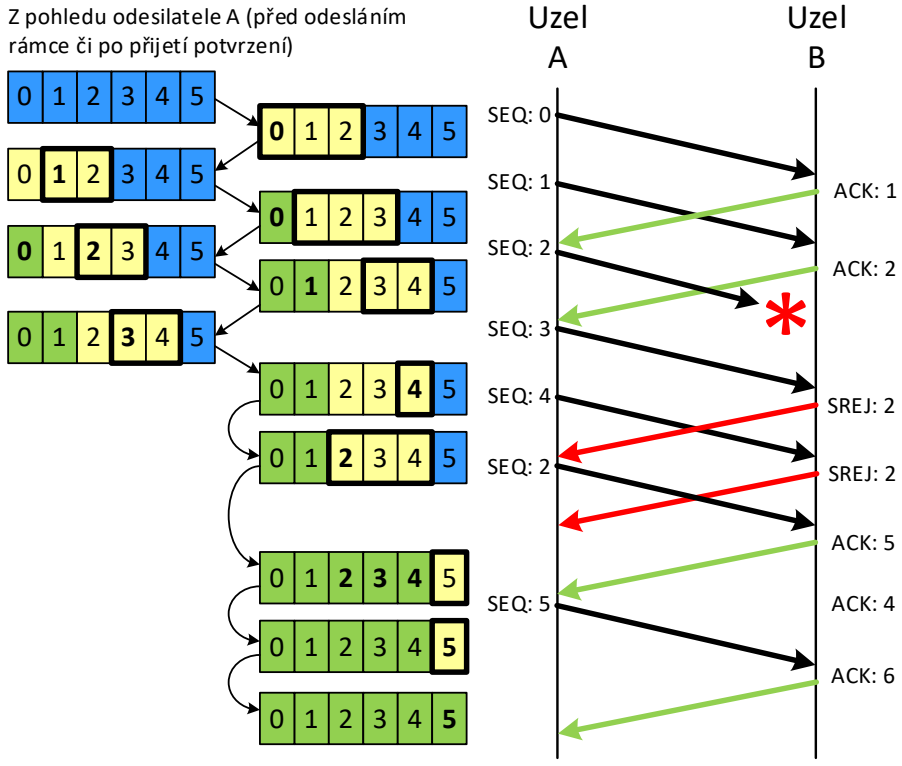
Nyní si ukážeme řešení obdobné situace jako na **Obr. 6-16** pro GBN, avšak pro SR (**Obr. 6-17**). Počet přenášených rámců, velikost okna i způsob potvrzování je v obou případech záměrně stejný. Jak už vyplynulo z popisu, u SR nemusí docházet k opakovanému přenosu rámců. Jakmile se vysílač dozví, že nebyl doručen rámec 2, vrátí se k jeho přenosu, a když mu to aktuální velikost okna či přijetí potvrzení dovolí, pokračuje rámcem 5.



Obr. 6-15: Ukázka fungování mechanizmu a) Go-back-N ARQ, b) Selective Repeat ARQ



Obr. 6-16: Ukázka fungování mechanizmu Go-back-N ARQ s konkrétním nastavením okna a dalších parametrů a s jednou chybou při přenosu



Obr. 6-17: Ukázka fungování mechanizmu Selective Repeat ARQ se stejným nastavením a jednou chybou při přenosu jako u příkladu na GBN

6.7.6 Technika klouzavého okna a řízení toku

Jak bylo popsáno v předcházejících kapitolách, technika klouzavého okna slouží primárně k zajištění **spolehlivého přenosu rámců** přes kanály s chybami. Ve skutečnosti však má tento mechanizmus ještě další dvě funkce.

První funkcí, kterou je možné z předcházejících kapitol také vyvodit, je zajištění správného **pořadí rámců** (*sequence number*). Jestliže má každý rámec sekvenční číslo, je velice jednoduché na straně příjemců ošetřit správné pořadí rámců při předávání vyšším vrstvám. Jak je patrné z kap. 6.7.4 a kap. 6.7.5, přístupy mohou být odlišné, avšak výsledek je vždy stejný. Když pomineme ztráty rámců, ke změně pořadí může docházet i při komunikaci bez chyby, pokud je topologie složitější a za běhu se změní. V těchto případech totiž může nastat situace, že jeden rámec bude zaslán jinou trasou než ostatní a kvůli různému zpoždění na trase příjemce obdrží rámce v jiném pořadí. Všechny tyto problémy lze snadno vyřešit již na spojové vrstvě, pokud jsou rámce číslovány.

Třetí role techniky klouzavého okna souvisí s problematikou **řízení toku** (*flow control*). Jeden z hlavních aspektů řízení toku spočívá ve vyřešení problému, jak má dát příjemce vědět vysílači, že nestačí rámce zpracovávat a že má vysílač zpomalit. Jinými slovy, je třeba zajistit, aby vysílač nezahltil příjemce (ten může např. přijímat rámce i od jiného vysílače a právě proto může být jeho vytížení vyšší). Velikost okna a potvrzující zprávy, které přijímač zasílá zpátky vysílači, umožňují přijímači ovlivňovat chování vysílače, poskytnout mu určitou zpětnou vazbu a provádět tak řízení toku. Potvrzující zpráva tedy neslouží pouze k ujištění vysílače, že přenos proběhl úspěšně, ale zároveň informuje o tom, že přijímač je schopen zpracovat další rámce. Detaily jsou nicméně nad rámec tohoto textu.

6.8 Logická versus fyzická topologie

Topologie sítě představuje určité uspořádání mezi uzly a také způsob jejich propojení. Síťové topologie mohou být vyhodnocovány z logické nebo fyzické úrovně.

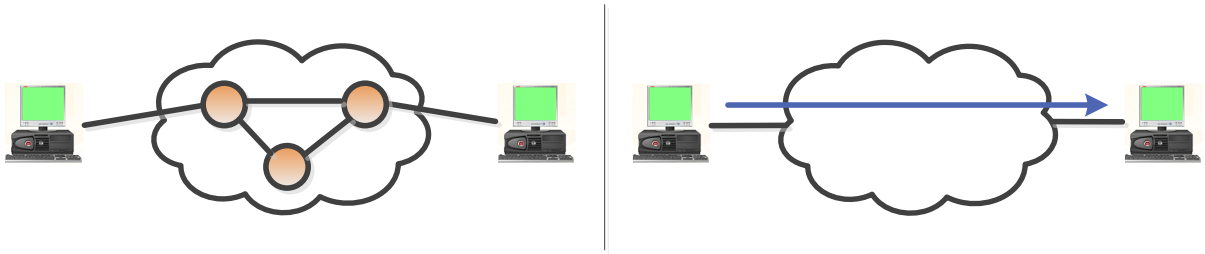
Fyzická topologie představuje skutečný způsob, jak jsou uzly propojeny kabely nebo kanály. Jedná se tedy o skutečnou topologii, kterou lze v některých případech zjistit např. zmapováním kabeláže. Fyzická topologie se tedy vždy skládá pouze z uzlů a spojů.

Logická topologie naproti tomu reprezentuje spíše způsob, jak síť funguje z hlediska přenosu rámců mezi jednotlivými uzly, než jak se síť jeví z pohledu uzlů a linek. Logická topologie nemusí s tou fyzickou souviset anebo může fyzické topologii odpovídat. Tato topologie je definována na úrovni spojové vrstvy a je dána především použitým protokolem. Přístup na médium a s ní související práce s rámci je pak závislá právě na logické topologii.

Nejběžnější logické (i fyzické) topologie jsou:

- topologie bod-bod
- hvězda
- strom
- topologie s vícenásobným přístupem
- kruh

Grafickou reprezentaci těchto topologií lze nalézt na **Obr. 4-2** a **Obr. 4-3**. Rozdíl mezi fyzickou a logickou topologií té stejné sítě demonstruje na příkladu topologie bod-bod následující **Obr. 6-18**.



a) b)

Obr. 6-18: Topologie sítě bod-bod z pohledu spojové vrstvy: a) skutečná fyzická topologie b) logická topologie

6.9 Přenosové technologie z pohledu řešení spojové vrstvy

V současnosti se můžeme potkat s mnoha typy přenosových technologií. U všech je pak nějakým způsobem řešena spojová vrstva. Cílem této podkapitoly je poskytnout základní přehled.

Technologie můžeme dělit mnoha způsoby, např. podle rozlohy (viz kap. 4.3), zde však použijeme rozdělení na:

- **pevné lokální sítě** – zde je nejběžnější Ethernet existující v několika variantách. Pevné (často metalické) prostředí je v mnoha ohledech výhodné a s tím také souvisí poměrně jednoduchý přístup ke způsobu řešení problémů spojové vrstvy popsaných v celé této kapitole. Typicky se zde jedná o propojení většího množství uzlů poměrně vysokou rychlostí.
- **bezdrátové lokální sítě** – sítě standardu 802.11 (Wi-Fi), Bluetooth. K nejrychlejšímu vývoji v současnosti dochází nejspíše právě v této oblasti. Bezdrátové prostředí z principu věci vyžaduje náročnější protokoly a vyšší úroveň řízení, jelikož otevřené prostředí má i četné nevýhody. I zde se setkáváme s propojením většího množství uzlů, typicky je však uzlů méně než u pevných technologií a rychlosti jsou o něco nižší.
- **WAN sítě bod-bod** – DSL (*Digital Subscriber Line*) technologie, E nebo T kanály (viz kap. 5.7.4), případně kanály SDH či SONET (taktéž viz kap. 5.7.4). Přestože se (zejména v případě xDSL technologií) jedná spíše o lokální přípojky, lze tímto způsobem vytvořit určitou formu WAN sítě. V těchto sítích jsou typicky využity vedení telefonních společností a nad ním je pak využit protokol PPP, jehož rámec byl popsán v kap. 6.4. Spojová vrstva může být poměrně jednoduchá. Jedná se tedy o propojení dvou uzlů, rychlosti jsou obvykle nižší než u lokálních sítí (v případě DSL, E a T linek), resp. srovnatelné (v případě SDH a SONET).
- **spojované WAN sítě** – sítě Frame Relay a ATM, které umožňují vytvářet různé druhy propojení mezi různými uzly sítě. Často zde dochází před vlastním přenosem k určité formě navázání spojení, tj. rezervaci přenosové kapacity sítě. V těchto sítích je tedy více uzlů než u sítí bod-bod, a zpravidla o něco méně než u lokálních sítí. Rychlosti jsou typicky o něco nižší než u WAN sítí typu bod-bod. Všechny tyto aspekty vedou k tomu, že spojová vrstva je o něco složitější. Do této skupiny lze s určitou mírou nepřesnosti zařadit i techniku MPLS (*Multiprotocol Label Switching*). Ta je běžně řazena na pomezí mezi druhou a třetí vrstvu a často je pro své výhodné vlastnosti

považována za nástupnickou technologii sítí Frame Relay a ATM. Popis fungování MPLS je však již nad rámec tohoto textu.

U některých technologií je obtížné jejich zařazení do čtyř skupin uvedených výše. Spíše do kategorie spojovaných WAN sítí spadají i sítě kabelových operátorů a FTTH, které byly zmíněny v kap. 3.3. Tyto technologie vyžadují složitější spojovou vrstvu z důvodu náročnějšího způsobu sdílení přenosových kapacit a také různorodosti přístupových metod. Obdobně pak mobilní sítě 3G/4G/LTE a satelitní sítě bychom řadili spíše do složitějších a navíc bezdrátových WAN sítí s částečně spojovaným charakterem komunikace a tím i složitější spojovou vrstvou. Avšak žádné z těchto zařazení není zcela přesné.

6.10 Zařízení spojové vrstvy

Přepínač (*switch*) představuje zařízení pracující primárně na spojové vrstvě, typicky s větším množstvím portů, umožňuje propojení většího množství zařízení. Nejčastěji se s ním potkáme v Ethernetových sítích. Jeho základní funkcí je na základě cílové MAC adresy v rámci rozhodnout na který port má být rámec odeslán, a na který ne. U běžných přenosů je rámec přepnut pouze na jeden konkrétní port, na kterém je připojen jeho adresát.

Toto zařízení již tedy disponuje určitou logikou, která mu umožňuje sledovat obsah rámců a podle toho se rozhodovat, jak s rámci naložit. Přepínač sleduje, na kterém portu se která stanice nachází (pomocí zdrojové MAC adresy v rámci), a právě na základě tohoto mechanizmu je později schopen rámec doručit konkrétnímu adresátovi.

Velkou výhodou přepínačů je, že při jejich výhradním použití přestávají existovat kolize, jelikož na každém segmentu vedení je pouze jedna stanice. To zrychluje a zefektivňuje komunikaci a umožňuje o něco vyšší zatížení sítě. Přepínač samozřejmě musí operovat i s fyzickou vrstvou, jelikož každý rámec musí být přijat či odeslán, tj. pracuje i s kódováním, modulacemi apod.

Výše uvedený popis se vztahuje k přepínači, označovaném často jako L2 (druhá vrstva). V moderních sítích se často potkáme také s takzvanými L3 přepínači. Na první pohled jsou to stejná zařízení jako L2 přepínače, avšak jak již vyplývá z názvu, jsou schopny pracovat i na třetí (síťové) vrstvě. Z tohoto důvodu o nich můžeme mluvit jako o směrovačích, které jsou však popsány až v kap. 7.15. Rozdíl mezi směrovačem a L3 přepínačem je pouze v implementaci. Základní funkce směrovače jsou zpravidla prováděny v softwarové části zařízení, zatímco u L3 přepínače v hardwarové části.

7 Síťová vrstva přenosových systémů

7.1 Přepojování paketů

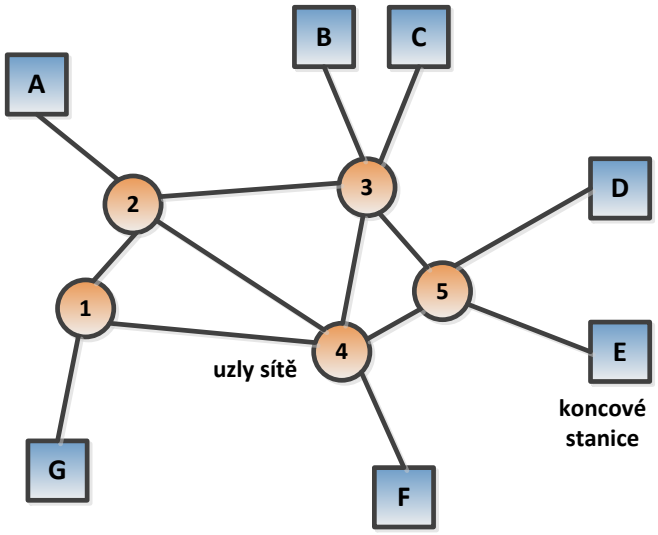
7.1.1 Principy přepojování paketů

Síťová vrstva musí být použita všude tam, kde spolu chtějí **komunikovat dva nesousedící účastníci spojení**, tj. neexistuje-li mezi nimi přímé spojení. V tomto případě je nutné mezi nimi najít vhodnou cestu jdoucí přes mezilehlé uzly od jednoho koncového uzlu ke druhému. Možných cest může být samozřejmě více, ale vybrána může být jedna, po které je poté zajištěno správné předání dat. V praxi to tedy znamená celou řadu rozhodnutí, které musí v síti proběhnout.

Existuje více druhů sítí a způsobů komutace, které byly popsány v kap. 3.2. Základní dva způsoby jsou **komutace okruhů a komutace paketů**. Pro další výklad budeme předpokládat sítě s přepojováním paketů, s kterými se můžeme v praxi setkat častěji. Tyto sítě se obvykle použijí tam, kde **není trvalá potřeba přenosu dat mezi zdrojem a cílem dat** (tj. po většinu času je kanál nečinný).

Typická horní **hranice pro délku paketů je 1000 až 1500 bajtů**. Jestliže jsou uživatelská data delší, musí být zpráva pro přenos rozdělena do většího množství paketů, o čemž bude pojednáno v kap. 7.8. Každý paket pak obsahuje své záhlaví (řídicí informaci paketu) a určitou část uživatelských dat. Na základě svého záhlaví je pak paket směrován sítí od uzlu k uzlu dle následujícího příkladu.

Uvažujme jednoduchou síť, kterou zachycuje **Obr. 7-1**. Nechť paket je vyslán ze stanice A do stanice E. Paket bude mít v řídicím poli adresu cílové stanice, tj. E. Vlastní paket je vyslán ze stanice A do uzlu 2. V tomto uzlu je uchován, je zde určen následující uzel (např. č. 3) a poté je paket předán do fronty paketů na lince mezi uzly 2-3. Když je spoj dostupný, je pak přenesen do dalšího uzlu, tj. 3. Odtud pak stejným způsobem do uzlu 5 a konečně do stanice E.



Obr. 7-1: K vysvětlení podstaty přepojování paketů v síti

7.1.2 Techniky přepojování paketů

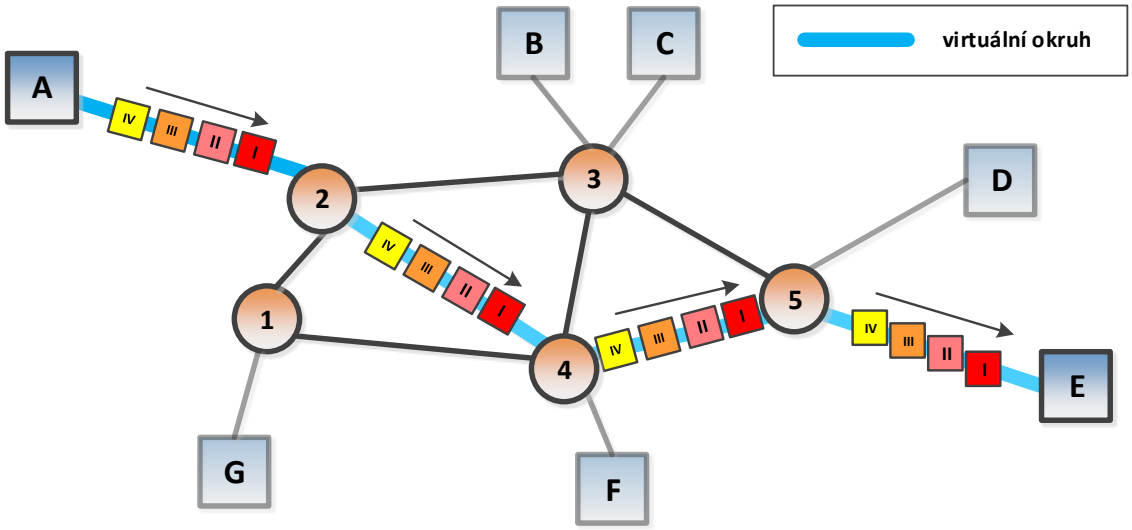
Síťové spojení poskytuje prostředky přenosu dat mezi transportními vrstvami. Existují dva základní způsoby přepojování paketů:

- **Služba se spojením** (*Connection-Oriented Network Services*), před přenosem se určitým způsobem navazuje spojení a během přenosu je pak zřejmé, odkud kam pakety putují a proto nemusí být u každé jednotky přímo informace o tom, komu je určena. Jednotky jsou zpravidla označeny určitým **identifikátorem toku** (*flow label*), který umožňuje identifikovat související pakety a zasílat je správným směrem. Tento způsob je na úrovni síťové vrstvy méně častý. Hovoříme o tzv. službě **virtuálních okruhů**, což znamená, že síťová vrstva poskytuje (resp. snaží se poskytovat) dokonalý bezchybný kanál dodržující pořadí datových jednotek při přenosu. Demonstrace fungování spojované služby je na **Obr. 7-2**.

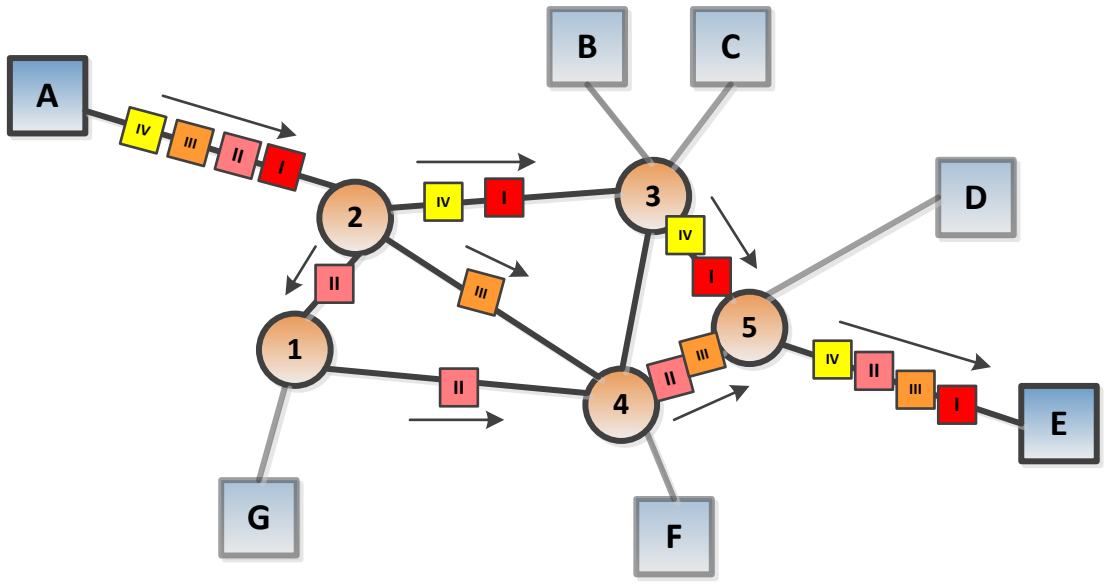
  Virtuální spojení mohou být:

  - **dočasný virtuální okruh** (SVC = *Switched Virtual Connection*), které je charakteristické třemi fázemi: přípravou, udržením a ukončením spojení, které se musí provádět vždy před výměnou dat a jsou nadefinovány pouze na dobu konkrétního přenosu.
  - **pevný virtuální okruh** (PVC = *Permanent Virtual Connection*) je takové, kdy spojení je sestaveno dlouhodobě, tj. musí být stabilně nadefinováno i v komunikačních uzlech (ve směrovacích tabulkách uzlů). Spojení se automaticky sestaví po zapnutí a je ve stavu přenosu dat po celou dobu spojení. Tento kanál nemůže být dále využit pro jiného uživatele.

- **služby bez spojení** (*ConnectionLess Network Services*), každý paket je *de facto* nezávislou jednotkou a musí být opatřen cílovou adresou, aby jej bylo možné doručit. S touto technikou se na síťové vrstvě setkáváme nejčastěji. Tato služba bývá nazývána též jako **datagramová**. U této metody může dojít při přenosu paketů k změně pořadí přijatých paketů u cílové stanice. Také se může stát, že některý paket není doručen, aniž by síťová vrstva příjemce byla informována. Demonstrace fungování nespojované služby je na **Obr. 7-3**.



Obr. 7-2: Ukázka přepojování paketů s vytvářením virtuálního okruhu (služba se spojením)



Obr. 7-3: Ukázka nespojovaného přepojování paketů v síti (služba bez spojení)

Následující tabulka poskytuje porovnání tří základních komunikačních technik, se kterými se můžeme setkat.

Tab. 2: Porovnání tří základních komunikačních technologií síťové vrstvy

Komutace okruhů	**Služby bez spojení (komutace paketů)**	**Služby s (virtuálním) spojením (komutace buněk)**
• Vyhrazená přenosová cesta • Průběžný přenos dat • Dostatečně rychlé pro interaktivní komunikaci • Zprávy nejsou uchovávány v síti • Cesta se sestavuje jednou pro celou délku spojení • Zpoždění při sestavování spojení, nepatrné přenosové zpoždění • Obsazovací signál, jestliže volaná stanice je obsazena • Přetížení sítě smí blokovat zřízení cesty, ale neomezuje již zřízená spojení • Uživatelská ochrana pro případy ztráty zprávy při přenosu • Pevná šířka přenosového pásma • Nevyžaduje záhlaví po sestavení spojení • Klasické telekomunikace (komutace okruhů)	• Není vyhrazena zvláštní přenosová cesta • Přenos dat v paketech • Dostatečně rychlé pro interaktivní komunikaci • Pakety smí být uchovány do jejich předání příjemci • Směrovací procedury jsou prováděny pro každý paket zvlášť • Zpoždění přenosu paketu • Odesílatel smí být informován o tom, že paket nebyl předán • Přetížení zvyšuje zpoždění paketů v síti • Síť je odpovědná za jednotlivé pakety • Dynamické přidělování šířky pásma (možnost priorit) • Každý paket musí obsahovat záhlaví s adresou cíle • Komutace paketů (většina současných datových sítí)	• Není vyhrazena zvláštní přenosová cesta • Přenos dat v paketech • Dostatečně rychlé pro interaktivní komunikaci • Pakety jsou uchovány do jejich předání příjemci • Směrování se provádí jednou pro celé spojení • Zpoždění při sestavování spojení, zpoždění při přenosu každého paketu • Odesílatel je informován, jestliže spojení je odmítnuto • Přetížení smí blokovat sestavení spojení, zvyšuje zpoždění paketu v síti • Síť je zodpovědná za posloupnost přenášených paketů • Dynamické přidělování šířky pásma • Každý paket musí obsahovat záhlaví s adresou cíle

7.1.3 Vliv velikosti paketů na přepojování

Důležitou vlastností paketové sítě je velikost paketů. Z hlediska zpoždění přenosu dat sítí se jeví výhodné volit velikost paketů spíše menší, neboť tyto pakety je možno v komunikačním uzlu rychle zkontrolovat a odeslat je do dalšího uzlu. Záhlaví má zpravidla fixní délku. Změnou poměru záhlaví paketu k délce užitečných dat se snižuje reálná propustnost kanálu mezi komunikačními uzly.

V paketové síti existují **tři základní druhy zpoždění**:

- **zpoždění dané šířením signálu** – má nezanedbatelný dopad zejména při komunikaci na velké vzdálenosti, např. přes satelitní spoje,
- **doba vysílání** – tj. doba nutná k odeslání paketu z uzlu, tj. doba, která uplyne mezi zahájením vysílání prvního bitu a vysíláním posledního bitu paketu,
- **zpoždění v uzlu** – tj. doba, která je nutná pro zpracování paketu v uzlu a jeho předání k vysílání.

Z popisu jednotlivých zpoždění (a také z kap. 3.6, kde bylo také o zpoždění pojednáno) je zřejmé, že velikost paketu má na zpoždění komunikace podstatný vliv.

7.2 Služby síťové vrstvy

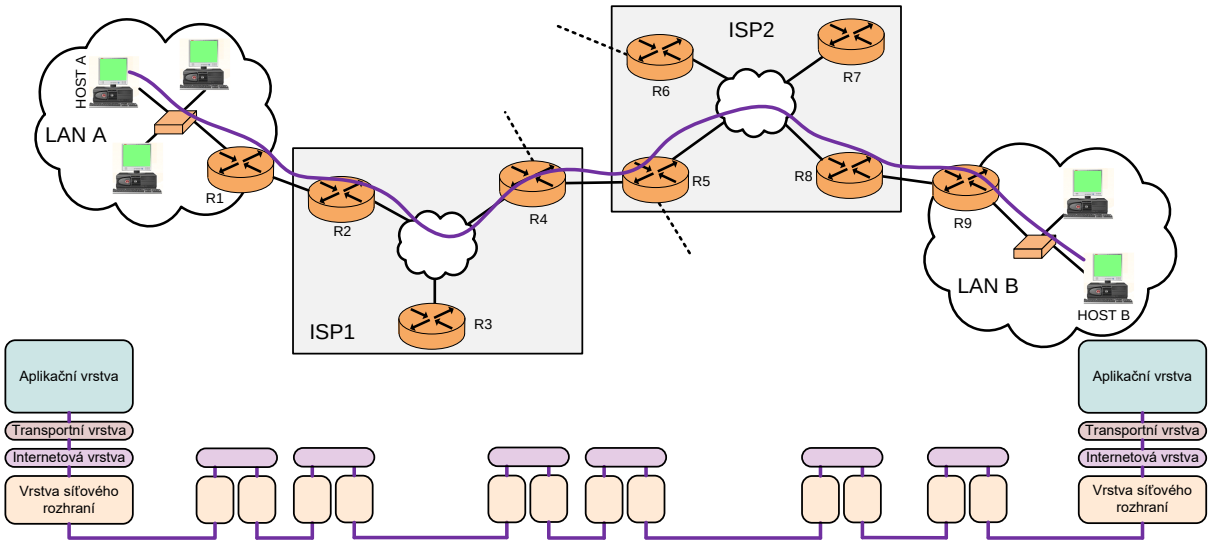
7.2.1 Úvod do služeb síťové vrstvy

K základnímu popisu služeb síťové vrstvy využijeme **Obr. 7-4**. Na tomto obrázku je znázorněna část Internetové sítě a dvě lokální sítě (LAN A, LAN B), jejichž uživatelé (Host A, Host B) si chtějí spolu vyměňovat data. Představme si situaci, že Host A chce odeslat malé množství dat Hostu B. V rámci jeho stanice dojde k průchodu aplikační transportní a Internetovou vrstvou, čímž bude vytvořen paket, na vrstvě síťového rozhraní pak rámec, který bude odeslán na výchozí bránu dané sítě (R1). Jednotky pak dále prochází přes síť poskytovatele ISP1 a směrovače R2 a R4, následně přechází do sítě poskytovatele ISP2, přes směrovače R5 a R8, až se dostává na hranici lokální sítě, kde se nachází druhý host (B), přičemž musí projít přes hraniční směrovač R9.

V dolní části obrázku je vyznačeno, jaké vrstvy jsou v rámci kterého uzlu zapojeny do přenosu paketů, a také je vyznačen průchod těmito vrstvami. Je zřejmé, že v koncových uzlech jsou zapojeny všechny vrstvy, zatímco v mezilehlých pak pouze dvě (z pohledu TCP/IP), resp. tři (z pohledu ISO/OSI, kde rozlišujeme fyzickou a spojovou vrstvu)[18].

Při pohledu na tento obrázek je možné si odvodit, jaké služby bude síťová vrstva poskytovat v jednotlivých částech přenosu. V následujících kapitolách budou tyto služby popsány.

[18] Ve skutečnosti bude situace s největší pravděpodobností ještě o něco složitější, z hlediska rozdílnosti přístupových technologií jednotlivých lokálních sítí. Např. LAN A by mohla být připojena pomocí kabelového operátora a LAN B pomocí DSL. To vnáší do přenosové trasy ještě na tzv. první míli trasy (připojení koncových sítí) další prvky, modemy, které se starají o přenos po trasách využívaných u těchto technologií. Z hlediska IP provozu a síťové vrstvy jsou to však do určité míry transparentní technologie, které proto nejsou v obrázku znázorněny.



Obr. 7-4: Znázornění průchodu paketu částí Internetové sítě

7.2.2 Účel výchozí brány

V rámci dané sítě nebo podsítě (viz kap. 7.5.7) mohou stanice komunikovat přímo, teoreticky i bez použití síťové a vyšší vrstvy. Nicméně pokud nastane situace, kdy stanice potřebuje komunikovat se zařízením na jiné síti, jako tomu bylo např. na **Obr. 7-4**, potřebuje k tomu zprostředkovatele (směrovač), často označován jako brána nebo také **výchozí brána**. Tento spojovací prvek slouží právě na propojování jednotlivých sítí.

Výhoda konceptu výchozí brány je v tom, že existence tohoto zařízení usnadňuje roli koncovým uzlům, které se nemusí zabývat dostupností jiných sítí. Stanici tak stačí znát adresu výchozí brány a ta se postará o další doručení jednotek.

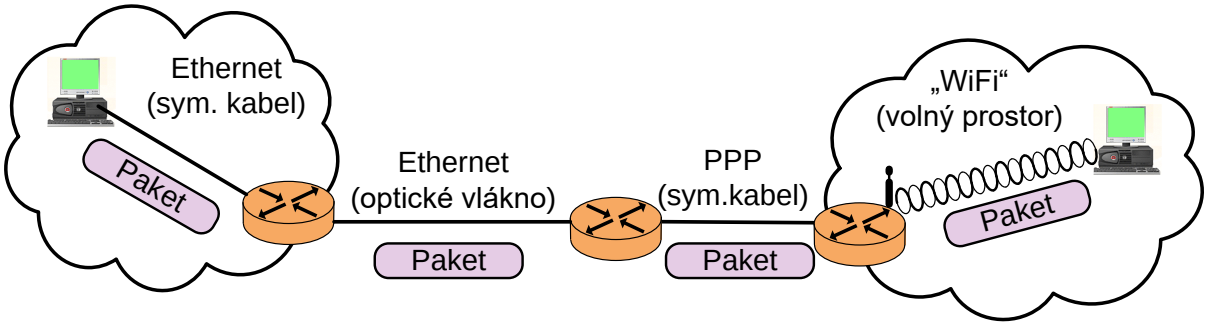
Komunikace následně funguje tak, že každý směrovač potřebuje typicky znát adresu následujícího skoku a tímto způsobem se data postupně dostanou od zdroje k cíli, tak jak je ukázáno např. na **Obr. 7-4**.

7.2.3 Nezávislost síťové vrstvy na přenosové technologii

Síťová vrstva je do určité míry nezávislá na konkrétní přenosové technologii. Fungování protokolů síťové vrstvy (nejčastěji IPv4 a IPv6) je téměř vždy prakticky stejné, nehledě na to, po jaké fyzické trase přenos probíhá. Konkrétní jednotka síťové vrstvy (paket) může být v nezměněné podobě přenášena např. po symetrickém kabelu, optickém kabelu nebo volným prostorem. Datagram je samozřejmě vždy zapouzdřen za pomoci rámce dané spojové vrstvy, s ohledem na konkrétní přenosové prostředí. Situace je ilustrována na **Obr. 7-5**.

7.2.4 Logické adresování

Aby síťová vrstva mohla poskytovat přenos mezi koncovými stanicemi, potřebuje univerzální identifikační prostředek jednotlivých uzlů. Tímto prostředkem jsou logické adresy, které jsou nejčastěji označovány jako síťové adresy či IP adresy. Tyto adresy slouží ke globální identifikaci daného uzlu a jsou přidělovány z určitého rozsahu. Více se těmto adresám budeme věnovat v kap. 7.5.



Obr. 7-5: Demonstrace nezávislosti síťové vrstvy na přenosové technologii

7.2.5 Základní služby síťové vrstvy poskytované z pohledu zdrojové stanice

U zdrojové stanice jsou síťovou vrstvou prováděny celkem čtyři služby. Jsou to:

- **vytváření paketů** – zapouzdření jednotky vyšší vrstvy do datagramu, tj. přidání záhlaví s odpovídajícími údaji, což jsou především logické adresy zdroje a cíle, informace o fragmentaci a další. Obsah záhlaví bude více diskutován v kap. 7.7.
- **vyhledávání logické adresy dalšího uzlu (skoku) směrem k cíli** – datagram obsahuje adresu zdroje a cíle, nicméně jak je patrné z **Obr. 7-4**, paket prochází i přes mezilehlé sítě a je proto nutné dohledat další skok trasy a jeho logickou adresu. K tomuto účelu se využívá směrovací tabulka a proces se nazývá směrování, více viz kap. 7.6.
- **vyhledání linkové adresy tohoto uzlu** – doručení paketů do dalšího uzlu není úlohou síťové vrstvy, ale vrstvy spojové. Spojová vrstva však k doručení potřebuje znát spojovou adresu dalšího skoku a tuto adresu zjišťuje vrstva síťová, na základě znalosti logické adresy. Více je této problematice věnováno v kap. 7.10.
- **rozdělení datagramu na menší jednotky** (pokud je nezbytné) – síťová vrstva musí sledovat, jaká je na dané síti, kterou má být paket odeslán, maximální možná velikost datagramu. Pokud je vytvořený paket větší, je nutné jej rozdělit (fragmentovat) na nezbytně nutný počet částí. Každý z fragmentů musí obsahovat záhlaví (viz první bod) a jednotlivé fragmenty se liší pouze informacemi o fragmentaci. Více viz kap. 7.8.

K zajištění těchto služeb jsou využívány informace, které si síťová vrstva opatří vlastními prostředky a také informace, které síťová vrstva obdrží od vyšší vrstvy. Jsou to především samotná data (a informace o jejich délce), logická adresa cílové stanice, identifikace protokolu použitého na vyšší vrstvě a typ požadované služby.

7.2.6 Základní služby síťové vrstvy poskytované na každém směrovači

Síťová vrstva na každém mezilehlém uzlu musí spolupracovat s dvěma spojovými vrstvami, jak je patrné i z **Obr. 7-4**. To je proto, že směrovač vždy pracuje minimálně s dvěma rozhraními – příchozím a odchozím. V rámci směrovače již nedochází k vytváření paketu (první služba u zdrojové stanice), ostatní kroky jsou však v principu stejné.

Přijatý paket je nejdříve zkontrolován (z hlediska chyb při přenosu) a poté jsou prováděny další kroky (již bez dalšího popisu). Služby síťové vrstvy mezilehlých prvků tedy zahrnují:

- **kontrola bezchybnosti přenosu paketu,**
- **vyhledávání logické adresy dalšího uzlu (skoku) směrem k cíli,**
- **vyhledání linkové adresy tohoto uzlu,**
- **rozdělení datagramu na menší jednotky** (pokud je tato služba používaným síťovým protokolem v mezilehlých uzlech povolena).

7.2.7 Základní služby síťové vrstvy poskytované z pohledu cílové stanice

V cílové stanici je již role síťové vrstvy relativně jednoduchá. Každý datagram případně fragment je nutné zkontrolovat z hlediska bezchybnosti přenosu. Jakmile dorazí všechny fragmenty původního paketu, je tento paket znovu složen a teprve poté předán vyšší vrstvě. Pokud k fragmentaci nedošlo, jsou data z paketu předávána vyšší vrstvě přímo.

Služby u cílové stanice tedy zahrnují:

- **kontrola bezchybnosti přenosu paketu,**
- **seskládání datagramu z jeho fragmentů** (pokud byl paket fragmentován).

7.2.8 Další důležité služby síťové vrstvy

Na síťové vrstvě se můžeme potkat i s dalšími službami, které mohou nebo musí být zabezpečeny např. přídavnými protokoly, některé nemusí být implementovány vůbec anebo souvisí více se službami na vyšších vrstvách. Jsou to:

- **řízení chybových stavů** (*error control*), v popisu služeb jednotlivých bodů přenosu v předcházejících kapitolách byla již určitá operace související s chybami popsána. Oprava chyb při přenosu však představuje pouze základní mechanizmus, který není přímo řízením chybových stavů. Řízení chybových stavů a jeho aspekty byly již popsány na spojové vrstvě, která může tuto službu poskytovat (viz kap. 6.7.1). Nicméně je otázkou, zda by se těmito problémy neměla zabývat i síťová vrstva. Běžně složitější mechanizmy řízení chybových stavů na síťové vrstvě nenalezneme, a pokud je daná aplikace vyžaduje, jsou řešeny až na vyšší vrstvě. Nicméně standardně síťová vrstva obsahuje samostatný protokol, který poskytuje částečné služby řízení chybových stavů. Je to protokol ICMP (*Internet Control Message Protocol*), či ICMPv6, kterým je věnována pozornost v kap. 7.13.
- **řízení toku dat** (*flow control*), které se snaží o to, aby přijímací strana nebyla zahlcena pakety od vysílací strany v takovém množství, že je nebude schopna zpracovávat. Běžně síťová vrstva tuto problematiku přímo v současné době neřeší a tento problém spadá u koncové problematiky spíše do vyšší vrstvy.
- **řízení provozu sítě v případě zahlcení** (*congestion control*), které je významné v případě, kdy je v síti nebo její oblasti přítomno příliš vysoké množství paketů. V takovém případě mohou směrovače začít zahazovat vybrané pakety, u nichž není kapacita pro jejich přenos. To však nemusí situaci v síti zlepšit, pokud se vyšší mechanizmy budou pokoušet ztracené pakety přenášet opakovaně. Mechanizmy řízení zahlcení se liší podle toho, zda je přenos v síti provozován se spojením nebo bez spojení.

  V případě sítí bez spojení je nutné nějakým způsobem informovat odesílatele paketů, že má zpomalit. K tomu lze použít určitou formu signalizace, s kterou se však v běžných

IP sítích nyní nepotkáváme. V praxi je zde využíván opět protokol ICMP, který umožňuje v případě zahlcení odeslat tzv. škrtící paket (*choke packet*), na který by měl zahlcující vysílač reagovat zpomalením přenosu. Odlišný způsob řešení pak spočívá v rozlišování paketů z hlediska jejich důležitosti pomocí určité návěsti přímo v záhlaví paketu. Na základě této návěsti lze pak v případě zahlcení zahazovat pakety s nižší důležitostí. U některých typů přenosu může být toto zahazování aplikací tolerováno.

V případě sítí se spojením je situace o něco snazší. Pokud si přijímací a vysílací strana dohodnou vhodné parametry, přenos by měl probíhat bez zahlcení.

- **kvalita služeb** (*quality of service* = *QoS*) spočívá především ve vyřešení problému, jak zabezpečit rychlou a dostatečně kvalitní výměnu dat u aplikací, které ji vyžadují (hovory, videokonference, obecně systémy přenosu v reálném čase). Tato problematika však běžně do úkolů síťové vrstvy nespadá a bývá řešena na vyšší vrstvě.
- **směrování** (*routing*), mechanizmus umožňující směrovačům dynamicky zjišťovat informace o vzdálených sítích, do kterých jsou pak následně směrovány pakety. K tomu jsou zpravidla využívány speciální směrovací protokoly, které jsou řazeny do síťové nebo někdy i vyšší vrstvy. Směrovacím protokolům je věnována kap. 7.6.
- **bezpečnost** (*security*), síťová vrstva byla původně navržena bez jakéhokoliv zabezpečení, což je ze současného pohledu velký problém. S mechanizmy zabezpečení přenosu se můžeme potkat i na vyšších vrstvách, na síťové vrstvě mluvíme nejčastěji o tzv. IPsec (*Internet Protocol Security*). Problematika bezpečnosti je nad rámec tohoto textu, nicméně zmínka o IPsec je uvedena v související kapitole o tunelování, viz. 7.9.

7.2.9 Služby síťové vrstvy poskytované transportní vrstvě

Služby, které byly popsány v předcházejících kapitolách lze rozdělit do kategorie služeb v rámci síťové vrstvy (viz kap. 7.2.10) a služeb poskytovaných transportní vrstvě.

Zejména následující služby poskytuje nebo může poskytovat síťová vrstva vrstvě nadřazené (transportní):

- **přenos datových jednotek** – je prováděn z pohledu transportní vrstvy transparentně.
- **výběr kvality služeb** – pokud je implementováno, vrstva určuje kvalitu služby tím, že definuje parametry, jako jsou chybovost, dostupnost služby, spolehlivost, propustnost, přenosové zpoždění či zpoždění zřízením spojení.
- **výběr typu síťového spojení** – pokud existuje více variant, které se mohou lišit svým charakterem např. z pohledu služby se spojením nebo bez spojení.
- **oznamování chyb** – neopravených síťovou a nižšími vrstvami.
- **dodržení pořadí datových jednotek** – sledování pořadí paketů a případně přeuspořádání před předáním transportní vrstvě.
- **řízení toku dat** – dle pokynů transportní vrstvy může být řešena úprava množství nebo rychlosti přenosu datagramů.

7.2.10 Služby uvnitř síťové vrstvy

Tento typ služeb je zacílen dovnitř vrstvy, aby umožnil splnění funkcí, které jsou vyšší vrstvou očekávány. Jsou to, případně mohou to být, zejména tyto:

- **směrování** – přepojování mezi různými sítěmi.
- **realizace síťového spojení** – pomocí protokolů na spojové vrstvě, může např. docházet k multiplexování různého počtu síťových spojení do jednoho spojení 2. vrstvy.
- **fragmentace a defragmentace** – rozdělování a znovu seskládání jednotek z důvodů přílišné velikosti.
- **detekce chyb** – kontrola kvality síťového spojení.
- **zotavení se z chyb** – např. mechanizmy opakovaných přenosů na této úrovni, pokud je implementováno.
- **řízení síťové vrstvy** – předávání chybových a řídicích zpráv mezi entitami síťové vrstvy, typicky pomocí protokolu ICMP nebo i směrovacích protokolů, např.:
  - test dosažitelnosti cíle (uzlu),
  - informace o nedoručitelnosti datagramu,
  - žádost o zpomalení vysílání datagramů,
  - zpráva o zničení datagramu z důvodů vypršení doby života (počtu bran), kterými má datagram projít,
  - detekce nesprávného záhlaví datagramu,
  - žádost o opravu směrovací tabulky – informace o změnách v propojení sítě,

7.3 Úloha síťové vrstvy s IP protokolem

IP protokol je v současné době hlavním protokolem síťové vrstvy. Tato vrstva v síťovém modelu TCP/IP zajišťuje potřebné směrování mezi jednotlivými dílčími sítěmi, a vyšším vrstvám tak vytváří iluzi jediné homogenní sítě. Sama však musí pracovat se skutečnou vnitřní strukturou, resp. způsobem vzájemného propojení. Síťová vrstva se musí vyrovnávat i s konkrétními odlišnostmi jednotlivých dílčích sítí, technologií a topologií – například s odlišným charakterem linkových adres, s různou maximální velikostí přenášených paketů, resp. rámců a jejich formátem, s odlišným charakterem poskytovaných přenosových služeb (spojovaných či nespojovaných) apod.

Ovladač na úrovni vrstvy síťového rozhraní (fyzická a spojová vrstva) dokáže odstínit síťovou vrstvu od konkrétního způsobu ovládání příslušné sítě a přesného formátu datových rámců. Není ovšem již v jeho silách zastřít před síťovou vrstvou rozdíl v používaném mechanizmu linkového adresování, resp. zajistit používání jednotných fyzických adres ve všech dílčích sítích. Tento jednotný způsob adresování může zajistit až **síťová vrstva**, která **provádí** tzv. **jednotnou abstrakci**, tedy sjednocení v:

- **způsob adresování (IP adresy),**

  Adresy, které protokol IP zavádí (IP adresy), jsou 32bitové (u protokolu IP verze 4). Z pohledu transportní a aplikační vrstvy je lze interpretovat jako lineární (resp. jednosložkové) adresy – což odpovídá představě jediné homogenní sítě. Jednoduše řečeno, každý uzel má svoji konkrétní adresu. Na úrovni síťové vrstvy se ale interpretují jako dvousložkové, tvořené číslem, resp. adresou hostitelského počítače a číslem, resp. adresou (dílčí) sítě, ve které se tento hostitelský počítač nachází.

  IP adresy jsou pouze abstraktními (logickými) adresami, které musí být posléze převedeny na skutečné fyzické adresy, protože ovladač na úrovni vrstvy síťového

rozhraní, pokud dostane nějaká data k odeslání, musí spolu s nimi dostat i konkrétní fyzickou adresu, na kterou je má odeslat. Na nižších vrstvách se totiž s IP adresami již nepracuje. Jde-li například o lokální síť typu Ethernet, dostane síťová vrstva (od vrstvy transportní) adresu cílového hostitelského počítače ve formě 32bitové IP adresy, ale příslušný ovladač vrstvy síťového rozhraní musí získat 48-bitovou Ethernetovou adresu (MAC). Mechanizmus, jakým se v TCP/IP sítích provádí zjištění fyzické adresy na základě IP adresy, bude podrobněji popsán v kap. 7.10.

- **formát datových paketů používaných na síťové vrstvě (IP datagramy),**

  Podobně jako jednotný formát adres a způsob adresování, zavádí protokol IP i jednotný formát přenášených dat na úrovni síťové vrstvy – tzv. IP datagramy. Jde o datové pakety, označované jako datagramy proto, že jsou přenášeny pomocí nespojované (datagramové) síťové přenosové služby. Na úrovni vrstvy síťového rozhraní jsou tyto datagramy přenášeny pomocí takových rámců, se kterými příslušná dílčí síť pracuje. Rámce se tak mohou síť od sítě lišit.

- **nespolehlivá nespojovaná přenosová služba z pohledu vyšších vrstev** (transportní a aplikační).

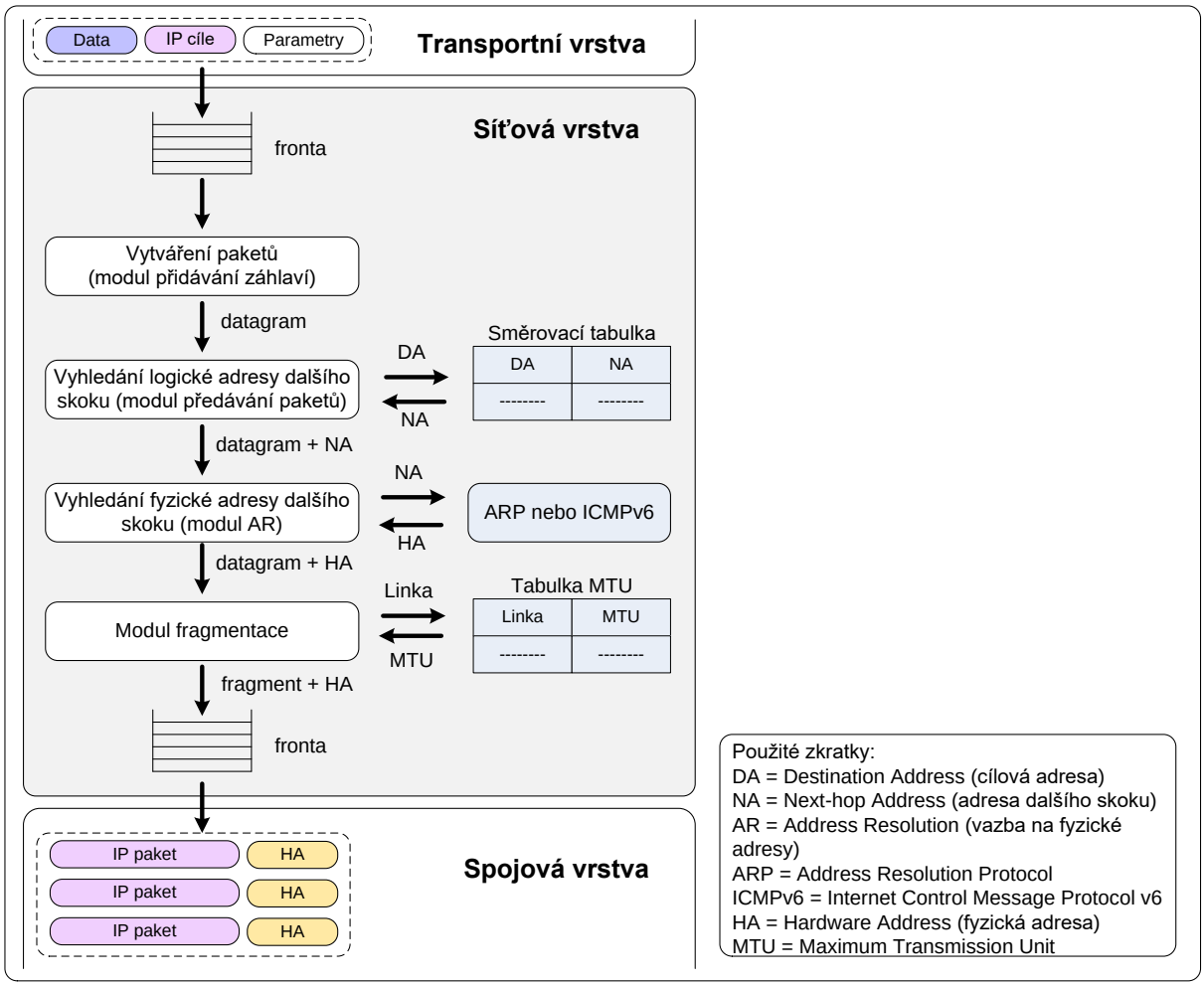
  Síťová vrstva se nepokouší vytvářet na svojí úrovni službu se spojením, nezávisle na tom, jaká služba je poskytována na úrovni spojové. Lze konstatovat, že síťová vrstva poskytuje z pohledu přenosu co nejjednodušší službu, kterou je následně schopna poskytovat libovolná spojové technologie. Pokud daná aplikace funkce spojovaného nebo spolehlivého přenosu vyžaduje, musí být tyto mechanizmy implementovány ve vyšší vrstvě.

7.4 Struktura síťové vrstvy s IP protokolem

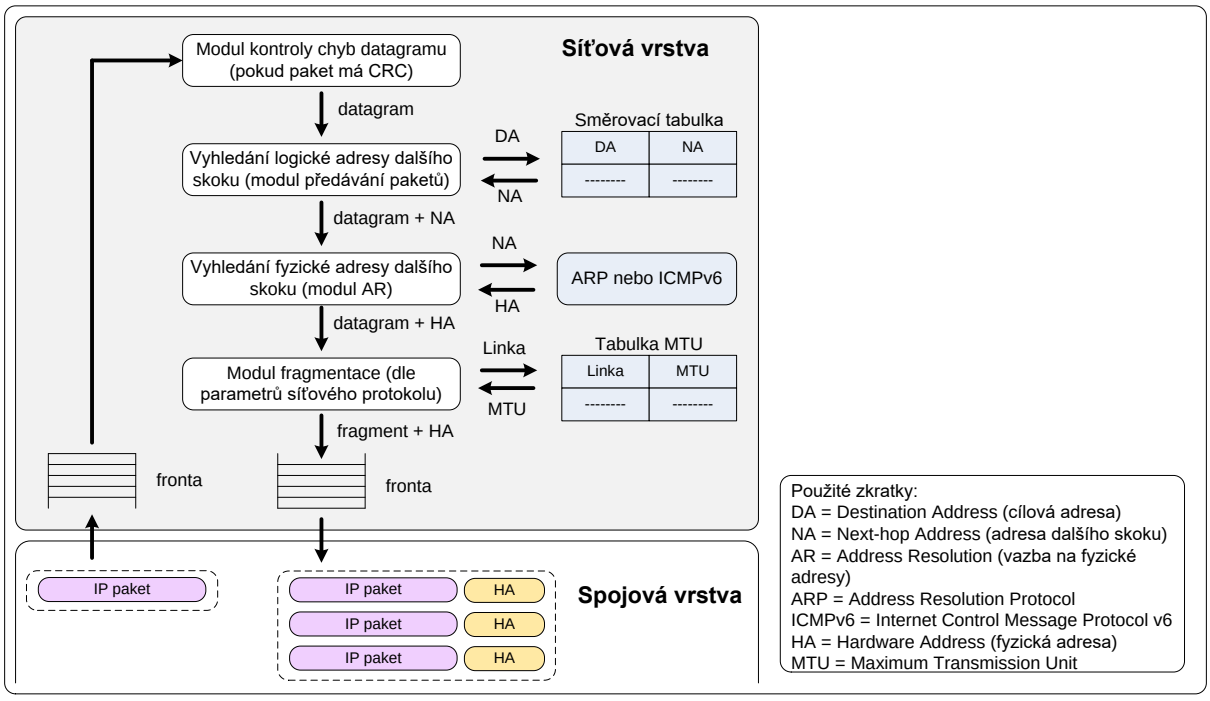
Funkce a služby síťové vrstvy byly popsány v předcházejících kapitolách. Na **Obr. 7-6** je ukázána struktura síťové vrstvy v návaznosti na dvě nejbližší sousední vrstvy a také funkce, které vrstva zajišťuje. V tomto obrázku je ukázána pouze průchodová cesta směrem od vyšší vrstvy k nižší, tj. situace, kdy probíhá příprava datové jednotky k odeslání u zdrojové stanice (vysílací strana).

U mezilehlého uzlu (směrovače), bude struktura síťové vrstvy velice podobná, pouze se již pakety nebudou vytvářet, ale přeposílat. Situaci zachycuje **Obr. 7-7**.

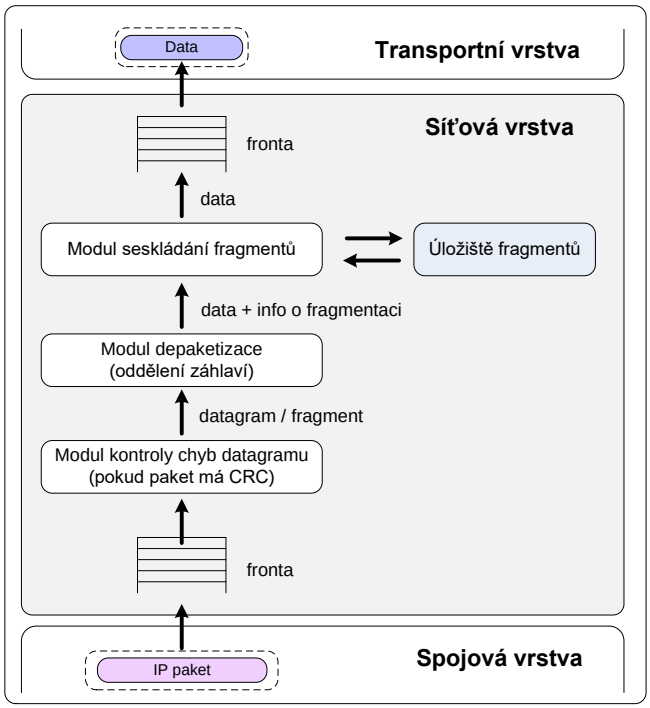
Poslední částí řetězce je struktura síťové vrstvy u příjemce dat, která je znázorněna na **Obr. 7-8**.



Obr. 7-6: Struktura síťové vrstvy s IP protokolem z pohledu odesílání paketů (zdroj)



Obr. 7-7: Struktura síťové vrstvy s IP protokolem u mezilehlého uzlu - směrovače



Obr. 7-8: Struktura síťové vrstvy s protokolem IP u příjemce paketu

7.5 Adresy síťové vrstvy u IPv4 protokolu

7.5.1 Úvod do adresování v IPv4

V sadě TCP/IP má každé zařízení v libovolné síti svoji vlastní logickou adresu, která bývá nejčastěji označována jako IP adresa[19]. Tato adresa je unikátní a univerzální a vztahuje se vždy ke konkrétnímu rozhraní daného zařízení. Dvě zařízení v rámci Internetové sítě nesmí mít nikdy stejnou IP adresu[20]. Platí, že pokud má dané zařízení více rozhraní, bude mít i vyšší počet adres.

Délka adresy je měřena v bitech popř. bajtech a je u protokolu IPv4 rovna 32 bitům, což představuje 4 bajty. IP adresu má každé zařízení, které pracuje alespoň na úrovni síťové vrstvy, což jsou zejména všechny koncové stanice a směrovače. Počet všech možných adres bývá běžně označován jako tzv. adresní prostor (*address space*). Lze snadno odvodit, že tento rozsah by měl být 2^{32} = 4 294 967 296 ≈ 4 miliardy adres. Toto číslo definuje teoretický počet zařízení, resp. rozhraní, které mohou být přímo připojeny k Internetu. Nicméně jak uvidíme dále, počet reálně použitelných adres je ve skutečnosti nižší, zejména kvůli různým omezením a také speciálním typům adres.

7.5.2 Přidělování adres

Internet byl vždy vyvíjen formou otevřené spolupráce, nicméně existují organizace, které se od počátku starají o různé části s Internetem související problematiky. Z hlediska

[19] Pokud je použita zkratka IP, nejčastěji je v praxi myšlen protokol IP verze 4.

[20] Z tohoto pravidla existují výjimky, z nichž jednou je tzv. privátní adresování, o kterém bude řeč později, a druhou pak technika *anycast*, jejíž bližší popis je však nad rámec tohoto textu.

přidělování adres je zásadní organizace **IANA** (*Internet Assigned Numbers Authority*)[21]. Základní informace o ní lze shrnout do těchto bodů:

- podčást větší organizace ICANN (*Internet Corporation for Assigned Names and Numbers*); její technický správce, přidělování a správa různých veličin,
- spravuje systém DNS, administrace tzv. DNS root zóny (.), o kterých bude řeč v kapitole o aplikačních protokolech; provozuje domény .int a .arpa,
- **správa a přidělování IP (v4 a v6) adres**,
- správa a přidělování čísel autonomních systémů (větších Internetových sítí),
- správa registru protokolů ve spolupráci s IETF (*Internet Engineering Task Force*),
- Zastoupení v jednotlivých regionech – **RIR** (*Regional Internet Registry*)
  - AFRINIC (*African Network Information Center*) – Afrika,
  - APNIC (*Asia Pacific Network Information Centre*) – Asie a Pacifik,
  - ARIN (*American Registry for Internet Numbers*) – Severní Amerika,
  - LACNIC (*Latin American and Caribbean Internet Addresses Registry*) – Latinská Amerika,
  - **RIPE NCC** (*Réseaux IP Européens Network Coordination Centre*) – Evropa a Blízký východ.

V uvedených regionech pak působí organizace označované jako **LIR** (*Local Internet Registry*), které již komunikují přímo s koncovými zákazníky, a prostřednictvím nich lze získat adresní prostor jednoho z IP protokolů. Seznam LIR působících v ČR je poměrně dlouhý[22], jelikož jako LIR jsou zaregistrováni prakticky všichni poskytovatelé Internetu.

7.5.3 Zápis IP adres

Počítače pracují zejména s binární reprezentací dat. 32bitové IP adresy si lze tedy představit jako celá kladná čísla zapsané v dvojkové soustavě. Pro člověka tento zápis ale není příliš srozumitelný, a tak se pro symbolický zápis IP adres zavedla konvence, označovaná jako tečkovaná desítková notace (*dotted decimal notation*). Spočívá v tom, že 32 bitů IP adresy se rozdělí na čtyři části po osmi bitech (oktety), a každá část se pak vyjádří jako celé desítkové číslo bez znaménka (s použitím tečky jako oddělovače jednotlivých částí). Z matematického hlediska lze říci, že je využita číselná soustava o základu 256, jelikož osmi bitové číslo může nabývat hodnot od 0 po 255.

Uveďme si příklad, např. nepříliš snadno zapamatovatelný binární tvar IP adresy je:

10010011 11100101 10010111 00000001

dostává v tečkované desítkové notaci lepší podobu:

147.229.151.1

Pro zápis IPv4 je možné využít i hexadecimální číselnou soustavu, nicméně není to příliš běžné. S touto číselnou soustavou se pak potkáváme až u IPv6 adres.

[21] Bližší informace lze nalézt na http://www.iana.org/

[22] Aktuální seznam lze nalézt na stránkách RIPE NCC, konkrétně http://www.ripe.net/membership/indices/CZ.html.

7.5.4 Maska sítě

Jak bylo uvedeno v kap. 7.3, IP adresy jsou dvousložkové, tj. tvořené číslem resp. adresou (dílčí) sítě, ve které se hostitelský počítač nachází, a číslem, resp. adresou tohoto hostitelského počítače. Nejmenší jednotkou je u IP adresy bit a proto platí, že určité bity z celé adresy jsou vyhrazeny pro adresu sítě a následně zbývající bity pro adresu stanice. V praxi však ze samotné IP adresy nelze běžně poznat, které bity jsou vyhrazeny pro který účel, a proto potřebujeme další parametr, který nám pomůže rozlišovat jednotlivé části. Vždy platí, že bity pro adresu sítě tvoří vždy nepřerušenou řadu z leva, na kterou navazuje souvislá řada bitů pro adresy stanic. Není tedy možné využívat bity na přeskáčku apod.

Jestliže všechny bity, které jsou vyhrazeny pro adresu sítě, budou nahrazeny binární „1“ a ostatní bity pak nastaveny jako „0“, získáme masku sítě. Pokud převedeme takto vzniklé číslo na zápis používaný u IP adres, dostaneme opět čitelnější formát. Z předcházejícího popisu a i následujícího příkladu je zřejmé, že maska má stejnou délku jako IP adresa, tj. 32 bitů:

Konkrétní IP adresa (10) – 147.229.151.1

Dělení bitů na adresu sítě a adresu stanice zde mějte např. 1:1, tj. na poloviny:

10010011 11100101 | 10010111 00000001

Maska sítě (2) – 11111111 11111111 00000000 00000000

Maska sítě (10) – 255.255.0.0

Maska sítě se taktéž často zapisuje tzv. **délkou prefixu**, která vyjadřuje počet jedniček v binárním zápisu masky a píše se s lomítkem za adresu IP. Prefix představuje začátek adresy, který vyjadřuje adresu sítě.

Totožné zápisy IP adresy a masky, resp. délky prefixu tedy jsou:

147.229.151.1 255.255.0.0

147.229.151.1 / 16

V některých případech je využívána tzv. *wildcard* maska. Ta představuje převrácenou hodnotu síťové masky, tj. každý bit je invertován (binární funkce NOT). Uveďme si příklad:

Maska sítě (10) – 255.255.0.0

Maska sítě (2) – 11111111 11111111 00000000 00000000

Wildcard maska (2) – 00000000 00000000 11111111 11111111

Wildcard maska (10) – 0.0.255.255

7.5.5 Rozsah adres, adresa sítě a všesměrová adresa

V praxi se často pracuje nejen s konkrétními IP adresami, ale i určitým rozsahem těchto adres, např. pro použití v konkrétní síti. Každý rozsah má vždy první adresu, poslední adresu a množství adres mezi nimi, které je dané velikostí rozsahu v mocninách dvou.

První adresa rozsahu je vždy označována jako tzv. **adresa sítě** (*network address* či spíše *subnet address*). Tato adresa reprezentuje daný rozsah a nemůže být přiřazena konkrétní stanici. Tyto adresy jsou běžně využívány pro směrování, více viz. kap. 7.6.

Poslední adresa rozsahu je pak označována jako tzv. **všesměrová adresa** (*broadcast address*). Pakety odeslané na tuto adresu budou doručeny všem stanicím na dané síti, tj. na všechny adresy v rámci daného rozsahu.

Všechny adresy mezi první a poslední adresou je možné využít pro běžné adresování stanic v konkrétní síti. Počet těchto adres je dán především počtem bitů, které jsou v IP adrese vyhrazeny pro adresování stanic. Jestliže využijeme výše uvedený příklad, kde bylo k dispozici celkem 16 bitů pro adresy stanic, počet všech adres pro stanice lze vypočítat jako $(2^{16} - 2) = 65\,534$. Je zřejmé, že volba počtu bitů pro adresy stanic následně limituje maximální možný počet stanic v síti s unikátní adresou.

Adresu sítě je možné vypočítat na základě znalosti libovolné IP adresy z daného rozsahu a masky sítě. Výpočet se provádí binárně po bitech pomocí funkce AND. Následuje příklad:

Zadané hodnoty:

libovolná IP adresa rozsahu (10) – 147.229.230.55

Maska sítě (10) – 255.255.0.0

Binární zápis:

IP (2) – 10010011 11100101 11100110 00110111

Maska sítě (2) – 11111111 11111111 00000000 00000000

Výsledek (AND po jednotlivých bitech)

Adresa sítě (2) – 10010011 11100101 00000000 00000000

Adresa sítě (10) – 147.229.0.0

Všesměrovou adresu lze taktéž vypočítat na základě znalosti libovolné adresy daného rozsahu a *wildcard* masky sítě. Získáme ji tak, že veškeré bity, sloužící pro adresaci stanic, nastavíme na „1“. V binární aritmetice se jedná o provedení funkce OR po jednotlivých bitech. Uveďme si opět příklad:

Zadané hodnoty:

libovolná IP adresa rozsahu (10) 147.229.230.55

Wildcard maska (10) – 0.0.255.255

Binární zápis:

IP (2) – 10010011 11100101 11100110 00110111

Wildcard (2) – 00000000 00000000 11111111 11111111

Výsledek (OR po jednotlivých bitech)

Všesměrová adresa (2) – 10010011 11100101 11111111 11111111

Všesměrová adresa (10) – 147.229.255.255

7.5.6 Třídy IPv4 adres

V době, kdy vznikl koncept IP adres, bylo navrženo členění celého adresního prostoru na tzv. třídy. Tomuto typu adresování se říká třídní (*classful*). Přibližně v polovině 90. let se

nicméně tento koncept ukázal jako problematický a proto bylo přistoupeno k tzv. beztřídnímu adresování (*classless*), které upravuje původní striktní členění adresního prostoru na poněkud volnější. Nicméně znalost třídního adresování je stále velmi důležitá, jelikož je velmi zakořeněno v různých protokolech a je také východiskem pro beztřídní adresování.

Podle prvních bitů adresy a také počtu bitů využívaných pro adresu sítě (a tedy i počtu bitů zbývajících pro uvažované stanice v rámci sítě) dělíme IP adresy na třídy, viz **Tab. 3**. Původní filozofie TCP/IP počítá s malými (C), středními (B) a velkými sítěmi (A). Toto rozdělení se později ukázalo jako příliš hrubé a neefektivní.

Tab. 3: Historické dělení adresního prostoru IP protokolu na třídy

Třída	Rozsah prvního oktetu adresy (dekadicky)	Dělení adresy na adresu Sítě a Hosta	Standardní maska sítě (dekadicky)	Délka prefixu sítě	Počet možných sítí / hostů na jednu síť
A	0 – 127	S.H.H.H	255.0.0.0	/8	128 / 16 777 214
B	128 – 191	S.S.H.H	255.255.0.0	/16	16 383 / 65 534
C	192 – 223	S.S.S.H	255.255.255.0	/24	2 097 150 / 254
D	224 – 239	-	Multicastové adresy		
E	240 – 255	-	Experimentální adresy		

Z tabulky je možné odvodit, že třída A zabírá celkem 50 % adresního rozsahu, třída B pak 25 %, třída C pouze 12,5 % a následně třída D a E shodně každá jen 6,25 %. V tabulce je uvedena i maska sítě u každé třídy. Nicméně v době třídního adresování nebyla maska sítě vůbec potřeba, jelikož bylo podle prvního bajtu adresy automaticky jasné, kolik bitů je využito pro adresu sítě a kolik pro adresu stanice. To však dnes již neplatí.

Vzhledem k adresnímu prostoru představuje Internet síť sítí. Každá dílčí síť využívá určitý adresní rozsah, který spadá do určité třídy (A, B nebo C) a který jí byl přidělen Internetovými organizacemi. V prvotních dobách toto přidělování probíhalo skutečně dle tříd, tj. bylo přiděleno několik opravdu velkých rozsahů (z třídy A), nějaké střední rozsahy třídy B a i určité množství malých sítí kategorie C. Jak uvidíme dále, v průběhu let se přešlo na přidělování rozsahů prakticky nezávislých na původních třídách. Hovoříme zde často o tzv. podsíťování.

7.5.7 Podsíťování

Vzhledem k neefektivnímu rozdělení adres, které bylo popsáno v kap. 7.5.6, spočívajícím především v nedostatku adres stanic ve třídě C a naopak prakticky nevyčerpatelnému množství adres uzlů ve třídě A, se přistoupilo k mechanizmu tzv. podsíťování (*subnetting*).

Třídní IP Adresa je původně dvojsložková, tj. tvořena adresou sítě a adresou stanice. V případě podsíťování se adresa stává trojsložkovou, tj. tvořenou adresou sítě, **adresou podsítě** a adresou stanice (viz **Obr. 7-9**). Adresa podsítě je vytvořena rozdělením části původně určené k adresaci stanice na dvě části: adresu podsítě a adresu stanice. Původní adresa sítě zůstává nezměněna.

***n* bitů**	***m* bitů**	**(32 – *n* – *m*) bitů**
adresa sítě	**adresa podsítě**	**adresy stanic**

Obr. 7-9: Grafické znázornění trojsložkové adresy při podsíťování

Podsíťování přináší dvě základní výhody. První je, že organizace, která má již nějaký adresní rozsah k dispozici může tento blok rozdělit na menší části a vytvořit tak několik podsítí. Tím se usnadní správa (rozdělení na menší jednotky) a je možné také lépe pracovat například s bezpečnostními pravidly sítě. Druhou výhodou je, že při dalším přidělování adres jiným organizacím bylo možné používat rozsahy různé velikosti (více než tří tříd), a to tak, že se více zohledňovaly potřeby konkrétní sítě. Hlavní nevýhody jsou taktéž dvě. První je, že podsíťování přináší nutnost využívat masku sítě (v souvislosti s podsíťováním nazývána jako maska podsítě), bez které není možné určit hranice daného rozsahu. Druhou nevýhodou je pak zvýšení počtu sítí a s tím související zvýšení počtu záznamů ve směrovacích tabulkách směrovačů, což vede ke zpomalení funkce směrování (viz kap. 7.6).

Podstatou podsíťování je, že v rámci jednoho adresního bloku vytvoříme několik podjednotek – podsítí. Pro podsíťové adresy se dodržuje pravidlo platné pro adresu sítě, kde se využívá souvislé řady bitů zleva od (nedotknutelné) adresy sítě. Jakmile se používá podsíťování, je třeba přesně rozlišovat mezi adresou sítě, adresou její podsítě a adresou stanice v rámci dané podsítě. Maska podsítě má stejný formát jako maska sítě (viz kap. 7.5.4), tedy 32bitové číslo, obsahující „1“ v souvislé řadě zleva na pozicích označujících adresu sítě a podsítě. Zbývající „0“ označují pozice bitů pro adresu hosta. Jak je patrné z **Tab. 4**, při podsíťování nelze použít zcela libovolný počet bitů pro adresaci podsítí; musí zůstat zachována možnost adresovat alespoň nějaké stanice v rámci vytvořené podsítě. Proto se nepoužívají poslední dva bity, aby zbyly 4 volné adresy v rámci podsítě, tj. adresa podsítě, dvě adresy pro stanice a broadcastová adresa[23].

Tab. 4: Parametry podsíťování původních tříd IP adres

Třída dělené sítě	Délka prefixu dělené sítě	Pořadí bitů použitelných původně pro adresaci stanice	Pořadí bitů použitelných pro adresu podsítě	Možná délka prefixu podsítě	Celkem bitů použitelných pro podsítě	Maximální možný počet podsítí v rámci jedné původní sítě
A	/8	9. – 32.	9. – 30.	/9 – /30	22	2^{22}
B	/16	17. – 32.	17. – 30.	/17 – /30	14	2^{14}
C	/24	25. – 32.	25. – 30.	/25 – /30	6	2^{6}

Z pohledu současné praxe je třeba vzít v potaz, že pojem adresa sítě a adresa podsítě do určité míry splývají, vzhledem k velkému rozšíření podsíťování. Tj. jestliže se hovoří o adrese sítě či o adrese podsítě, zpravidla je myšleno to stejné, což může být matoucí. Z hlediska pojmů, tak jak byly v tomto textu definovány, se oběma těmito pojmy v praxi často myslí rozsah bitů použitých jak pro adresu sítě, tak i adresu podsítě dohromady.

[23] Toto pravidlo má taktéž výjimky, avšak ty jsou nad rámec tohoto textu. Zájemce odkazujeme na dokument RFC 3021.

Uveďme si příklad na práci s podsítěmi:

Mějme danou síť 193.1.1.0 / 24. Potřebujeme tuto síť rozdělit na 3 části – podsítě.

a) Kolik bitů z části pro hosty musíme využít?

b) Kolik může být v každé podsíti stanic?

c) Jaká je podsíťová maska, resp. délka prefixu podsítě?

d) Jaké podsítě budou vytvořeny?

Řešení

a) 3 podsítě – potřebujeme 2 bity, protože $2^2 = 4$ je nejbližší vyšší mocnina dvou, (jedna podsíť tedy zůstane nyní nevyužita).

b) Pro stanice jsme původně měli 8 bitů, nyní jsme 2 odebrali pro podsítě, takže nám zůstane 6 bitů. Hostů může tedy v každé z podsítí být $2^6 = 64$, resp. o dva méně, tedy 62, jelikož první adresa je vždy adresa (pod)sítě a poslední je vždy všesměrovou adresou.

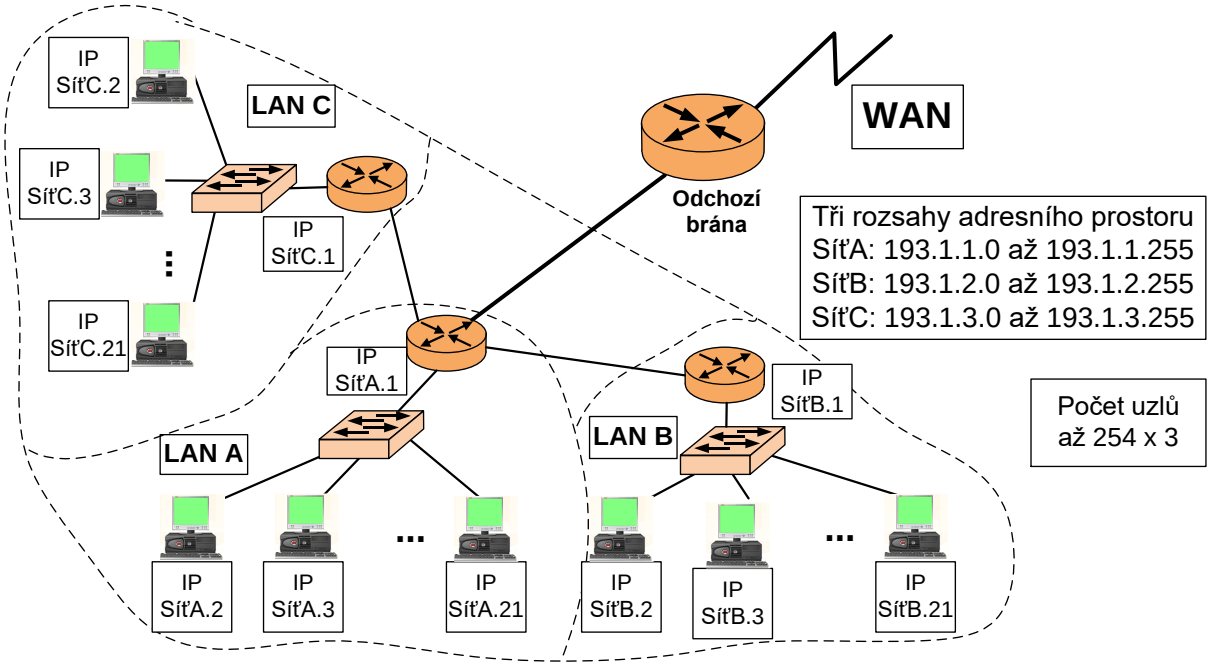
c) Původní maska (24 binárních „1“) se o dvě „1“ rozšíří, tj. o bity použité pro tvorbu podsítí, délka prefixu tedy bude /26 a maska podsítě v dekadickém zápise pak 255.255.255.192

d) Přehled podsítí je uveden v následující tabulce.

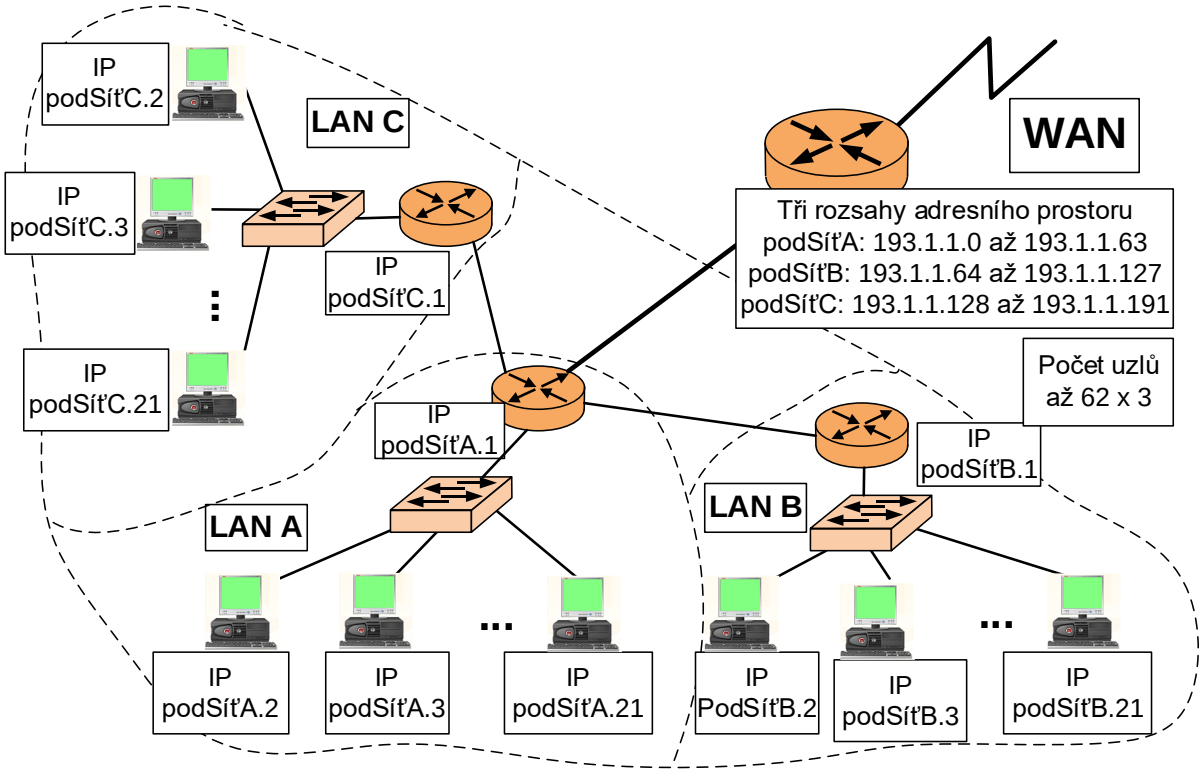
Tab. 5: Přehled vytvořených podsítí k příkladu na podsíťování

Číslo podsítě	Adresa podsítě (první adresa rozsahu)	Maska dané podsítě	Rozsah adres použitelných pro stanice v podsíti	Počet možných stanic v podsíti	Všesměrová adresa podsítě
0	193.1.1.0	255.255.255.192	193.1.1.1 – 193.1.1.62	62	193.1.1.63
1	193.1.1.64	255.255.255.192	193.1.1.65 – 193.1.1.126	62	193.1.1.127
2	193.1.1.128	255.255.255.192	193.1.1.129 – 193.1.1.190	62	193.1.1.191
3	193.1.1.192	255.255.255.192	193.1.1.193 – 193.1.1.254	62	193.1.1.255

Situaci řešenou v předcházejícím příkladu můžeme chápat ještě z mírně odlišného úhlu pohledu. Mějme ve výchozí situaci 3 samostatné LAN sítě (např. jednotlivá patra budovy), kde v každé z nich je přibližně 20 stanic. Pro tyto stanice potřebujeme IP adresy, abychom mohli provozovat aplikace založené na TCP/IP. Kdyby neexistovala technika podsíťování, bylo by nezbytné využít tři samostatné rozsahy třídy C, např. 193.1.1.0/24, 193.1.2.0/24 a 193.1.3.0/24. V každé z těchto sítí pak bude přibližně 230 nevyužitých IP adres, což není příliš efektivní. Situaci znázorňuje **Obr. 7-10**. Nevýhodou tohoto stavu řešení jsou pak i trojnásobné náklady, jelikož získání bloků IP adres není zpravidla zadarmo.



Obr. 7-10: Tři sítě adresované s použitím třídního adresování



Obr. 7-11: Tři sítě adresované s využitím beztřídního adresování (technika podsíťování)

Z předcházejícího popisu je zřejmé, že celkem potřebujeme přibližně 70 IP adres (včetně IP adres směrovačů, adres sítí a všesměrových adres). Toto množství lze s velkou rezervou umístit pouze do jedné sítě třídy C. Při podsíťování nám tak postačuje pouze jedna

síť C, např. 193.1.1.0/24. Tu pak rozdělíme na čtyři podsítě, jak bylo popsáno v předcházejícím příkladu. V každé podsíti máme možnost mít až 62 stanic, takže nám zůstala poměrně velká rezerva pro budoucí rozšíření počtu stanic. Stejně tak nám zůstala v rezervě jedna celá podsíť[24]. Situace při využití techniky podsíťování je naznačena na **Obr. 7-11**.

7.5.8 Pojmy major network, supernet a beztřídní adresování

V současné době se v praxi potkáme s tzv. **beztřídním adresováním** (*classless addressing*), u kterého se nepracuje s třísložkovou adresou, ale pouze dvousložkovou. Adresa sítě a adresa podsítě splývají v jeden pojem, přičemž souhrnný název neexistuje. Spíše se běžně používají oba pojmy, a to ve stejném významu. Samozřejmostí je pak potřeba znalosti masky, pokud chceme rozeznat, jaký má daná síť rozsah adres. Příslušnost k původní třídě již nemá přímý vliv na délku prefixu, nicméně znalost původního třídního konceptu adresování je nadále nevyhnutelná. Mimo jiné s ním souvisí i pojmy *major network* a *supernet*, o kterých je pojednáno dále. V rámci **Obr. 7-12** je uveden ještě pojem suffix, který reprezentuje adresu stanic. Je zřejmé, že v rámci konkrétní sítě je vždy stejný prefix a následně pak u každé stanice různý suffix.

***n* bitů**	**(32 – *n*) bitů**
prefix (síť)	**suffix (stanice)**

Obr. 7-12: Grafické znázornění dvousložkové beztřídní adresy

Major network představuje původní třídní adresu sítě, do které daná podsíť spadá. Pokud sloučíme více sítí typu *major network*, tj. pokud vytvoříme síť větší, než jaká odpovídá původnímu třídnímu členění, vytvoříme tzv. **supernet**. Toho lze typicky využít u sítí z původní třídy C, kde je každá ze sítí velmi malá, nicméně *supernet* lze definovat i u dalších tříd. Pokud sloučíme více sítí, vytvoříme větší blok s větším počtem adres.

Uveďme si příklad na stanovení *major network*:

Mějme konkrétní síť

149.10.10.0 / 24

Její maska je

255.255.255.0

Výše uvedená síť patří do původní třídy B, proto je její major network:

149.10.0.0 / 16

Další příklad je na vytváření *supernet*u:

Pokud bychom měli k dispozici celý výše uvedený *major network*, a i následující *major network*

149.11.0.0 / 16

Bylo by možné vytvořit *supernet*

[24] Ta by se však v reálném případě rozdělila na menší podsítě a využila pro adresování sítí spojujících směrovače, to je však již nad rámec tohoto textu.

149.10.0.0 / **15**

Je zřejmé, že kapacita takto vytvořené sítě je rovna součtu kapacit slučovaných sítí.

Obdobně pak lze postupovat i v případech kdy slučujeme více sítí. Vytváření větších adresních rozsahů nejen nad rámec původních třídních *major network* souvisí s tzv. sumarizací případně agregaci adres, o které bude pojednáno v kap. 7.6.

7.5.9 Speciální typy IPv4 adres

Lokální smyčka *(loopback)* je softwarovou smyčkou uvnitř počítače, použity jsou libovolné adresy z velkého rozsahu 127.0.0.0 / 8. Pakety s touto adresou nikdy neopustí počítač. To je vhodné např. pro meziprocesovou komunikaci, či lokální testování sady TCP/IP.

Z celého rozsahu IP adres jsou vyčleněny rozsahy tzv. **privátních adres** (*private addresses*), které byly navrženy pro adresování v sítích nepřipojených k Internetu. Dnes se však z důvodu nedostatku IPv4 adres používají pro lokální sítě s vlastním mechanizmem adresace, které mají navenek např. pouze jednu *veřejnou adresu* (funkce NAT = *Network Address Translation* – překlad privátních adres na veřejné, kap. 7.11). Privátní adresy tak musí být unikátní pouze v rámci jedné lokální sítě, ale celosvětově se můžou libovolně opakovat, jelikož se vždy skrývají za veřejnou adresou hraničního směrovače své lokální sítě. O tyto adresy tedy není nutné žádat. V souvislosti s NATem je často diskutovaná vlastnost, že NAT zvyšuje bezpečnost vnitřních počítačů, protože jejich adresy nejsou z vnější sítě snadno dostupné.

Rozsahy vyčleněné pro privátní adresy jsou

- Třída A 10.0.0.0 – 10.255.255.255 (1 síť o 16 777 214 možných hostech)
  tj. síť 10.0.0.0 / 8
- Třída B 172.16.0.0 – 172.31.255.255 (16 sítí, každá o 65 534 hostech)
  tj. sítě 172.16.0.0 / 16 až 172.31.0.0 / 16
- Třída C 192.168.0.0 – 192.168.255.255 (256 sítí, každá o 254 hostech)
  tj. sítě 192.168.0.0 / 24 až 192.168.255.0 / 24

S těmito rozsahy je možné pracovat úplně stejně, jako se standardními (veřejnými) IP adresami, lze provádět podsíťování, či vytváření super sítí.

Lokální linkové adresy (*link-local addresses*) mají vyhrazen rozsah 169.254.0.0 až 169.254.255.255. V případech, kdy se stanici nepodaří nakonfigurovat IP parametry automaticky, např. v případě selhaní automatické konfigurace (DHCP server, viz kap. o aplikační vrstvě), sama si po určité době nastaví IP z uvedeného rozsahu. Pokud tak učiní na stejné síti více stanic, mohou být tyto adresy využity pro jejich lokální komunikaci. Pravděpodobnost kolize, tj. stavu, že si dvě stanice na síti přiřadí stejnou IP, je velmi nízká, nicméně i pokud by k této situaci došlo, umí toto stanice vyřešit. Problematika je popsána jako součást kap. 7.10.

Celý blok 0.0.0.0 / 8 je vyhrazen pro **lokální identifikaci stanic**, adresa 0.0.0.0 / 32 je pak vyhrazena pro vlastní identifikaci stanice ve speciálních případech jako je např. komunikace s DHCP serverem před přidělením IP adresy. V tomto případě ještě stanice nemá žádnou IP adresu, nicméně potřebuje odeslat paket a v něm je nutné vyplnit adresu odesílatele (viz kap. 7.7).

Další speciální adresou je 255.255.255.255 / 32, která slouží jako **lokální všesměrová adresa**. Pokud je paket odeslán na tuto adresu, není směrovači předáván do dalších sítí, nicméně v rámci lokální sítě by měl být doručen na všechny stanice. Tato adresa je v praxi taktéž využívána při komunikaci s DHCP serverem a to jako adresa cílová, jelikož stanice nemůže znát na počátku konkrétní adresu serveru.

V adresním rozsahu IPv4 existuje několik dalších bloků, které jsou vyhrazeny pro speciální účely. Např. adresní prostor 192.0.2.0 / 24 je vyhrazen pro příklady v textech, síť se nazývá **TEST-NET-1** a představuje neexistující doménu example.com. Tyto adresy nejsou ve skutečnosti Internetem směrovány, nicméně ani použití těchto adres v příkladech není až tak časté. Dále se můžeme setkat s adresami vyhrazenými pro určité typy tunelů, např. 6to4, které patří do rozsahu 192.88.99.0 / 24 a jsou vyhrazeny na přechodové mechanizmy IPv6 protokolu, více viz kap. 7.14.4.

Celkově je všech speciálních bloků více než 15 a zabírají přibližně 14 % celého adresního rozsahu[25] (počítán je i blok pro multicast a budoucí použití, které jsou uvedeny v **Tab. 3**).

Adresní prostor a jeho alokace (rozčlenění) se v čase mění. Aktuální a platný stav lze nalézt na stránkách organizace IANA[26].

7.5.10 Způsoby a důvody rozdělování stanic do samostatných sítí

Způsoby členění stanic do samostatných sítí jsou dány na základě následujících faktorů:

- **geografie** – tento způsob spočívá v tom, že stanice, které jsou geograficky v jedné lokalitě (městě, budově, patru, případně místnosti), jsou sdruženy do jedné sítě a stanice v jiné lokalitě pak do další.
- **účel** – stanice jsou rozděleny podle účelu a jejich primárních potřeb. Typicky se pak jedná o rozdělení sítě dle jednotlivých oddělení společnosti, nezávisle na geografickém uspořádání.
- **vlastnictví** – stanice jsou rozděleny do sítí podle svého vlastnictví. Základní stanice dané organizace využívané lokálně tvoří jednu síť, zařízení určená pro vzdáleně připojené uživatele pak druhou síť a třetí síť pak např. hostující zařízení (typicky v bezdrátové síti).

Na adresování a členění na sítě se můžeme podívat i z jiného pohledu než doposud. Představme si, že nepropojujeme již existující sítě, ale vytváříme nové. Pokud pomineme problémy s množstvím adres, můžeme všechny stanice dát do jedné velké sítě. Nicméně to je z několika důvodů nevýhodné.

[25] Detaily je možné zjistit v RFC 5735, viz http://tools.ietf.org/html/rfc5735 .
[26] Konkrétně na adrese http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml .

Důvody proč stanice rozdělit do většího počtu samostatných sítí mohou být

- **výkonnostní** – jestliže je síť příliš velká, může docházet k zahlcení nebo přetížení centrálních síťových prvků, případně přenosových tras. Jestliže nějakým způsobem stanice rozdělíme do samostatných sítí, zpravidla zvýšíme celkový počet síťových prvků a přenosových tras, čímž celkově zvýšíme přenosovou kapacitu systému.
- **bezpečnostní** – jestliže stanice rozdělíme do skupin, typicky dle vlastnictví nebo účelu, můžeme mezi těmito sítěmi snáze definovat bezpečnostní pravidla, co je povoleno a co je zakázáno.
- **adresní** – jestliže budeme mít stanice rozděleny do menších sítí, bude role těchto stanic snazší. Jelikož se sousedy na síti komunikují stanice přímo, bude výhodné, že přímých sousedů je méně a bude tak třeba pracovat s menším počtem adres.

7.6 Techniky směrování

Úkolem síťové vrstvy je poskytnout transparentní přenos dat z transportní vrstvy jednoho do transportní vrstvy druhého uživatele. Síťová vrstva tak musí najít cestu mezi systémy, jež komunikují přes jeden nebo více mezilehlých uzlů, ve kterých jsou prováděny funkce **směrování** (*routing*).

Z hlediska **popisu směrování** se potkáváme se dvěma pojmy:

- **doručování paketů** (*delivery*) – způsob zacházení s pakety v sítích řízených síťovou vrstvou. Přímé doručování paketů je možné v případě, kdy zdrojová a cílová stanice jsou na stejné síti.
- **předávání paketů** (*forwarding*) – způsob, jak je paket doručen následující stanici v řetězci od odesílatele k příjemci, dalšímu skoku přenosové trasy. Tato funkce je zpravidla považována za směrování jako takové.

Kdykoliv předá transportní vrstva nějaká data vrstvě síťové, přidá k nim pouze informaci o tom, kdo má být konečným příjemcem dat. Síťová vrstva tak jednoznačně identifikuje adresáta komunikace pomocí síťové adresy. Nicméně síťová vrstva pak musí skutečně rozhodnout, kterým směrem data reálně odeslat. Jakmile toto síťová vrstva provede, předá paket příslušné spojové vrstvě spolu s údajem o zvoleném směru.

Pro účely směrování vyžaduje síťová vrstva určité **informace o topologii sítě a adresách uzlů.**

Konkrétních způsobů směrování existuje celá řada: od jednoduchých až po adaptabilní, které se umí přizpůsobit provozu v síti (zatížení, výpadkům spojů nebo uzlů v síti, apod.).

Mechanismus směrování paketů je závislý zejména na topologii sítě z hlediska redundance přenosových linek. Dvěma extrémy z hlediska topologie sítě jsou sítě stromové a sítě tvořící úplný polygon. Ve stromových sítích existuje mezi každou dvojicí uzlů pouze jedna cesta, směrování je proto jednoznačné. V úplném polygonu pak existuje přímé spojení mezi každou dvojicí uzlů. Většina sítí má topologii obecného (neúplného) polygonu, kde alespoň pro některé dvojice uzlů existuje více alternativních cest., resp. neexistuje přímé spojení.

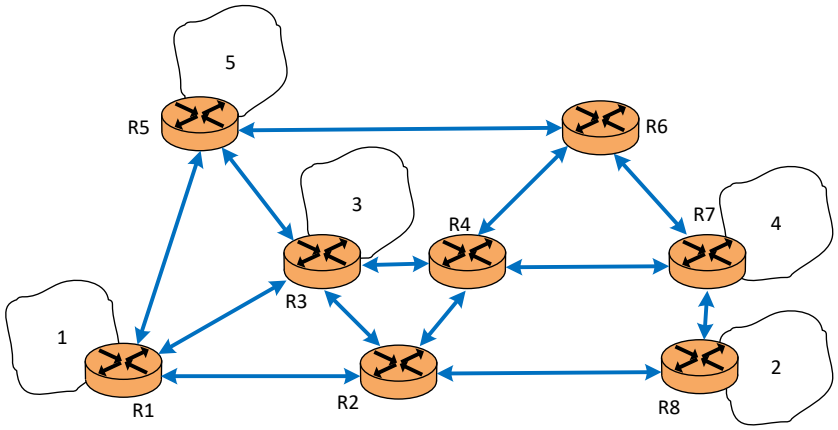
Směrovací funkce určují cestu mezi síťovými adresami. Základní **atributy směrovacích technik** a protokolů jsou:

- **výkonnostní kritéria** (množství uzlů, náklady, zpoždění a propustnost),
- **rozhodovací čas** (pro datagramy, virtuální obvody),
- **rozhodovací místo** (každý uzel, tj. distribuovaně; centrální uzel, tj. centralizovaně),
- **zdroje informací o síti** (žádné, místní, připojené uzly, všechny uzly),
- **směrovací techniky** (pevné, lavinovité, nahodilé, adaptivní),
- **časová aktualizace adaptivního směrování** (průběžné, periodické, hlavní změny zátěže, změny topologie).

7.6.1 Možné strategie směrování nedynamického charakteru

7.6.1.1 Použití pevných cest (statické směrování)

Nejjednodušším řešením je metoda založená na použití pevných (statických) cest. V každém uzlu sítě je definováno, která výstupní linka má být využita pro pakety určené konkrétnímu adresátovi. Je zřejmé, že tím, že je systém nastaven fixně, nemůže pružně reagovat na změny sítě, výpadky či přetížení. Při tomto druhu směrování neexistuje rozdíl mezi nespojovaným přenosem a virtuálními spojeními, neboť cesta všech paketů je vždy stejná. Představme si topologii, která je ukázána na **Obr. 7-13**. Směrovací tabulka např. směrovače R4 poté může principiálně vypadat např. tak, jak ukazuje **Tab. 6**.



Obr. 7-13: Jednoduchá topologie pro účely vysvětlení směrovací strategie

Tab. 6: Možná statická směrovací tabulka z pohledu směrovače R4

Cílová síť	Cesta kudy (další skok)
1	R2
2	R7
3	R3
4	R7
5	R6

7.6.1.2 Náhodné směrování

Teoretická možnost směrování spočívající v tom, že pakety nejsou v uzlech duplikovány, ale náhodně odesílány s tím, že za určitou dobu dojdou k cíli. Pohybují se tak v síti chaoticky. Existují úpravy, kdy různými pravděpodobnostmi pro jednotlivé linky uzlu lze realizovat provoz více deterministický. Metoda značně zatěžuje síť a v praxi se s ní není možné potkat. Její jedinou výhodou je značná jednoduchost.

7.6.1.3 Lavinové směrování

Tento typ směrování spočívá taktéž v jednoduchém principu. Paket je v každém uzlu nakopírován a odeslán přes všechny spoje, s výjimkou té, odkud přišel. Než se tak učiní, testuje se, zda paket už v tomto uzlu nebyl. Tato metoda je velmi odolná vůči poruchám sítě a navíc teoreticky zaručuje, že paket přijde k adresátovi za nejkratší možnou dobu (zkouší se totiž všechny cesty – a tedy i ta nejkratší). Velkou nevýhodou je enormní zátěž sítě mnoha zbytečnými přenosy (*flooding*).

Lavinové směrování proto připadá v úvahu pouze u sítí s malou hustotou provozu nebo lépe pouze v počáteční fázi komunikace, pro nalezení aktuálně nejrychlejší cesty mezi dvojicí uzlů. Z jednoho uzlu se vyšle krátký testovací paket, který si pamatuje, kudy procházel. První takový paket, který dojde k cíli, obsahuje momentálně nejrychlejší cestu.

Tento mechanizmus je využíván u některých mechanizmů hromadné komunikace (tzv. *multicast*), jejichž popis je však nad rámec tohoto textu.

7.6.2 Možné směrovací strategie dynamického charakteru

Cílem dynamických (adaptivních) metod je pružně reagovat na poruchy linek/uzlů, popř. i na přetížení uzlů, a to použitím alternativních cest. Toho lze nejjednodušeji dosáhnout rozšířením směrovacích tabulek tak, že pro každého adresáta obsahují několik výstupních linek, které určují alternativní cesty. Standardně se využívá první z nich, při jejím výpadku se použije záložní trasa. Nutným předpokladem fungování je vysílání služebních zpráv v síti, které o těchto událostech (např. výpadcích uzlů) informují. V uzlech se pak podle nich upravují směrovací tabulky. Nevýhodou těchto metod je vyšší složitost, nároky na paměť a procesorový čas.

7.6.2.1 Centralizované směrování

Adaptivní algoritmus může být koncipován tak, že veškeré informace o aktuálním stavu celé sítě se průběžně shromažďují v jediném centrálním bodě, které pak na jejich základě přijímá všechna potřebná rozhodnutí, a ostatním uzlům je oznamuje. Pak jde o tzv. **centralizované směrování**. Jeho výhodou je možnost optimálního rozhodování na základě znalosti skutečného stavu celé sítě a snadná správa celého systému. Problémů zde existuje více. Pokud má být centralizované směrování opravdu adaptivní, tedy má-li průběžně reagovat na aktuální stav sítě, musí být vyhledávání nejvhodnějších cest prováděno dostatečně často. Dále platí, že výpadek směrovacího centra způsobí kolaps systému. Nezanedbatelná není ani určitá zátěž přenosových cest, která je vytvářena přenosem aktuálních informací o stavu sítě do směrovacího centra, stejně tak jako zpětná distribuce výsledků.

7.6.2.2 Izolované směrování

Alternativou k centralizovanému směrování je tzv. izolované směrování, založené na myšlence, že rozhodovat o nejvhodnější cestě si bude každý uzel sám za sebe, a to na základě takových informací, které dokáže získat sám, bez jakékoliv spolupráce s ostatními uzly. To, že mechanizmus nepovoluje získávání informací od sousedů, je v praxi příliš striktní omezení. Z tohoto důvodu je metoda využívána pouze okrajově, např. ve formě tzv. zpětného učení. U tohoto algoritmu směrovač průběžně sleduje, ze kterého směru dostává pakety pocházející od jiných uzlů. Tím se postupně učí, ve kterém směru se které uzly nacházejí a následně je schopen tyto informace využít. Problémem je, že ani tato metoda nedokáže příliš reagovat na výpadky.

7.6.2.3 Distribuované směrování

Jestliže nebudeme mít v systému směrování žádný centrální prvek, ale zároveň povolíme výměnu informaci mezi sousedy, dostáváme se k tzv. distribuovanému směrování. To předpokládá, že jednotlivé uzly si průběžně vyměňují informace o stavu sítě, a podle nich si pak samy volí příslušné cesty. Tento princip je **v současné praxi nejčastější** a je v souladu s původními koncepcemi budování Internetové sítě, kdy požadavkem bylo, aby existovalo co nejméně centrálních prvků a centralizovaných systémů.

Oproti centrálnímu systému je určování tras distribuováno na jednotlivé uzly systému, avšak oproti izolovanému směrování je povoleno využití informací od sousedů.

Tento druh směrování je v současnosti nejčastěji označován jako dynamické směrování a mechanizmy výměny informací mezi směrovači pak jako směrovací protokoly.

7.6.3 Fungování směrování v sítích TCP/IP

Směrování (*routing*) představuje proces hledání cest z jednoho bodu do jiných bodů v rámci propojených sítí. V rámci sítí TCP/IP je typicky využíván distribuovaný dynamický způsob směrování tak, jak byl popsán v 7.6.2.3. Směrování je netriviální úloha a provádějí ho zpravidla zařízení, které se nazývají směrovače (*routers*). Problémy směrování spočívají především ve **volbě optimální** (nejkratší, nejrychlejší, nejspolehlivější, …) **cesty (***routes***)** ze sítě A do sítě B. Je přitom třeba brát v potaz, že topologie propojených sítí se mění, všechny kanály nemusí být vždy funkční apod.

Každý směrovač, přes který paket na cestě ze sítě A do sítě B prochází, se musí **lokálně rozhodnout** kam paket dále předávat (v případě, že existuje více cest). Toto lokální rozhodnutí je vždy založeno na určité úrovni znalosti globální topologie, což představuje základní problém směrování. Globální topologie je totiž nepředstavitelně složitá a rozsáhlá, dále dynamická, tj. proměnná v čase a navíc je obtížné všechny informace o ní sbírat.

Směrovač potřebuje zpravidla k úspěšnému plnění **směrovací úlohy** tyto informace:

- adresátovu adresu (IP),
- možné cesty do *všech* vzdálených sítí,
- aktuálně zvolenou nejlepší cestu do cílové sítě,
- sousední směrovače, od kterých se může dozvědět o cestách, a poslat jim data,
- způsob, jak se dozvědět o cestách, jak tyto informace aktualizovat a udržovat.

Může samozřejmě nastat i situace, že směrovač nebude vědět kudy paket dále směrovat. V takovém případě paket zahodí a měl by odesílatele paketu o této skutečnosti informovat ICMP zprávou, více viz kap. 7.13.

V rámci Internetu funguje tzv. **hierarchické směrování** neboli směrování s více úrovněmi. Celá síť je rozdělena do tzv. autonomních systémů, které uvnitř provádí směrování na jedné úrovni a mezi těmito částmi je pak prováděno směrování na vyšší úrovni. Na obou těchto úrovních jsou využívány tzv. směrovací protokoly.

Hlavní úlohou směrovacích protokolů je efektivně shromažďovat relevantní směrovací informace. **Základní požadavky na tyto protokoly jsou:**

- **minimalizace velikosti směrovacích tabulek** – z důvodu rychlého vyhledávání a také následně menšího množství vyměňovaných informací mezi sousedy.
- **minimalizace počtu přenášených kontrolních zpráv** – aby nedocházelo k zbytečnému zatížení přenosových linek provozem servisního charakteru, který pro běžného uživatele nemá hodnotu.
- **robustnost** – nesmí docházet ke vzniku „černých děr“, kde by se ztrácely pakety, nebo směrovacích smyček (zacyklení výměny paketů); žádoucí je rychlá konvergence procesu výměny směrovacích informací.
- **využívání optimálních tras –** *optimální* nemusí vždy být nejkratší nebo nejrychlejší.

Pro příklad na dynamické směrování můžeme využít výše uvedený **Obr. 7-13** a následnou **Tab. 6**, která obsahovala možnou podobu staticky nastavené směrovací tabulky. Při použití dynamického směrování by byl hlavní rozdíl v tom, že směrovací tabulka by se v čase měnila a její prvotní obsah by závisel na použitém směrovacím protokolu.

7.6.4 Shrnutí směrování z pohledu síťové vrstvy

V zájmu minimalizace objemu směrovacích tabulek je **směrovací proces** v TCP/IP sítích **založen jen na adresách** (dílčích) **sítí,** případně podsítí (viz 7.5), nikoliv na adresách jednotlivých hostitelských počítačů v rámci těchto sítí. Síťová vrstva tedy pracuje s představou členění Internetu na dílčí sítě a nikoliv s představou jednolité, dále nestrukturované výsledné sítě.

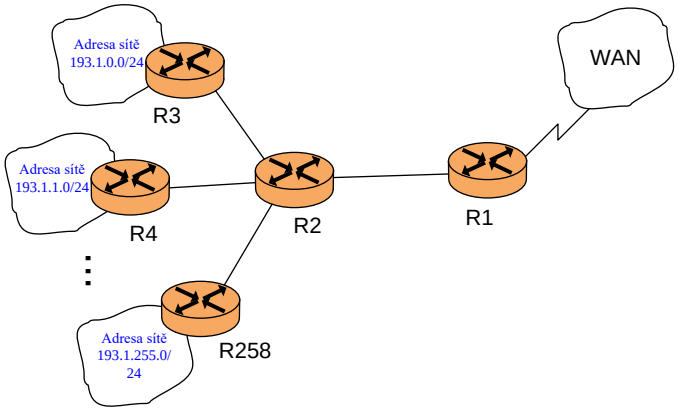
Každý hostitelský počítač, který chce odeslat nějaký IP datagram jinému hostitelskému počítači, dokáže z IP adresy příjemce rozpoznat, zda leží ve stejné (lokální) síti či nikoliv. Pokud ano (nachází-li se například v téže síti typu Ethernet), pošle mu odesílatel svůj datagram přímo, dojde k přímému doručení. Pokud se ale příjemce nachází v jiné síti, předá odesílatel svůj datagram nejbližší bráně (směrovači) na hranici své sítě. Na ní je pak rozhodnout, kudy datagram poslat dále. Podstatné přitom je, že při svém rozhodování vychází brána pouze z té části IP adresy konečného příjemce, která vyjadřuje příslušnou cílovou síť. **Každá brána má své směrovací tabulky ve formě seznamu dvojic <síť, následující skok>** a podle cílové sítě příjemce si v nich najde, které další bráně má příslušný datagram poslat dále. Zbývající část IP adresy příjemce, která vyjadřuje adresu hostitelského počítače v rámci cílové sítě, pak využije až ta brána (poslední v řetězci), která již leží na hranici příslušné cílové sítě, a která pak datagram doručí přímo jeho konečnému adresátovi.

7.6.5 Agregace směrovacích cest

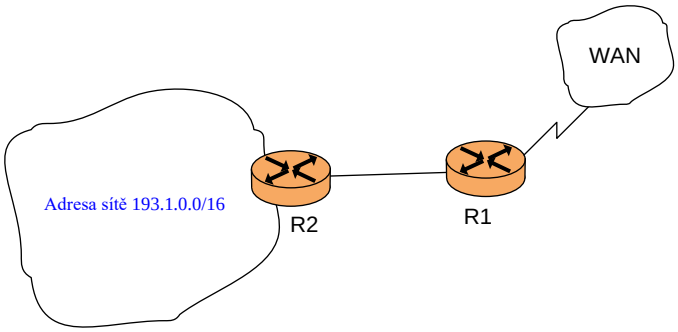
Agregace (*aggregation*) (nazývána i jako **sumarizace**, *summarization*) směrovacích cest představují jednu ze základních technik, které mají za cíl redukci velikosti směrovacích tabulek, a tedy i zrychlení směrovacího procesu. Obě techniky jsou v principu totožné a spočívají ve shrnutí několika směrovacích informací do jedné, souhrnné, nebo též nadřazené. Uveďme si příklad:

Mějme výchozí situaci, která je naznačena na **Obr. 7-14**. Z pohledu směrovače R1 jsou za směrovačem R2 sítě 193.1.0.0/24 až 193.1.255.0/24, připojené přes směrovače R3 až R258. Bez agregace směrovacích cest bude mít směrovač R1 ve své směrovací tabulce informaci o každé z těchto sítí zvlášť, přestože jsou všechny (z jeho pohledu) v jednom směru a všechny dostupné skrz směrovač R2. Je zřejmé, že toto řešení není příliš efektivní, jelikož směrovači R1 by postačoval jeden záznam, že všechny tyto sítě (souhrnně 193.1.0.0/16) jsou dostupné přes R2. Toho lze dosáhnout agregací (sumarizací) směrovacích údajů[27].

V případě aktivované sumarizace se situace z pohledu směrovače R1 (a případně i z celé dále připojené WAN) zjednoduší tak, jak je naznačeno na **Obr. 7-15**. Vytváření nadřazeného adresního rozsahu spočívá především v tom, že se zkracuje délka masky tak, aby sumarizovaná síť obsáhla všechny dílčí sítě. Z toho je zřejmé, že sumarizovat nemůžeme libovolným způsobem, ale že vždy musíme respektovat binární podobu adres, které nám umožňuje postupovat v násobcích mocnin dvou. Zkrácení délky prefixu o jedničku tedy reprezentuje zdvojnásobení rozsahu adres. Tj. např. sumarizovaný rozsah s prefixem délky /23 pojme dvě sítě s délkou prefixu /24, suma /22 pak čtyři sítě /24, atd.



Obr. 7-14: Výchozí situace před agregací směrovacích cest



Obr. 7-15: Zjednodušený pohled směrovače R1 po agregaci směrovacích informací ze sítí na směrovači R3 až R258

[27] Tuto funkci mohou směrovače vykonávat automaticky nebo na základě nastavení administrátora.

Výše uvedený příklad je záměrně vybrán tak, aby sítě měly standardní masky sítě podle dřívějšího rozdělení na třídy. Agregace adres však může fungovat (a velmi účinně funguje) i v případě implementace podsítí. Při sumarizaci se vždy hledá **nejbližší nadřazený adresní prostor**, do kterého spadají všechny (pod)sítě, které chceme agregovat.

Lze konstatovat, že agregace cest je také možnost, jak **izolovat určité směrovače od informací o změnách topologie**, které pro něj **nejsou podstatné**. Tento efekt může být velmi přínosný z hlediska stability směrovacího procesu v celé síti.

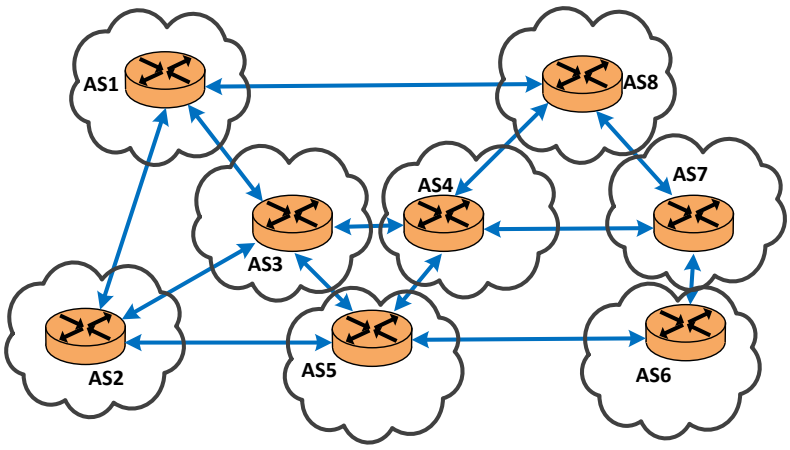
7.6.6 Autonomní systémy

Jednotlivé sítě představují velmi malou jednotku, řadově jich existují milióny. Řešit úlohu směrování v globálním měřítku na úrovni sítí by bylo velice složité. Z tohoto důvodu existují vyšší jednotky Internetové sítě z hlediska topologie, tzv. autonomní systémy, jak bylo v obecné rovině uvedeno již v kap. 3.5. V celém Internetu jsou řádově desetisíce těchto jednotek, které si lze představit jako sítě sítí. Zde se na autonomní systémy podíváme především z hlediska síťové vrstvy a problematiky směrování.

Autonomní systém (AS) představuje souhrn sítí pod společnou správou, kde se zpravidla používá společná (vnitřní) směrovací strategie. Celý AS se obvykle skládá z menších oblastí (sítí). Z hlediska identifikace má každý autonomní systém přiřazeno unikátní 16-bitové číslo nebo 32bitové číslo, tzv. ASN (AS Number)[28]. Aktualizovaný seznam obsazených čísel autonomních systémů lze najít na webových stránkách organizace IANA, http://www.iana.org/assignments/as-numbers/, což je organizace, která tato čísla přiděluje a spravuje.

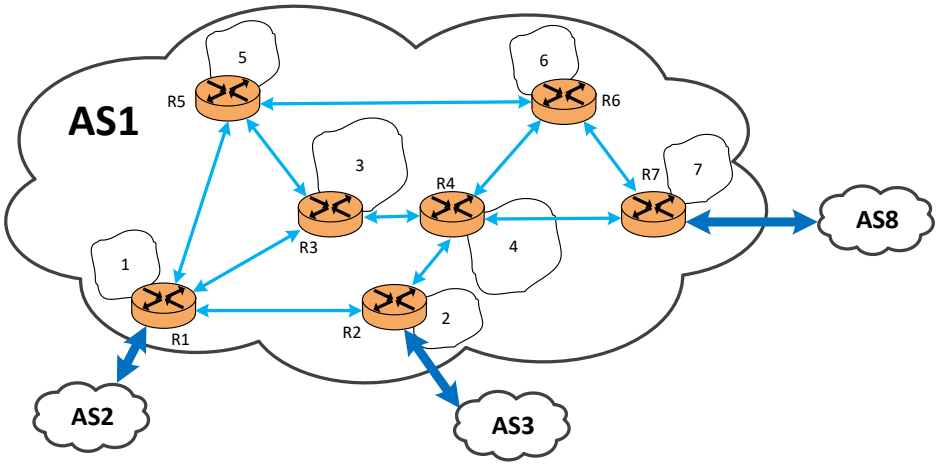
Aby mělo rozdělení na AS smysl, musí mezi nimi existovat jednotný systém předávání směrovacích informací, v pevně daném formátu. To lze vyjádřit tak, že v rámci svého vlastního (autonomního) systému má každý možnost zajistit si přenos a aktualizaci směrovacích údajů podle svého, ale navenek musí všichni postupovat jednotně.

Autonomní systém si lze také představit jako jeden geograficky distribuovaný směrovač, jehož porty jsou reprezentovány porty všech hraničních směrovačů celého AS. Zjednodušený pohled na autonomní systémy a část globální topologie je ukázán na **Obr. 7-16**. Detail jednoho z autonomních systémů je pak pro názornost ukázán na **Obr. 7-17**.



Obr. 7-16: Autonomní systémy a jejich propojení – abstraktní pohled na část globální topologie

[28] Např. česká akademická síť CESNET má číslo ASN 2852.



Obr. 7-17: Detail vnitřního uspořádání jednoho z AS (konkrétně AS1), ilustrační struktura

7.6.7 Směrovací protokoly

Vzhledem k existenci autonomních systémů se směrovací protokoly dělí na dvě základní skupiny:

- **Protokoly pro použití uvnitř autonomního systému** (*interior protocols*), též jako *Internal Gateway Protocols* = IGP. Používané pro **přenos směrovacích informací** mezi jednotlivými směrovači **uvnitř autonomního systému**, tj. mezi směrovači R1 až R7 z **Obr. 7-17**. Mezi používané IGP protokoly patří např.:
  - **RIPv2** (*Routing Information Protocol* verze 2),
  - **EIGRP** (*Enhanced Interior Gateway Routing Protocol*),
  - **OSPF** (*Open Shortest Path First*),
  - **IS-IS** (*Intermediate System to Intermediate System*).

  Podrobnější seznámení s IGP protokoly je nad rámec tohoto textu. Nicméně hlavní odlišnosti mezi těmito protokoly spočívají především v tom, jakým způsobem mají nastaveny parametry komunikace mezi směrovači a také se liší výpočtové mechanizmy pro určení optimální trasy z jedné sítě do druhé.
- **Protokoly pro použití mezi autonomními systémy** (*exterior protocols*), též *External Gateway Protocols* = EGP. Výměna směrovacích informací na úrovni AS (ať už při přímém nebo zprostředkovaném propojení, viz **Obr. 3-3**). Zde je v současnosti využíván výhradně jediný protokol a tím je:
  - **BGP** (*Border Gateway Protocol*).

  I tento protokol je z hlediska fungování nad rámec tohoto textu, základem jeho fungování je především o něco menší úroveň automatizace a o něco větší zásah administrátorů do procesu výběru nejlepších cest pro přenos paketů z jednoho AS do druhého AS. Jako autonomní systém si můžeme představit např. síť poskytovatele připojení (viz např. **Obr. 7-4**), ve které je ve skutečnosti více různých sítí.

7.6.8 Detailní pohled na směrovací tabulku

Směrovací tabulka představuje místo, kam si směrovač ukládá informace o tom, jak má naložit s pakety z hlediska různých cílových sítí, především tedy to, kam je má dále předat. Naplnění směrovací tabulky je typicky důsledkem běhu některého ze směrovacích protokolů, případně i více protokolů dohromady. Směrovací tabulka obvykle obsahuje větší množství záznamů (cest). Každá z nich pak sestává zejména z údajů:

- **původce informace**, typicky některý ze směrovacích protokolů,
- **síťová adresa a maska**, které definují, pro jaký okruh cílových adres tento záznam platí,
- **metrika**, vyjadřující typicky vzdálenost cílové sítě nebo normovanou rychlost tras směrem k této síti,
- **adresu dalšího skoku**, tj. síťová adresa sousedního směrovače, který má být využit k předání paketů směrem k adresátovi,
- **další údaje informativního charakteru**, např. doba, jak dlouho je cesta aktivní.

Zjednodušená ukázka směrovací tabulky (konkrétně ze směrovače firmy Cisco) je na následujícím **Obr. 7-18**, kde každý řádek představuje právě jeden směrovací záznam. Příznaky „R“ a „O“ znamenají RIP, resp. OSPF, následuje informace o síti a masce, pak metrika a následně adresa dalšího skoku. Poslední informací v ukázce je hodnota času, jejíž význam je závislý na konkrétním protokolu.

```
R    192.168.51.0/24 [1] via 172.16.12.1, 00:00:04
R    192.168.50.0/24 [1] via 172.16.12.1, 00:00:24
R    192.168.49.0/24 [1] via 172.16.12.1, 00:00:16
O    192.168.30.0/24 [1563] via 172.16.23.3, 00:00:37
O    192.168.25.0/24 [1563] via 172.16.23.3, 00:00:37
O    192.168.40.0/24 [1563] via 172.16.23.3, 00:00:37
```

Obr. 7-18: Zjednodušená ukázka směrovací tabulky

7.7 IPv4 datagramy

Jak již bylo uvedeno v kap. 7.3, síťová vrstva zavádí jednotnou abstrakci i v případě formátu datových jednotek používaných na této vrstvě, tzv. IP datagramy, též nazývané **pakety**.

IP paket je na úrovni síťového rozhraní vždy zabalen do rámce příslušné technologie (Ethernet, ATM, …), který se mění, tak, jak paket prochází přes dílčí sítě. Zabalený paket však zůstává ve stejném formátu a nemění se, tedy s výjimkou proměnných polí, jako je hodnota čítače, která vyjadřuje životnost paketu (viz dále).

Základní struktura paketu je naznačena na **Obr. 7-19**, detail pak na **Obr. 7-20**. Popis významu jednotlivých polí následuje níže.

20 – 60 bajtů	až (65 535 – záhlaví) bajtů
Záhlaví	Datová část (segment)

Obr. 7-19: Základní pohled na datagram IPv4 protokolu

Bity 0-3	4-7	8-15	16-18	19-31
Verze IP	Délka záhlaví	Typ služby	Celková délka IP datagramu	
Identifikace IP datagramu			Příznaky	Posunutí fragmentu od počátku
Doba života (TTL)		Protokol vyšší vrstvy	Kontrolní součet záhlaví datagramu	
IP adresa odesílatele paketu				
IP adresa příjemce paketu				
Volitelné položky záhlaví				
Přenášená data				

Obr. 7-20: Detail struktury IPv4 datagramu z hlediska položek záhlaví a umístění datové části

Význam jednotlivých polí záhlaví je:

- **Verze** (*version*) – 4 bity, obsahuje verzi protokolu IP a zajišťuje, aby ostatní systémy, které zpracovávají datagram během přenosu, mohly různá pole datagramu správně použít. Verze IPv4 zde má samozřejmě hodnotu 4.
- **Délka záhlaví** (*header length*) – 4 bity, hodnota se musí uvádět, protože záhlaví může mít kvůli volitelným položkám proměnnou délku v násobcích 32 bitů, viz dále. **Minimální délka záhlaví je 20 bajtů** (5 . 32 bitů = 160 bitů = 20 bajtů) délky záhlaví, maximum 60 bajtů, nevyužité pozice rozšiřujících záhlaví musí být "vycpány" daty bez významu[29].
- **Typ služby** (*type of service*, *ToS*) – 8 bitů, položka by měla sloužit ke specifikaci požadované kvality přenosu IP datagramu. Směrování pak může brát ohled na hodnotu ToS a volit z alternativních tras tu, která nejlépe odpovídá požadavkům datagramu. Využití pole v praxi je sporadické, můžeme se setkat s tím, že se položka používá k podobným účelům – nese značku pro mechanizmy zajišťující služby s definovanou kvalitou služby (*QoS*).
- **Celková délka IP datagramu** (*total length*) – 16 bitů, definuje úplnou délku datagramu včetně záhlaví a uživatelských dat. Teoretické maximum je 65535 bajtů.
- **Identifikace IP datagramu** (*identification*) – 16 bitů, primárně určeno k identifikaci k sobě patřících fragmentů, viz kap. 7.8. Vždy přiděleno odesílatelem a hodnota se nemění.

[29] Záhlaví IP protokolu představuje nezbytnou položku přenosu a jako takové tvoří *overhead* přenosu, který snižuje reálnou propustnost sítě.

- **Příznaky** (*flags*) – 3 bity, používají se dva: DF-bit (*don't fragment*) označuje případný požadavek na nepoužití fragmentace, tj. dodatečného dělení paketu na menší části. MF-bit (*more fragments*) říká, že datagram byl fragmentován a že bude následovat další část.
- **Posunutí fragmentu od počátku** (*fragment offset*) – 13 bitů, indikuje pozici obsahu dat datagramu vzhledem k začátku původního (rozděleného) paketu.
- **Doba života datagramu** (*Time To Live* - TTL) – 8 bitů, tato hodnota definuje maximální počet skoků (*hops*) daného paketu. Každý směrovač sníží při zpracování hodnotu položky o 1. Pokud dojde ke snížení na nulu, není paket dále směrován a je zahozen.
- **Protokol vyšší vrstvy** (*protocol*) – 8 bitů, obsahuje číselnou identifikaci protokolu vyšší vrstvy, který využívá IP datagram ke svému přenosu, typicky některý z transportních protokolů.
- **Kontrolní součet záhlaví datagramu** (*header checksum*) – 16 bitů, je počítán pouze ze záhlaví datagramu, nikoliv datové části. Přepočítává se v každém uzlu z důvodu změny obsahu záhlaví paketu. Pokud se při kontrole zjistí, že součet nesedí (tj. došlo k chybě), paket se zahodí.
- **IP adresa odesílatele/příjemce paketu** (*source/destination address*) – každá 32 bitů, jedná se o logickou adresu v rámci IP protokolu. Tato pole jsou zásadní z hlediska směrování paketu. Adresa příjemce slouží k určení trasy paketu jako takového, adresa odesílatele pak pro vytvoření odpovědi.
- **Volitelné položky záhlaví** (*options*) – nepovinné, až do délky 40 bajtů, nevyužívá se příliš často, některé jsou dokonce v současnosti zakázány, protože se v praxi neosvědčily. Např.:
  - zaznamenej směrovače (*record route*) – zjištění kudy paket procházel,
  - zaznamenej čas (*time stamp*),
  - explicitní směrování (*loose source routing*) – umožňuje výslovně zadat, přes které směrovače má být IP datagram dopravován (nemusí být uvedeny všechny),
  - striktní explicitní směrování (*strict source routing*) – musí být zadány všechny mezilehlé směrovače.
- **Přenášená data** (*payload*) – teoreticky až do 65536 bajtů délky (v součtu se záhlavím), jsou to údaje, které IP vrstvě předal protokol vyšší vrstvy, tedy např. TCP segment.

7.8 Fragmentace paketů

Teoretická maximální velikost IP datagramu je 65535 bajtů (položka Celková délka IP datagramu o 16 bitech), viz **Obr. 7-20**. Jak bylo ukázáno např. na **Obr. 7-5**, datagram může od odesílatele k cíli procházet různými sítěmi s využitím různých přenosových technologií. Konkrétní rámec dané technologie pak vždy pracuje s určitým omezením maximální délky datové části, což je z pohledu rámce právě datagram. Tato hodnota se označuje jako maximální velikost paketu, anglicky MTU (Maximum *Transmission Unit*) a v různých sítích je její hodnota rozdílná. Několik hodnot je pro ukázku shrnuto v **Tab. 7**.

Tab. 7: Maximální délka datagramu podle jednotlivých technologií (výběr)

Linkový protokol	MTU [bajty]
Ethernet II	1500
Ethernet 802.3 SNAP	1492
Frame Relay	1600
FDDI	4352
PPP	296
ATM	48

Stanice, která vytváří paket, zpravidla neví, jaké přenosové technologie se po trase směrem k adresátovi konkrétně nacházejí a není tedy (v IPv4 prostředí) běžně schopna stanovit délku paketu tak, aby paket v původní délce prošel až k adresátovi.

Z výše uvedených důvodů vyplývá, že musí existovat **mechanizmy fragmentace** (rozdělení na menší části) a seskládání (opětovné složení datagramu z jednotlivých částí).

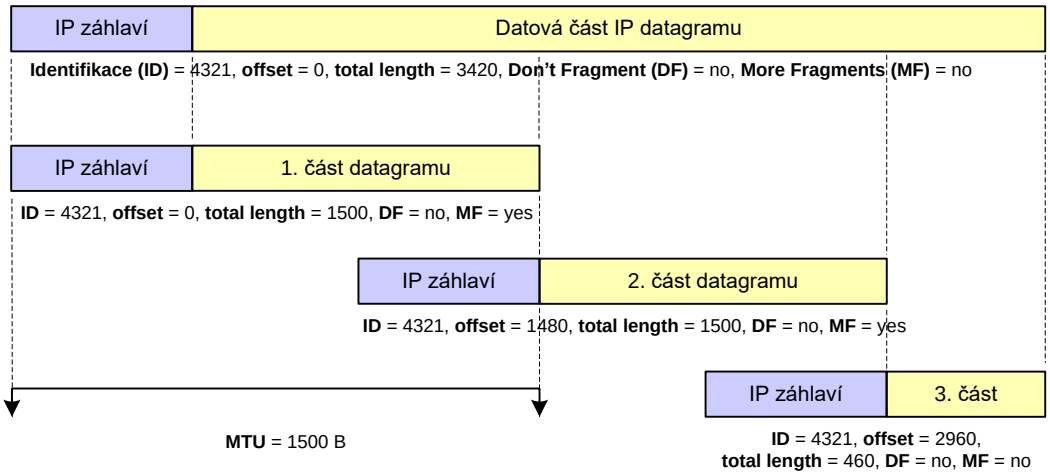
Stanice zpravidla vyšle paket takové délky, aby na síti, kde je přímo připojena, nebyl z pohledu délky paketu a MTU problém. Nicméně během přenosu k cíli může nastat situace, že hodnota MTU další sítě je nižší než MTU předcházející a zároveň nižší než délka přenášeného paketu. Paket v současné podobě není možné dále přenášet a v zásadě existují **dvě** principiální **možnosti**:

- **paket zahodit** a dále nesměrovat, odesílatel by měl být informován (viz kap. 7.13),
- **paket rozdělit** na menší části (fragmenty) a tyto pak přenášet samostatně, což je situace, která nás bude zajímat v dalším popisu.

Pokud v síti připustíme mechanizmus fragmentace, mohou stále nastat obě varianty:

- **fragmentace se provede**, protože není odesílatelem paketu explicitně zakázána, tj. bit DF (*don't fragment*) v IP záhlaví není nastaven,
- **fragmentace se neprovede**, protože bit DF je odesílatelem paketu nastaven, směrovač datagram zahodí a informuje o tom odesílatele pomocí ICMP zprávy. Více o protokolu ICMP je uvedeno v kap. 7.13.

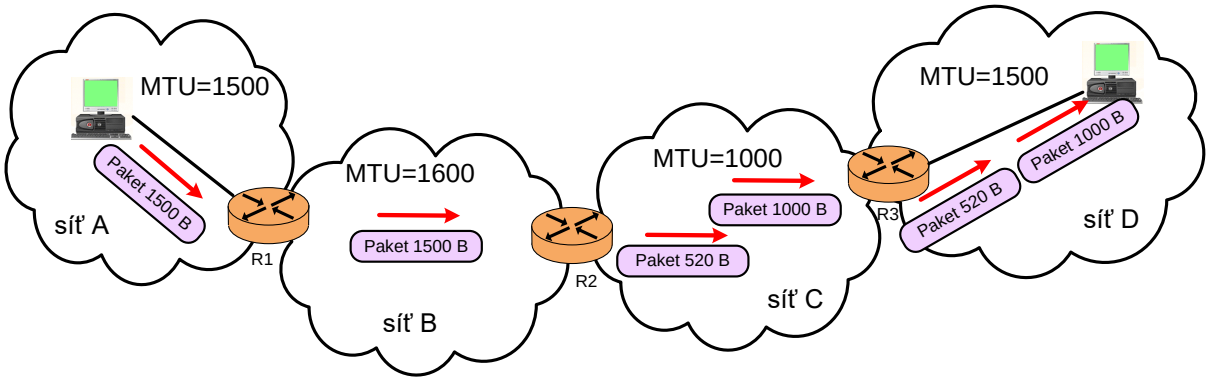
Následující **Obr. 7-21** ukazuje první případ, tj. když ke fragmentaci dojde. Původní paket délky 3420 B je rozdělen (vzhledem k MTU další technologie na trase přenosu) na tři části o délkách 1500 B, 1500 B a 460 B. Součet velikostí dílčích fragmentů je hodnota o 40 B větší než velikost původního datagramu, což je dáno tím, že potřebujeme o dvě IP záhlaví více. U fragmentů se nastavuje pole *Posunutí fragmentu* od počátku (*offset*), celková délka datagramu (*total length*), příznakové bity označující možnost fragmentace (DF) a zda bude následovat další fragment (MF). Že fragmenty patří k sobě, se pozná podle stejné hodnoty pole *Identifikace IP datagramu* (ID). Všechna ostatní pole IP záhlaví zůstávají stejná jako v původním datagramu. Každý fragment je následně opatřen záhlavím příslušné technologie, např. Ethernet, případně i zápatím, a vyslán do přenosového kanálu.



Obr. 7-21: Schematická ukázka fragmentace velkého IP datagramu

Pakety, u kterých byla v průběhu přenosu provedena fragmentace, jsou dále směrovány každý samostatně a k jejich znovusložení dochází vždy až u konečného příjemce. Nedochází tedy k tomu, že by v průběhu přenosu byly fragmenty v některém z mezilehlých uzlů opět složeny. Fungování ilustruje následující **Obr. 7-22**.

Výchozí síť (A) má MTU 1500 bajtů, a aby byla kapacita sítě co nejlépe využita, stanice vytvoří paket právě této délky. Následující síť (B) má MTU vyšší, což nemá na náš paket žádný vliv, je ponechán a přenášen směrem k adresátovi. Síť C má MTU pouze 1000 bajtů, hraniční směrovač proto musí provést fragmentaci, v tomto případě na dva pakety příslušné délky. Poslední síť, v které se nachází příjemce MTU vyšší, než je současná délka paketů (fragmentů), takže je možné je bez jakékoliv další změny doručit adresátovi.



Obr. 7-22: Fungování fragmentace paketu při přenosu přes sítě s různým MTU

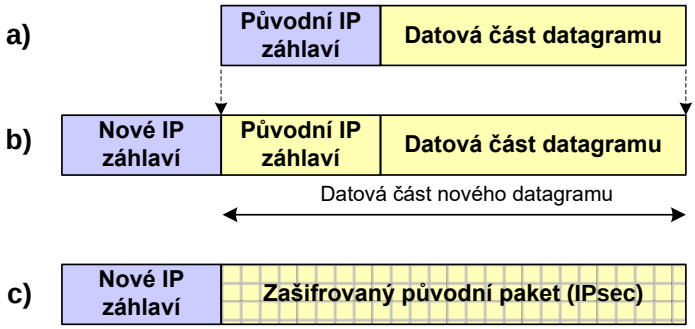
7.9 Tunelování paketů

Existují situace, kdy je nutné propojit několik vzdálených sítí tak, aby se tvářily jako jedna síť. Tyto vzdálené sítě jsou typicky propojeny přes veřejný Internet. Principem tunelování je zapouzdřování původního IP paketu do nového IP paketu (záhlaví). Nový IP paket se liší především tím, jakou má cílovou IP adresu a samozřejmě také jakou zdrojovou IP adresu. Zapouzdření typicky provádí odchozí brána jedné lokální sítě, zapouzdřený paket

putuje Internetovou sítí a po přijetí bránou cílové sítě je paket zbaven přídavného záhlaví a zaslán standardními postupy k adresátovi.

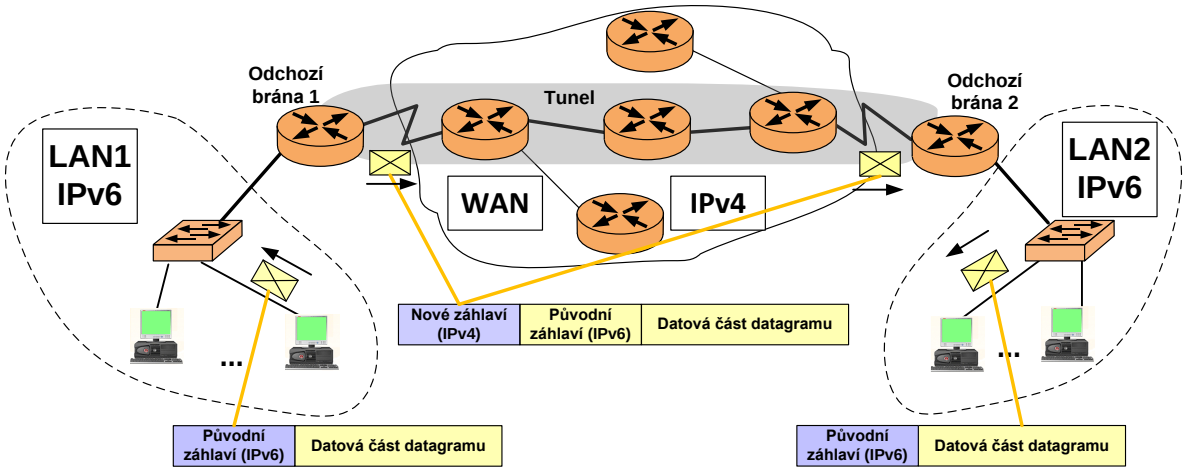
Tunelování je typicky dvojího druhu

- **tunelování** prováděné ve spolupráci **s IPsec** protokolem[30]. IP adresy bran na hranicích privátních a veřejných sítí jsou užity ke směrování. Celý obsah paketu, včetně vnitřní IP adresy zdroje a cíle hostitelského počítače vnitřní sítě, je skrytý vnějšímu světu. Grafické znázornění zapouzdřování je možné nalézt na **Obr. 7-23**.



Obr. 7-23: Zapouzdření paketu při tunelování a) původní paket b) zapouzdřený paket c) zapouzdřený paket s použitím protokolu IPsec

- IP **tunelování** je velmi užitečné v situaci, kdy existuje **více verzí IP protokolu** (typicky IPv4 a IPv6). Mějme např. dvě oddělené počítačové sítě používající stejný síťový protokol (např. IPv6), které jsou navzájem propojeny sítí s odlišnou verzí protokolu IP, tedy IPv4. (O protokolu IPv6 a i o využití tunelování je krátce pojednáno v kap. 7.14.) Mezilehlá síť (WAN) neumí směrovat pakety s IPv6 záhlavím. Aby byl přenos těchto paketů možný, musí být původní pakety zabaleny do nového záhlaví. Dochází tedy k tunelování původních paketů IPv6 prostřednictvím IPv4 sítě. Možnou situaci ilustruje **Obr. 7-24**. Tento typ tunelování představuje jeden z přechodových mechanizmů na protokol IPv6, které jsou popsány v kap. 7.14.4.



Obr. 7-24: Tunelování paketu IPv6 sítí s protokolem IPv4

[30] Popis tohoto protokolu je nad rámec textu, nicméně jeho hlavní funkcí je možnost šifrování celého původního paketu, včetně záhlaví.

7.10 Návaznost IP adres na adresy nižší úrovně

V kap. 7.5 jsme si popsali IP adresy. Ty představují jednotný způsob adresace, který používá libovolný konglomerát vzájemně propojených sítí na bázi soustavy protokolů TCP/IP. Jsou však stále jen **abstrakcí na úrovni síťové vrstvy**, která odpovídá představě jednotné virtuální sítě. Ta je ale ve skutečnosti realizována dílčími sítěmi více či méně odlišného typu, které používají své vlastní linkové mechanizmy adresování a formáty adres. Proto také IP adresy musí být převáděny na tyto konkrétní (fyzické) adresy. Tato operace byla zahrnuta i v popisu síťové vrstvy v jednotlivých bodech přenosu, konkrétně u zdroje (**Obr. 7-6**) a i u mezilehlého síťového prvku (**Obr. 7-7**).

Základní způsoby řešení převodu IP adresy na fyzickou adresu jsou dva:

- **pomocí přímého převodu**

  Velmi jednoduchá myšlenka, která se v této souvislosti sama nabízí, je řešit převod přímočaře, např. pomocí vhodné transformační nebo matematické funkce (způsobu převodu). Tohoto principu lze v některých případech využít. Např. tam, kde si zřizovatel sítě může fyzické adresy jednotlivých uzlů volit sám, podle vlastních potřeb. Má-li například volitelná fyzická adresa rozsah 8 bitů, je nejjednodušší volit ji shodně s posledním oktetem (posledními osmi bity) IP adresy. U větších sítí pak lze využít více bitů IP adresy. Transformace IP adresy na fyzickou se pak stává zcela triviální matematickou úlohou a není třeba udržovat tabulku odpovídajících si adres či nějak složitě řešit převod, protože na základě IP adresy je ihned známá i fyzická adresa. V současnosti se tento princip využívá u techniky *multicast,* což je však problematika nad rámec tohoto textu.

- **pomocí dynamické vazby**

  Nastavování fyzické adresy přímo na každém síťovém adaptéru při jeho instalaci je v praxi únosné jen pro sítě malého rozsahu. Technika je pak spojena s potenciálním rizikem lidských chyb, které mohou vyústit v existenci dvou adaptérů, resp. uzlů se stejnou fyzickou adresou v jedné síti. Většina běžných síťových technologií se proto k problému staví opačně – uživateli nedává možnost ovlivnit fyzickou adresu síťového adaptéru, ta je totiž u každého adaptéru předem pevně dána[31].

  Takto je tomu například u lokálních sítí typu Ethernet. Ty používají fyzické adresy v rozsahu 48 bitů (viz kap. 6.5.1), které v příslušných síťových adaptérech nastavuje přímo jejich výrobce.

  Jakmile je ale potřeba transformovat 32bitové IPv4 adresy (či 128bitové IPv6 adresy) na 48-bitové Ethernetové, nezbývá jinak, než využít převodní tabulky, definující vzájemnou vazbu mezi jednotlivými adresami. Aby vše fungovalo v proměnném prostředí, tabulka nemůže být statická, ale naopak musí být dynamická, která se vytváří a modifikuje průběžně, podle okamžitého stavu sítě.

Komunikující uzel potřebuje při odesílání rámce zjistit fyzickou adresu, což v IPv4 sítích provádí protokol **ARP** (*Address Resolution Protocol*), který je popsán v následující kapitole.

[31] Většina operačních systémů pak samozřejmě umožňuje fyzickou adresu přenastavit administrátorem systému, není to však nezbytně nutné.

7.10.1 Address Resolution Protocol (ARP)

Problém transformace adres vyšší úrovně na adresy nižší úrovně, konkrétně nejčastěji **nalezení odpovídající fyzické adresy k IPv4 adrese, se označuje jako *address resolution problem***. Je možné jej řešit například formou tabulky, obsahující seznam vzájemně si odpovídajících adres. Je to následně spojeno s četnými problémy - kdo a jak zajistí počáteční naplnění tabulky, kdo ji bude udržovat a přizpůsobovat momentálnímu stavu sítě, kdo zajistí, aby její velikost nepřesáhla únosnou mez atd. **ARP** je protokolem, který **řeší *address resolution problem*** právě **pomocí tabulky** dočasných záznamů (*cache*).

Základní vlastnosti ARP

- Dynamický, distribuovaný protokol, schopný reagovat na změny v síti,
- určen primárně k hledání neznámé fyzické/linkové (MAC) adresy na lokální síti, v situaci kdy známe adresu IP; v obecném případě zjištění adresy druhé úrovně na základě znalosti adresy třetí úrovně.
- informace o odpovídajících si adresách se ukládají do tabulky, podle potřeby se obnovují, položky jsou zpravidla uloženy pouze dočasně na několik minut a pak vymazány, protože se mohly stát neaktuální (IP adresa uzlu se mohla změnit) anebo již nejsou potřeba,
- ARP pracuje „mezi“ spojovou a síťovou vrstvou, používá rámce spojové, např. u Ethernetu je v položce *typ* hodnota indikující ARP rovna 0x0806.

Struktura ARP paketu je znázorněna na **Obr. 7-25**. Jednotlivé položky mají následující význam:

- **Typ média** (*Hardware type*) – 16 bitů délky, hodnota indikující typ použitého média, resp. spojové technologie, např. pro Ethernet je hodnota 0x0001, ATM má 0x0010.
- **Typ protokolu** (*Protocol type*) – 16 bitů, hodnota indikuje typ vyššího protokolu, v rámci něhož se logická adresa používá, pro IP je hodnota 0x0800.
- **Délka fyzické adresy** (*Hardware length*) – 8 bitů, délka fyzické adresy v bajtech, pro Ethernet 0x06.
- **Délka logické adresy** (*Protocol length*) – 8 bitů, délka logické adresy taktéž v bajtech, pro IPv4 adresu 0x04.
- **Operace** (*Operation*) – 16 bitů, specifikuje operaci, kterou odesílatel paketu provedl – hodnota 0x0001 pro požadavek na zjištění fyzické adresy, hodnota 0x0002 pro odpověď.
- **Fyzická adresa zdroje / hledaná** (*Sender / target hardware address*) – délka je specifikována v poli *délka fyzické adresy*, obsahuje fyzickou adresu zdroje / hledanou.
- **Logická adresa zdroje / hledaná** (*Sender / target logical address*) – délka je specifikována v poli *délka logické adresy*, obsahuje logickou adresu zdroje / hledanou.

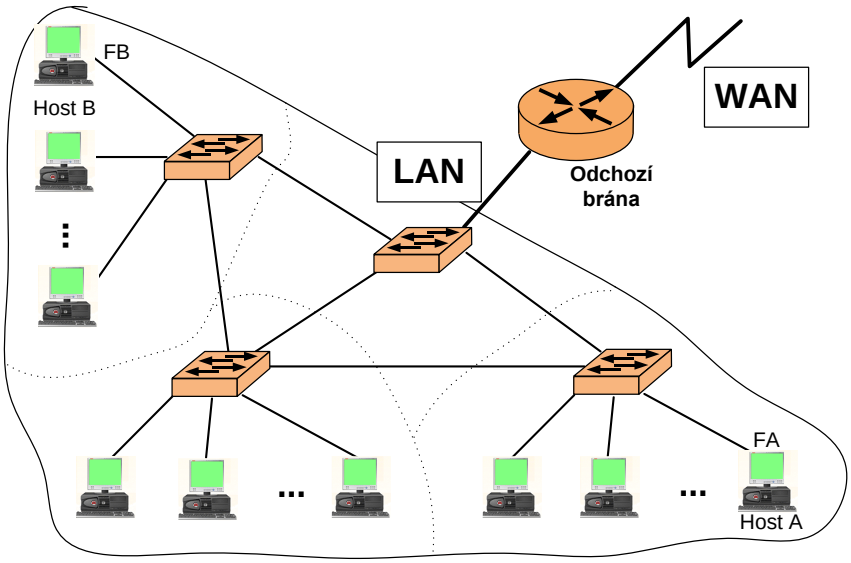
Typ média		Typ protokolu
Délka fyzické adresy	Délka logické adresy	Operace
Fyzická adresa zdroje (zpravidla MAC adresa)		
Logická adresa zdroje (zpravidla IP adresa)		
Hledaná fyzická adresa (zpravidla MAC adresa)		
Hledaná logická adresa (zpravidla IP adresa)		

Obr. 7-25: Členění ARP paketu na jednotlivá pole

Pochopení fungování protokolu ARP je popsáno na dvou následujících příkladech.

Popis první situace – Představme si dva hostitelské počítače A a B, které mají IP adresy IA a IB. Předpokládejme dále, že **jde o uzly téže (dílčí) sítě**, které díky tomu mohou mezi sebou komunikovat přímo, viz **Obr. 7-26**. V rámci „své“ dílčí sítě přitom mají oba uzly fyzické adresy FA a FB. Jestliže nyní síťová vrstva počítače A dostane od své transportní vrstvy za úkol přenést určitá data počítači s IP adresou IB (tj. počítači B), musí být schopna zajistit převod IP adresy (IB) na fyzickou adresu (FB). Tento údaj je nezbytný, aby mohl být vytvořen rámec.

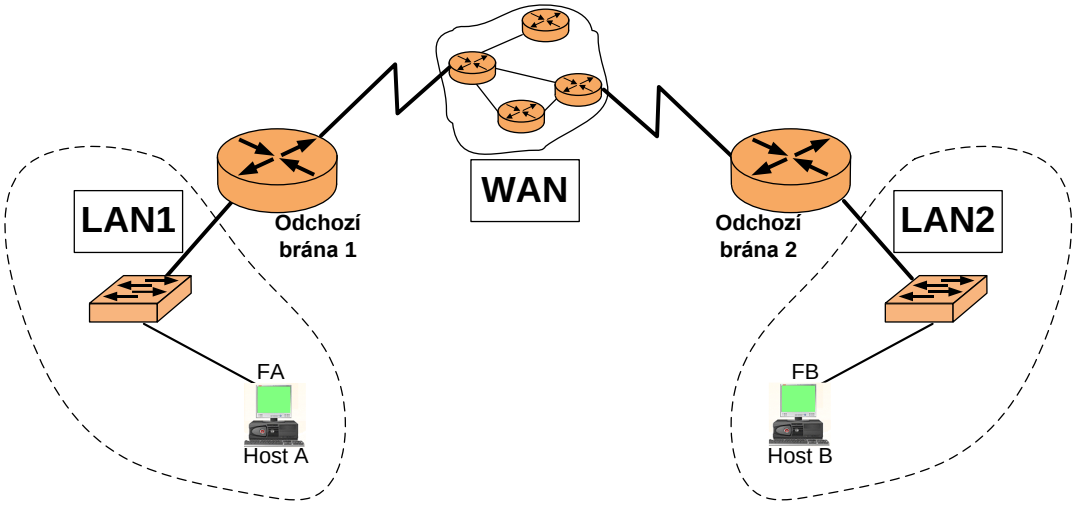
Řešení první situace – Stanice A prozkoumá svoji tabulku odpovídajících si fyzických a síťových adres (ARP *cache*), a pokud nenajde informaci odpovídající záznam, musí použít protokol ARP. Vyšle žádost (*request*) protokolu ARP s informacemi o zdrojové dvojici adres (fyzické a síťové – FA a IA) a s hledanou IP adresou (IB), kterou adresuje všem stanicím v síti (na všeobecnou adresu MAC – *broadcast*). Žádost přijmou všechny stanice v síti. Odpověď pak odešle pouze stanice B, ostatní paket zahodí. Stanice B odpoví (*reply*) zprávou s „vyplněným“ polem hledané fyzické adresy FB, která je zaslána přímo na adresu (FA) zdrojové stanice. Současně stanice B zkontroluje obsah své paměti ARP, zda ji nedoplnit o dvojici adres (IA a FA) obdržených v žádosti ARP, pro pozdější použití.



Obr. 7-26: Ilustrace situace, kdy zdrojová a hledaná stanice jsou na jednom segmentu sítě

Popis druhé situace – stejně jako v předcházejícím příkladu, jen stanice A a B nejsou v rámci jedné sítě, viz **Obr. 7-27**.

Řešení druhé situace – Pokud hledaná stanice není ve stejné síti (lze zjistit snadno výpočty, které byly popsány v kap. 7.5), potom stanice zasílá rámec na fyzickou adresu výchozí brány (*default gateway*) a ta se chová jako zástupce hledané stanice. Pokud stanice fyzickou adresu směrovače nezná, zjistí si ji stejně, jakoby zjišťovala adresu stanice, tj. opět pomocí ARP. Výchozí brána následně paket odešle směrem do sítě, kde se nachází cílová IP adresa (IB).



Obr. 7-27: Ilustrace situace, kdy zdrojová a hledaná stanice nejsou na stejné síti

Obdobně si lze fungování ARP představit i z pohledu směrovačů. Ty také pracují s pakety, u kterých se rozhodují, kterým směrem je zaslat, případně komu je doručit. Z tohoto důvodu musí také vytvářet rámce konkrétní spojové technologie, do kterých je třeba vyplnit cílovou linkovou adresu. Pokud směrovač tuto adresu nezná, může si ji pomoci protokolu ARP zjistit stejným způsobem, jako to činí stanice v předcházejících dvou příkladech.

ARP tabulka

Jak již bylo uvedeno, protokol ARP pracuje s tabulkou záznamů vzájemně odpovídajících si záznamů – IP adres a fyzických adres. Na následujícím **Obr. 7-28** je ukázka krátké tabulky[32]. Pokud bude chtít tato stanice odeslat paket např. na IP adresu 100.100.100.192, v tabulce najde odpovídající záznam a bude vědět, že cílová adresa v rámci má být 50:e5:49:35:6b:e2. Ve třetím sloupečku je informace o tom, že tato informace byla získána dynamicky, tzn., že její zápis do tabulky je výsledkem běhu protokolu ARP.

Poslední řádek představuje speciální typ adresy (*multicast*), u které je vidět, že se jedná o statický překlad. V tomto případě byla informace zjištěna jiným způsobem (výpočtem) a nepředpokládá se u ní změna v čase.

Tabulka neukazuje u žádného ze záznamů časové údaje, nicméně v rámci konkrétní implementace protokolu ARP se s nimi pracuje. Každý záznam je zapsán pouze na omezenou dobu, typicky v řádu minut a poté je odstraněn. Pokud je opět potřeba, musí se znovu dynamicky zjistit zprávou ARP žádost a na základě následné odpovědi je výsledek znovu

[32] Výpis byl obdržen na systému Windows 7, v IPv4 síti nad Ethernetem.

zapsán. Tento mechanizmus umožňuje systému reagovat na změny (typicky změny IP adres, ke kterým může v síti docházet).

```
C:\>arp -a

Rozhraní: 100.100.100.55 --- 0xa
  internetová adresa    fyzická adresa         typ
  100.100.100.1         00-17-a4-c2-09-00      dynamická
  100.100.100.192       50-e5-49-35-6b-e2      dynamická
  100.100.100.152       50-e5-49-3c-61-bb      dynamická
  224.0.0.252           01-00-5e-00-00-fc      statická
```

Obr. 7-28: Ukázka tabulky záznamů protokolu ARP

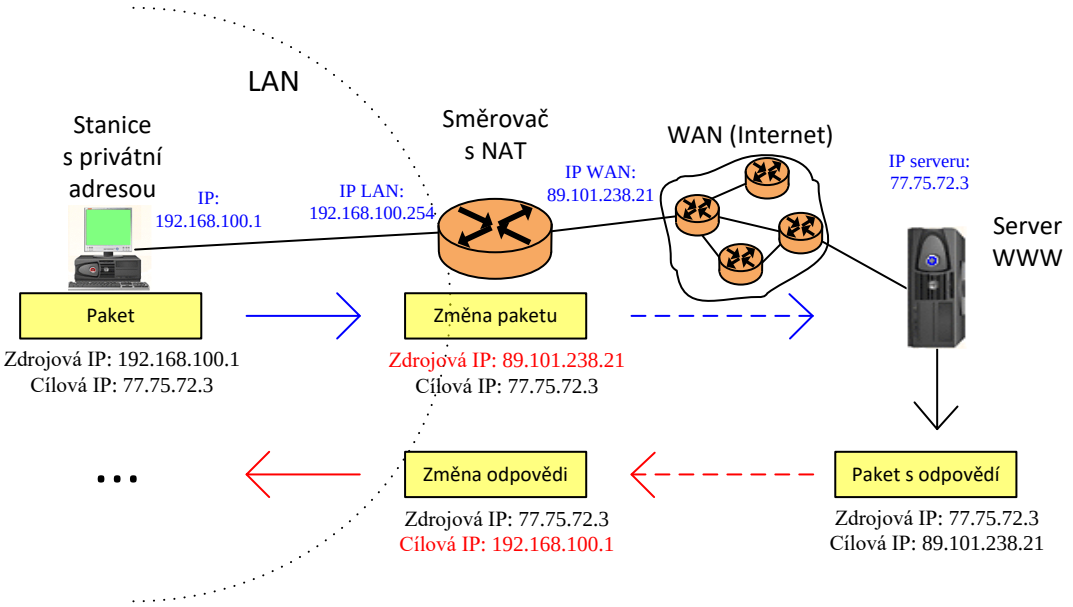
7.11 Network Address Translation (NAT)

NAT, česky *překlad síťových adres*, je funkce směrovače umožňující změnu IP adresy v záhlaví paketu, který jím prochází. Zpravidla se zdrojová nebo cílová IP adresa převádí mezi různými rozsahy (kap. 7.5). Nejběžnější formou je, když směrovač IP adresy z nějakého rozsahu mění na svoji IP adresu a naopak – tím umožňuje, aby počítače ve vnitřní síti vystupovaly v Internetu pod jinou (zpravidla jeho) IP adresou. Tuto funkci podporují prakticky všechny běžné směrovače. Technika překladu (IP) adres tedy umožňuje oddělit interní síť (*intranet*) od Internetu, což může být výhodné i z bezpečnostního hlediska.

Směrovač, na kterém běží NAT, musí být schopen navenek nějakým způsobem odlišit provoz jednotlivých stanic z vnitřní sítě do Internetu. To provádí na základě tabulky překladu adres, kterou si po celou dobu komunikace drží v paměti. Ve většině případů má k dispozici jen jednu (veřejnou) IP adresu, která je přiřazena na jeho tzv. WAN (*Wide Area Network*) port, napojený směrem do Internetu a ve vnitřní síti je více (privátních) IP adres. V tomto případě si směrovač nevystačí pouze se síťovými adresami a musí použít i vyšší (transportní) adresy – porty (viz kap. 8). NAT však může obecně překládat IP adresy i jiným způsobem a na jiných místech sítě.

Technika NAT může být provedena při přenosu paketu i **vícekrát**. Z jednoho bodu v komunikačním řetězci nelze stanovit, zda po trase někde k překladu dojde nebo ne. Překlad adres může probíhat i mezi verzemi protokolu IP (IPv4 a IPv6) – **PT** (*protocol translation*), častěji se však setkáme s technikou **IP tunelování**, více viz kap.7.9 a 7.14.4.

Uveďme si příklad na jednoduché využití NATu, u kterého pomineme v rámci zjednodušení problematiku transportních adres. Stanice v lokální síti s privátní adresou 192.168.100.1 se snaží o spojení s www serverem s IP adresou 77.75.72.3. Paket dorazí na směrovač s funkcí NAT a ten změní zdrojovou IP adresu v paketu na svoji (veřejnou), např. 89.101.238.21. Paket odchází ze směrovače do Internetu. Pokud na směrovač dorazí odpověď, je předána do vnitřní sítě, avšak až poté, co se cílová adresa změní na původní zdrojovou, tj. 192.168.100.1. Situaci ilustruje **Obr. 7-29**. Pozn.: Komunikace může být iniciována vždy pouze počítačem z vnitřní sítě.



Obr. 7-29: Ukázka fungování techniky překladu adres (NAT), bez zapojení modifikace transportních adres

7.11.1 Dva základní druhy překladu adres

Existuje mnoho technik, které jsou označovány jako NAT, či s NATem souvisí. V různých zdrojích je možné nalézt různé způsoby rozdělení. My si zde uvedeme pouze základní dělení na:

- **SNAT** (*Source NAT*) – prvotně je prováděn překlad zdrojové IP adresy a případně transportní adresy.
- **DNAT** (*Destination NAT*) – prvotně prováděn překlad cílové IP adresy a případně opět transportní adresy. DNAT se primárně používá ke „zveřejnění“ služby z interní sítě na veřejně přístupnou IP adresu.

Příklad z předchozí kapitoly je příkladem obou těchto technik NAT. Prvotně se však jedná o SNAT, jelikož překlad byl zahájen záměnou zdrojové IP adresy. Následná záměna cílové IP už je pouze nezbytná pro správné fungování přenosu.

7.11.2 Výhody a nevýhody NATu

Z předcházejícího textu je zřejmé, že NAT přináší i jistá omezení. Problém spočívá v tom, že při použití techniky NAT ztrácíme jednu ze základních předností Internetových sítí postavených na sadě TCP/IP – obousměrnou koncovou konektivitu (*end-to-end*). **Přímočaré spojení dvou koncových uzlů je s použitím techniky NAT vždy nějakým způsobem omezeno.** To může být problematické pro některé Internetové protokoly. Nejproblematičtější je situace, kdy oba koncové systémy jsou odděleny od vnějších sítí prostřednictvím NATu.

Za určitou nevýhodu je možné považovat i určité **časové zpoždění,** které s NATem nutně souvisí. Je logické, že překlad adres zabere více času, než jednoduché směrování.

Bezpečnostní výhody **NAT**u již byly uvedeny. V případě použití SNATu může komunikace vzejít pouze z vnitřní sítě. Z vnějšku nemůže být spojení zahájeno (to lze pouze s DNAT), což **uživatele ochrání před mnohými škodlivými útoky**.

Za velmi podstatnou výhodu se také v případě IPv4 sítí považuje **úspora adresního prostoru veřejného typu**. Za NATem jsou schovány privátní IP adresy, jejichž použití může být v rámci celého Internetu neomezeně opakováno. Samozřejmě za předpokladu, že tyto sítě jsou do veřejného Internetu odděleny prostřednictvím NATu.

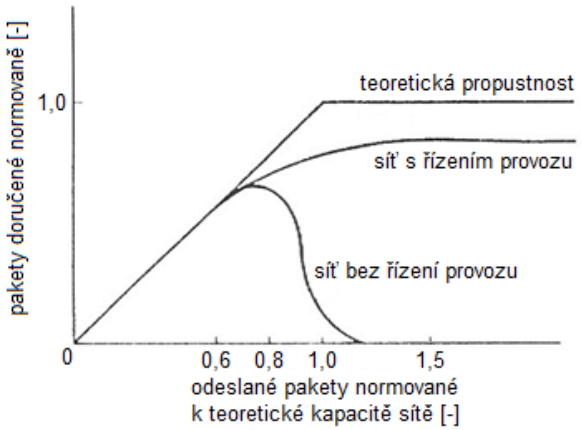
7.12 Mechanizmy řízení provozu v síťové vrstvě

O řízení toku byla již řeč v rámci spojové vrstvy, např. viz kap. 6.7.6. S řízením provozu se však setkáváme i na síťové vrstvě, kde již komunikace probíhá přes sítě, a tak dílčí řízení toku jednotlivých linek na nižší vrstvě nemusí být dostatečné. Snahou řízení přenosu paketů by mělo být to, aby nedocházelo k zahlcení mezilehlých uzlů sítě anebo k zahlcení přijímací strany.

Jak je patrné z následujícího přehledu, řízení toku je jen jedna z oblastí řízení přenosu paketů:

- **řízení toku dat** (*flow control*) – regulace přenosu paketů mezi dvěma uzly,
- **předcházení zahlcení sítě** (*congestion avoidance*), což je stav kdy většina uzlů sítě je zahlcena; či **předcházení stavu uváznutí**,
- **směrování s přerozdělováním zátěže** (*load balancing*), které umožňuje např. rozdělit pakety do více tras a tím snížit zátěž mezilehlých uzlů a linek.

Následující **Obr. 7-30** ilustruje, jak by měla vypadat propustnost sítě v ideálním případě, v síti bez řízení provozu a zejména v síti s řízením provozu. Je zřejmé, že od určité míry zatížení nelze vlivem reálných vlastností dosahovat teoretické propustnosti a v případě, že v síti neexistují žádné mechanizmy, propustnost sítě od určitého zatížení rapidně klesá.



Obr. 7-30: Propustnost sítě v různých situacích vzhledem k mechanizmům řízení provozu

7.12.1 Řízení toku dat v síťové vrstvě

Tento mechanizmus se používá pro omezení rychlosti generování datových jednotek ve vysílači, tj. ve zdrojové stanici. Cílem je zamezení zahlcení přijímače, což může mít samozřejmě vliv i na zahlcení celé sítě. Existují tři základní metody:

- **Úprava rychlosti generování datových jednotek** – tato úprava je realizována změnou prodlevy časovače, který řídí generování paketů. V principu lze využít tzv. škrtících paketů (*choke packets*), které vysílá přijímač. Vysílač při příjmu takového paketu sníží rychlost a zároveň startuje časovač. Po uplynutí časovače se opět rychlost zvýší, což vede k tomu, že snížení rychlosti je pouze dočasné. Pokud příjemce nadále nezvládá zpracovávat pakety, je možné zaslat *choke packet* opakovaně.
- **Odmítnutí paketu přijímačem** – je založeno na principu, že přijímač pakety nad jeho možnosti neuloží do paměti, dojde k jeho zahození (*discard*) a paket je ztracen. Přijímač o této skutečnosti může, ale nemusí informovat vysílač a ten může na vzniklou situaci reagovat.
- **Povolení k vysílání** – existuje více variant, všechny jsou však založeny na explicitním povolení vysílání přijímačem.

7.12.2 Předcházení zahlcení sítě

Pro předcházení stavu zahlcení sítě existuje více metod. My se zde zaměříme na jednoduchou metodu, která spočívá ve **snížení existující zátěže**. Postup je založen na zahazování určitého množství paketů tak, aby se snížil jejich celkový počet v síti. Lze zahazovat např. pakety, které jsou už příliš dlouho v síti nebo už prošly mnoha uzly. Je možné zahazovat všechny pakety, které vstupují do uzlu, po překročení nějaké přednastavené hladiny. Tento způsob snižování zátěže se používá běžně v datagramových sítích. Zahození paketu je krajní řešení, které nicméně nemá nenapravitelné důsledky. Transportní vrstva totiž typicky sleduje povolené prodlevy mezi příchody paketů na straně přijímače, případně prodlevy do příchodu potvrzení na straně vysílače. Řešením je vždy opakování přenosu (vyžádané přijímačem, automatické z vysílače). To může bez dalších mechanizmů vést k cyklickému zahlcování sítě.

7.12.3 Předcházení uváznutí sítě

Uváznutí sítě může nastat, když uzly nejsou schopny posílat pakety směrem k adresátovi, např. protože následující uzel má zaplněnu vyrovnávací paměť.

Jedna z možností, jak předcházet uváznutí sítě je vytvořit strukturovanou vyrovnávací paměť. Vyrovnávací paměť je organizována hierarchickým způsobem na několika úrovních. Hlavní část vyrovnávací paměti je použitelná bez omezení – jakýkoliv příchozí paket v ní může být uchován. Tato část tvoří většinu celé vyrovnávací paměti. Dále existují jeden nebo více bloků vyrovnávací paměti, které jsou určitým způsobem rezervovány pro pakety vyšší důležitosti. Běžný provoz poté nemůže nikdy zcela zahltit síťový prvek a způsobit uváznutí sítě.

Další možností, která je taktéž standardně využívána, je opatřit pakety v síti hodnotou definující maximální dobu životnosti paketu TTL (*Time to Live*), resp. *Hop limit*. Po vypršení tohoto počtu je paket bez náhrady zahozen. Tím je zajištěno, že nebudou přenášeny sítí donekonečna (např. při chybě směrování).

7.13 Internet Control Message Protocol verze 4 (ICMPv4)

7.13.1 Základní popis protokolu

IP protokol představuje základní protokol síťové vrstvy, který je široce využíván pro přenos paketů. Popis protokolu IPv4 a zejména jeho záhlaví lze nalézt v kap. 7.7, ze kterého je patrné, že protokol IP neobsahuje žádné mechanizmy hlášení chyb či oprav chyb, ke kterým dojde při komunikaci na síťové vrstvě. Občas ale v každé síti dojde k chybě, směrovač musí např. zahodit paket, protože mu vypršela doba života nebo není prostor ve vyrovnávací paměti (viz kap. 7.12) a bylo by vhodné v těchto i dalších případech upozornit původce zprávy na vzniklý problém.

IP protokol dále neumožňuje testovat dostupnost určité stanice či zobrazit aktuální zvolenou přenosovou trasu. Tyto informace jsou však často velmi důležité.

Protokol ICMP (*Internet Control Message Protocol* – protokol služebních hlášení) je servisní protokol, což znamená, že nepřenáší žádná uživatelská data. Jedná se o klasický příklad aplikace typu klient-server. ICMP je součástí sady TCP/IP protokolů a slouží IP protokolu především k vyřešení výše uvedených nedostatků, tj. umožňuje signalizaci mimořádných událostí v síti a testování konektivity. Jeho obsah se přenáší přímo v IP datagramech, viz **Obr. 7-31** znázorňující zapouzdřování ICMP konkrétně v prostředí Ethernetu.

Obr. 7-31: Zapouzdření ICMP paketu přenášeného v síti Ethernet

ICMP zprávy se dělí na dvě základní skupiny. První je určena k hlášení chyb, pro informování o nějakém nestandardním stavu při doručování IP datagramů (*error-reporting messages*). Druhá skupina je určena dotazování, typicky pak k testování konektivity (*query messages*).

Vybrané typy ICMP zpráv shrnuje **Tab. 8**.

Tab. 8: Vybrané typy ICMP zpráv

Kategorie	Typ	Zpráva
Hlášení chyb	3	nedoručitelný IP datagram (*destination unreachable*)
	4	snížení rychlosti odesílání (*source quench*)
	5	přesměrování (*redirection*)
	11	vypršení doby života (*time exceeded*)
	12	problém s parametry (*parameter problem*)
Dotazování	8	žádost na odpověď (*echo request*)
	0	odpověď na žádost o odezvu (*echo reply*)
	13	požadavek na časové razítko (*timestamp request*)
	14	odpověď na časové razítko (*timestamp reply*)

ICMP zpráva má následující formát, který ukazuje **Obr. 7-32**. Pole typ rozlišuje základní typ ICMP zprávy, část kód je využita ke specifikaci důvodu použití konkrétního typu či bližší specifikaci typu. Kontrolní součet je počítán z celé ICMP zprávy včetně záhlaví.

Bity 0-7	8-15	16-31
Typ	**Kód**	**Kontrolní součet**
Část záhlaví závislá na typu zprávy		
Datová část ICMP zprávy		

Obr. 7-32: Obecný formát ICMPv4 zprávy

7.13.2 Vybrané typy zpráv pro hlášení chyb v ICMPv4 protokolu

ICMP protokol pomáhá IP protokolu s chybami v tom duchu, že je umí hlásit, nikoliv opravovat. Následná oprava je ponechána na jiných mechanizmech. Chybová hlášení jsou vždy odesílána z místa, kde se chyba objeví a adresována původnímu zdroji paketu, kterého se chyba týká.

Existuje pět základních typů chyb, které jsou v ICMP řešeny:

- **nedoručitelný datagram** – v případě, že se paket dostane po trase na směrovač, který jej nebude dále směrovat, paket je zahozen a tato zpráva slouží k tomu, aby o tom byl informován odesílatel. Důvodem vzniku této situace může být např. to, že směrovač neví, kam má paket dále směrovat, nelze jej dále směrovat např. v souvislosti s fragmentací nebo bezpečnostními pravidly.
- **potřeba snížení rychlosti odesílání** – představuje jednoduchý mechanizmus určený k řízení toku a předcházení zahlcení sítě. Zpráva navíc odesílatele informuje o tom, že jeho paket byl zahozen z důvodu zahlcení. Směrovač ve stavu blížícímu se zahlcení odesílá tuto zprávu, na kterou by měl zdroj daného paketu reagovat zpomalením odesílání paketů. Fungování je problematické, protože směrovač standardně nepozná, kdo ho zahlcuje, jelikož bere každý paket jako samostatnou jednotku a nesleduje v čase, od koho je kolik paketů.
- **potřeba přesměrování** – tato zpráva je určena především pro řešení směrování ven z lokální sítě, kde se nachází více směrovačů (výchozích brán). V tomto případě směrovač paket nezahazuje, jen informuje odesílatele, že by bylo výhodnější využít jinou výchozí bránu.
- **vypršení doby života** – při každém skoku (na každém směrovači při přenosu) se snižuje hodnota TTL. Pokud dojde ke snížení na nulu, paket již není dále směrován a je zahozen. Touto zprávou směrovač informuje odesílatele, že došlo k zahození právě z tohoto důvodu.
- **problém s parametry** – pokud obsahuje záhlaví IP paketu nějakou nejednoznačnou informaci, neplatnou hodnotu, paket je zahozen a touto zprávou je odesílatel informován.

Každé chybové hlášení obsahuje v datové části záhlaví původního IP paketu, které slouží k identifikaci paketu, kterého se chyba týká. Kromě záhlaví původního paketu je v ICMP chybové zprávě uloženo i prvních 8 bajtů datové části původního paketu (typicky záhlaví UDP či TCP, tj. transportních protokolů, které slouží k bližší identifikaci).

7.13.3 Vybrané typy zpráv pro dotazování v ICMPv4 protokolu

ICMP zprávy druhé skupiny jsou určeny k diagnostice některých síťových problémů. V tomto případě je základem komunikace pouze protokol ICMP a režim dotaz-odpověď.

- **žádost o odezvu a odpověď** – slouží především k ověření, zda dvě síťové vrstvy vzdálených uzlů jsou spolu schopny komunikovat. Iniciátor komunikace odešle žádost o odezvu na IP adresu testovaného uzlu a ten, pokud k němu zpráva dorazí a není např. v souvislosti s bezpečnostními pravidly aplikováno nějaké omezení, odpoví. Nejčastější aplikací, která tyto zprávy využívá, je *ping*.
- **požadavek na časové razítko a odpověď** – je primárně určen k synchronizaci časů dvou stanic či měření zpoždění na přenosové trase v režimu RTT (*round-trip time*; tam a zpět).

Zprávu žádost o odezvu lze využít i k zobrazení informací o trase mezi dvěma uzly. Typicky je využívána aplikace jménem *traceroute* či *tracert*. Zjednodušená ukázka výstupu programu je na **Obr. 7-33**. Z výpisu je patrná přenosová trasa od zdroje k cíli, který je zadán IP adresou (217.31.205.50). U každého uzlu po trase je zobrazena doba odezvy a IP adresa. Důležité je si uvědomit, že program zobrazuje pouze uzly pracující na IP vrstvě, jak je patrné z následujícího vysvětlení.

Technické řešení spočívá ve využití zpráv žádost o odezvu v kombinaci s vhodným nastavením hodnoty TTL v záhlaví IP paketu. Stanice zašle žádost o odezvu cílové stanice (217.31.205.50), avšak TTL nastaví nejdříve pouze na 1. První směrovač po trase proto paket zahodí a zareaguje chybovou zprávou o vypršení časovače[33]. Tím se odesílatel dozví adresu prvního směrovače. Následně odešle znovu žádost o odezvu cílové stanice (217.31.205.50), avšak TTL zvýší na 2. Tím se stane to, že paket projde prvním směrovačem a k jeho zahození dojde až na druhém směrovači směrem k adresátovi. Tento směrovač informuje odesílatele o zahození stejným způsobem, jako to udělal ten první a my nyní známe druhý směrovač po trase. Obdobně mechanizmus postupuje i dále, dokud se nepodaří získat odezvu od cílové stanice. Aby byly výsledky časové odezvy reprezentativnější, zpravidla se žádost o odezvu s jedním nastavením TTL odesílá vícekrát, což však v ukázce není zahrnuto.

```
C:\>tracert 217.31.205.50

Výpis trasy k 217.31.205.50

  1     1 ms     147.229.146.1
  2     1 ms     147.229.252.137
  3     1 ms     147.229.252.201
  4     1 ms     147.229.253.233
  5     1 ms     147.229.252.17
  6     4 ms     91.210.16.13
  7     4 ms     217.31.205.50

Trasování bylo dokončeno.
```

Obr. 7-33: Zjednodušená ukázka výstupu programu *tracert* pro zobrazení přenosové trasy

[33] Ale pouze tehdy, pokud to není z bezpečnostních důvodů zakázáno.

7.14 Internet Protocol verze 6 (IPv6)

7.14.1 Motivace zavádění nového protokolu

Rozšiřitelnost sítí podle budoucích požadavků a vzrůstající počet zařízení s potřebou konektivity (v posledních letech zejména mobilních) vyžaduje dostatek IP adres a také vylepšení dalších parametrů síťové vrstvy. Od poloviny 90. let a později zvolna vyvstával problém s budoucím vyčerpáním adresního prostoru[34] při použití protokolu IPv4 a začalo se uvažovat o náhradě. Dalšími problémy, které postupně narůstají, pak jsou rozsah internetových směrovacích tabulek a neexistence skutečného end-to-end modelu komunikace (kvůli technice NAT a jejímu velkému rozšíření).

Aktivity spojené s řešením těchto problémů vyústily v IP protokol verze 6 (IPv6). IPv6 není jen novým protokolem síťové vrstvy, ale celou sadou protokolů, podobně jako původní TCP/IP, s tím rozdílem, že IPv6 řeší především úkoly síťové vrstvy. IPv6 kombinuje zvýšené množství adres s efektivnějším záhlavím protokolu.

Již na začátek je důležité konstatovat, že v současné době stále dominantní IPv4, není v ohrožení v tom smyslu, že by nebylo aktuální. Bude koexistovat s IPv6 a lze pouze předpokládat, že časem bude nahrazeno. **Současní a i budoucí síťoví odborníci budou nuceni pracovat s oběma protokoly.**

7.14.2 Základní vlastnosti IPv6

Kromě rozšíření adresního prostoru došlo k diskuzi nad dalšími vlastnostmi IPv6 (které je s IPv4 nekompatibilní), jako např.:

- **zjednodušení formátu záhlaví – méně povinných položek** (viz kap. 7.14.5),
- **snaha o zredukování velikosti směrovacích tabulek globální úrovně ve směrovačích,**
- malé **snížení hodnoty zpoždění při zpracování ve směrovačích** (nepřepočítává se CRC paketu, žádná fragmentace paketu v průběhu cesty),
- nové podpůrné protokoly, zejména ICMPv6,
- **jednotné adresní schéma pro celý Internet i vnitřní sítě**,
- tři druhy adres – individuální, skupinové a výběrové (unicast, multicast, anycast),
- a již zmiňované **rozšíření adresního prostoru**, z 32 bitů na 128 bitů, tedy z 2^{32} adres **na 2^{128} adres**.

7.14.3 Historie a současnost IPv4 a IPv6

Základní myšlenkou Internetu byla možnost přímočaré komunikace dvou libovolných koncových stanic. To je v současné době v IPv4 v souvislosti s masivním nasazením NATu **znesnadněno**. Ztrácí se tak přímočarost komunikace, jelikož je vždy nezbytné vytvářet nějaké pomocné prostředky, které komunikaci zprostředkují. Naproti tomu **uživatelé často využívají služby, které koncové spojení mezi stanicemi potřebují**, např. komunikační systémy pro přenos zpráv, internetová telefonie a videokonferenční systémy,

[34] V souvislosti s vyčerpáním IPv4 adres je zajímavý text a také grafy na http://www.potaroo.net/tools/ipv4/index.html.

sítě pro výměnu dat. V souvislosti se zavedením IPv6, tak bylo plánováno vrátit Internet do původně zamýšleného stavu, bez NATu, což se ale zároveň jeví jako nepříliš reálné vzhledem k tomu, jak je již NAT zakořeněn v síťových technologiích i myšlení síťových odborníků.

Reálné nasazení IPv6 probíhá již řadu let, dokončeno je jen v některých sítích a v posledních letech **se tempo nasazování zrychluje**. Přibližně do konce roku 2010 platilo, že díky mnoha vylepšením, rozšířením a úpravě managementu adresace (NAT, přísnější kritéria při přidělování) bylo IPv4 stále poměrně konkurenceschopné a uspokojovalo většinu současných požadavků.

IPv4 adresování má v sobě skryty velké rezervy. V počátcích Internetu se přidělovaly adresy po velkých blocích (třída A), které nejsou zcela využity. Navíc platí, že pouze cca 70% rozdělených adres je ve směrovacích tabulkách, tj. globálně dostupných, a je otázkou kolik z těchto teoreticky dostupných adres je reálně použito. Další velkou rezervou je experimentální třída IP adres označována jako E (rozsah od 240.0.0.0/8 po 255.0.0.0/8), která není stále využita. Nicméně i kdyby se všechny tyto adresy podařilo využít, znamenalo by to pouze oddálení problému s vyčerpáním adres, přičemž další problémy IPv4 by zůstaly nevyřešeny. K dlouho očekávanému vyčerpání IPv4 adresního prostoru (na globální úrovni) došlo počátkem roku 2011. Neznamená to však, že by neexistovaly na světě vůbec žádné volné veřejné IPv4 adresy.

Jako vše, i IPv6 přináší i **nevýhody**. Dvě poměrně nepříjemné souvisí zejména s obrovským adresním prostorem. První je, že z pohledu správce **nelze adresní prostor jedné sítě** (v rozumném čase) **testovat** a zjistit tak (ne)přítomnost určitých IPv6 adres. To může být z hlediska bezpečnosti považováno i za výhodu. Bezpečnost není primárním tématem tohoto textu, nicméně s velkým adresním rozsahem souvisí spousta nových L2 problémů. Všeobecně se předpokládá, že dlouhou dobu poběží (ať už fyzicky nebo spíše logicky) dvě paralelní sítě (IPv4 a IPv6) a všechny aspekty komunikace budou muset být řešeny dvakrát a zároveň bude náročné udržet obě tyto sítě funkční stejným způsobem tak, **aby pro koncového uživatele bylo irelevantní, zda bude komunikovat přes IPv4 nebo IPv6**. Druhou variantou je pak, že sice dílčí sítě podporují přímo pouze jeden z protokolů (např. již IPv6), avšak nějakým způsobem reflektují i existenci druhého protokolu (tedy IPv4) a existují speciální mechanizmy umožňující zprostředkování komunikace i mezi verzemi protokolů.

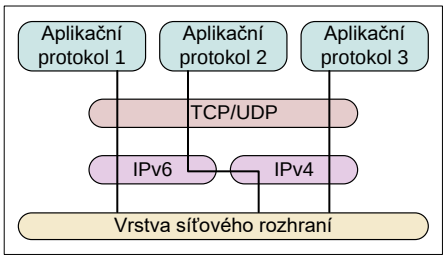
Přes všechny uvedené fakty **je IPv6 realitou**, **existují funkční** globální, regionální poskytovatelské i lokální IPv6 **sítě** a např. **v operačních systémech je podpora** již delší dobu **standardem**, což má za následek, že bez znalosti IPv6 se již neobejdeme (a to ani v čistě IPv4 síti). Velkým problémem je existence obrovského množství software a hardware vytvořeného na míru pro konkrétní použití nebo instituci, typicky bez podpory IPv6.

7.14.4 Zavádění IPv6

Základní překážkou v rychlém zavádění IPv6 je především jeho **nekompatibilita s IPv4**. Bylo proto navrženo několik mechanizmů umožňujících hladký přechod od IPv4 k protokolu IPv6. Souvisí zejména s následujícími technikami:

- **Souběh Internetových protokolů IPv6 a IPv4** (*dual stack*) – software a hardware podporuje plně oboje. To samozřejmě vede k zvýšení nákladů na vývoj zařízení, ladění a tím pádem i koncovou cenu. Na **Obr. 7-34** je pro aplikační protokol 1 použit IPv6 a naopak pro aplikační protokol 3 použit IPv4. Souběh IPv6 a IPv4 představuje jedinou smysluplnou cestu pro nejbližší roky, problémem však zůstává neustávající potřeba adres IPv4, stanice musí v tomto případě mít adresy obou typů (IPv4 i IPv6).

- **Tunelování**, tedy většinou zapouzdření IPv6 paketu do IPv4. Tunelování bylo popsáno v kap. 7.9. Technika umožňuje komunikaci přes sítě s odlišnou verzí protokolu IP. Bližší popis konkrétních mechanizmů je nad rámec tohoto textu.
- **Překlad adres** podobný technice NAT, s tím rozdílem, že při překladu se zaměňuje IPv4 adresa za IPv6 adresu nebo opačně. Obecně se technika nazývá NAT-PT (*Network Address Translator - Protocol Translator*) a její popis je opět nad rámec tohoto textu.



Obr. 7-34: Příklad na souběh IPv6 a IPv4 (*dual stack*) a také tunelování na straně hosta

7.14.5 IPv6 datagramy (pakety)

Struktura základního záhlaví IPv6 není stejná jako ta u IPv4 (viz **Obr. 7-20**). Detail jednotlivých položek je naznačen na **Obr. 7-35**, popis jednotlivých položek následuje.

- **Verze** (*version*) – 4 bity, stejně jako u IPv4 obsahuje verzi a zajišťuje, aby ostatní systémy, které zpracovávají datagram během přenosu, mohly různé pole datagramu správně použít. Verze IPv6 zde má očekávanou hodnotu 6.
- **Třída provozu** (*traffic class*) – 8 bitů, toto pole umožňuje nastavit prioritu paketu – přepravní třídu. Využití tohoto pole však ještě není přesně definováno, využití je proto minimální.
- **Identifikace toku dat** (*flow label*) – 20 bitů, označení toku dat, umožňuje zjednodušení směrování, experimentální záležitost.
- **Celková délka přenášených dat** (*payload length*) – 16 bitů, délka přenášených dat bez velikosti základního záhlaví. Informace je o počtu bajtů. Maximální délka tedy může být teoreticky až 64 kB.
- **Další záhlaví** (*next header*) – 8 bitů, informace o vnořeném záhlaví, typicky informace o protokolu vyšší vrstvy (často TCP nebo UDP).
- **Limit počtu skoků** (*hop limit*) – 8 bitů, odpovídá položce TTL u IPv4. Maximální počet skoků, které smí paket absolvovat, směrovače tuto hodnotu postupně dekrementují.
- **IPv6 adresa odesílatele/příjemce paketu** (*source/destination address*) – každá 128 bitů.

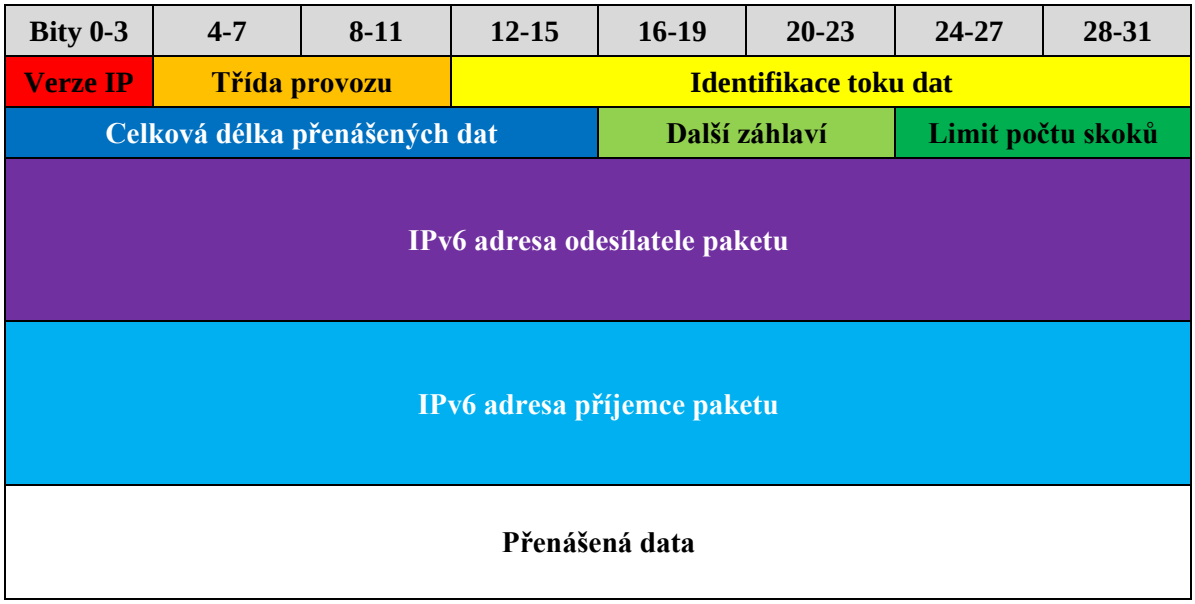
Z popisu je zřejmé, že celková délka základního záhlaví je 40 B. Tato délka je pevně dána, na rozdíl od protokolu IPv4, kde byla délka proměnná. Základní část záhlaví IPv4 má délku 20 B, což je polovina velikosti IPv6 záhlaví. Vzhledem k čtyřikrát delším adresám u IPv6 to však neznamená velký nárůst.

Malý přírůstek velikosti záhlaví je dán **vyřazením nadbytečných položek ze základního záhlaví**. Konkrétně se jedná o pole *rozšiřujících voleb, délka záhlaví, kontrolní*

součet a *fragmentace*. Některé operace lze provádět v případě potřeby pomocí rozšiřujících záhlaví.

Fragmentace je v současné době poměrně málo častý jev, který navíc komplikuje fungování směrování. Fragmentace se běžně nepředpokládá a byla odsunuta (pro speciální případy) do zvláštního rozšiřujícího záhlaví.

Kontrolní součet na IP vrstvě verze 6 již není prováděn vůbec. Výpočet a jeho kontrola v každém uzlu zbytečně zpomalovaly směrovací proces. Za dostatečnou je považována kontrola, která je standardně prováděna na spojové vrstvě. Pokud by tato kontrola nestačila, je nutné implementovat ještě další na vyšší než síťové vrstvě, což je např. u transportních protokolů běžné.



Obr. 7-35: Základní záhlaví IP datagramu verze 6

7.14.6 Adresní prostor a způsoby adresování

Sada IPv6 rozšiřuje adresní prostor na 2^{128} což je přibližně 3,4 . 10^{38}. To představuje téměř 7 . 10^{23} IPv6 adres na 1 m^2 povrchu Země (včetně moří), resp. stále těžko představitelných 7 . 10^{17} IPv6 adres na 1 mm^2 povrchu. Pro srovnání, IPv4 adres je přibližně 8 na 1 km^2 a to jen když počítáme celý rozsah, včetně adres, které jsou určeny pro speciální použití. Podstatnou změnou v IPv6 je, že jedno rozhraní běžně využívá více než jednu IPv6 adresu.

V IPv6 jsou definovány tři druhy adresování, které mají odlišné chování:

- **individuální** (*unicast*) – adresy identifikující jednotlivá síťová rozhraní, tak aby na ně mohly být zasílány pakety.
- **skupinové** (*multicast*) – jsou určeny pro adresování skupin. Platí, že pakety odeslané na tuto adresu by měly být doručeny všem členům skupiny. Tyto adresy zastupují i **všesměrové** (*broadcast*) adresy, které nejsou v rámci IPv6 definovány samostatně. V rámci adresního prostoru jsou definovány i některé speciální skupiny.

- **výběrové** (*anycast*) – také označují skupinu adresátů, rozdíl je však v tom, že pakety se posílají pouze jedinému jejímu členu, zpravidla tomu, který je „nejblíže“. Tento typ existuje i v IPv4.

7.14.7 Zápis IPv6 adres

Adresy IPv6 jsou skutečně velmi dlouhé. Jejich vhodný zápis představuje problém. Nakonec se přistoupilo k tomu, že adresy jsou zapisovány jako **8 skupin 4 hexadecimálních číslic oddělených dvojtečkou**, např.:

8000:0000:0000:0000:0ABC:DEF1:0345:789A

Každá část (mezi dvojtečkami) představuje 16 bitů. Jestliže adresa obsahuje mnoho nul (souvislý blok), je umožněno užívat zkrácený zápis:

8000::0ABC:DEF1:0345:789A

Znak „::“ se může ve zkráceném zápisu objevit pouze jednou. Navíc je definována i možnost v zápisu neuvádět úvodní nuly v každé čtveřici, tedy nejkratší možný zápis výše uvedené adresy je:

8000::ABC:DEF1:345:789A

Hexadecimální formát připomíná fyzické adresy, např. u Ethernetu je však jejich délka pouze 48 bitů.

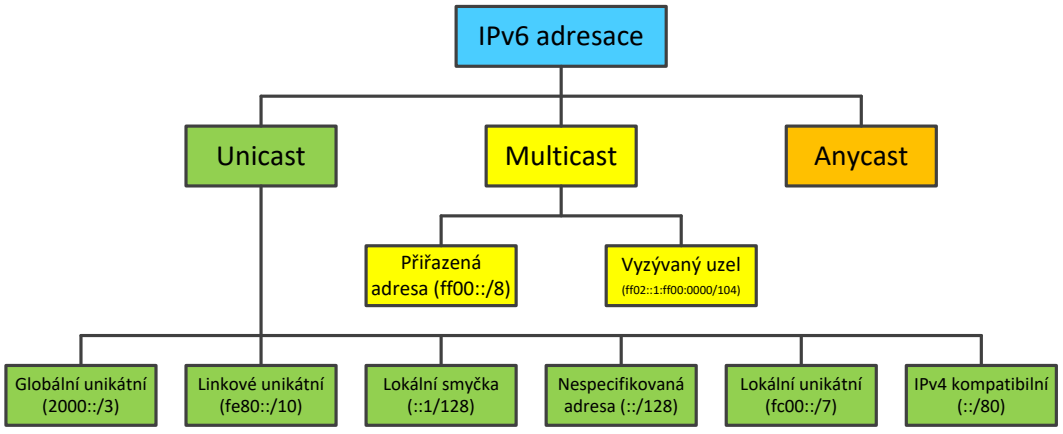
Podobně jako u IPv4 je i u IPv6 definován tzv. prefix, tj. začátek adresy, který představuje adresu sítě (nebo podsítě). Jeho délka se pak běžně zapisuje za lomítko umístěné za IP adresou, např.

8000::ABC:DEF1:345:789A / 64

znamená, že první polovina adresy je adresa sítě a zbytek je adresa stanice.

7.14.8 Typy adres

V rámci adresního rozsahu IPv6 jsou definovány speciální typy a podtypy adres. Pro ilustraci je na **Obr. 7-36** uvedeno grafické znázornění jednotlivých typů a jejich zařazení do základních kategorií.



Obr. 7-36: Grafické znázornění základních typů IPv6 adres a jejich příslušnosti do tří existujících kategorií adres

V rámci tohoto textu si v kap. 7.14.9 popíšeme pouze **globální individuální (unikátní) adresy**, což jsou adresy, které principiálně zastupují dnešní IPv4 adresy veřejného typu.

7.14.9 Globální individuální adresy

Globální individuální adresy (označované též jako globální unikátní adresy) slouží k identifikaci určitého síťového rozhraní v celém Internetu. Přidělování těchto adres mají na starost stejné organizace jako přidělování IPv4 adres, tedy IANA prostřednictvím svých regionálních zastoupení. Globální individuální adresa má kvůli maximálnímu možnému zjednodušení pevnou strukturu, která je naznačena na **Obr. 7-37**.

48 bitů	16 bitů	64 bitů
Globální směrovací prefix	**Identifikátor podsítě**	**Identifikátor rozhraní**
Veřejná topologie	Místní topologie	Lokální síť

Obr. 7-37: Struktura globální individuální adresy

Globální směrovací prefix odpovídá *de facto* adrese sítě v IPv4. Celkem těchto prefixů může být 2^{48}, tedy 2,8 . 10^{14}. Pro přiblížení, tento počet odpovídá přibližně 43 000 globálním sítím na jednoho obyvatele Země.

Identifikátor podsítě je určen k rozlišení jednotlivých podsítí v rámci celé sítě. Rozdělení na podsítě je důležité např. z pohledu rozdělení celé sítě na o něco menší a lépe spravovatelné jednotky. V rámci každé sítě může být až 2^{16} podsítí, tedy celkem 65 536 podsítí. Rozdělení na podsítě je plně v kompetenci dané organizace.

Identifikátor rozhraní slouží k odlišení koncových stanic v rámci lokální sítě. V jedné koncové podsíti pak může být až 2^{64} stanic (1,8 . 10^{19}).

7.14.10 Protokol ICMPv6

ICMP (*Internet Control Message Protocol*) je i ve verzi 6 režijním (servisním) protokolem pro IPv6. Nepřenáší žádná uživatelská data, jeho zavedení je ve všech implementacích a aktivních prvcích s podporou IPv6 povinné. Bez ICMPv6 je IPv6 nefunkční. Protokol ICMP se využívá pro ohlašování chybových stavů, testování dostupnosti síťové vrstvy a také k výměně určitých provozních informací. V rámci IPv6 se tento protokol používá také k tzv. objevování sousedů (obdoba ARP, viz kap. 7.10), podpoře správy multicastových skupin, překladu adres a zajištění mobility. Detaily jsou nad rámec tohoto textu.

7.14.11 Směrování v IPv6 sítích

O směrování a směrovacích protokolech v IPv4 sítích bylo pojednáno v kap. 7.6. Směrování v IPv6 je založeno na totožných principech, pouze se pracuje s poněkud delšími adresami. Tomu se musely přizpůsobit i směrovací protokoly, proto se prakticky u všech používaných IGP protokolů setkáváme s verzemi pro IPv6. Jsou to zejména:

- **RIPng** (*Router Information Protocol Next Generation*) – verze směrovacího protokolu RIP pro IPv6,
- **EIGRP for IPv6** (*Enhanced Interior Gateway Routing Protocol*) – směrovací protokol firmy Cisco ve verzi pro IPv6,
- **OSPFv3** (*Open Shortest Path First*) – OSPF pro IPv6 sítě,
- **IS-IS for IPv6** (*Intermediate System to Intermediate System*) – verze IS-IS pro IPv6.

V oblasti směrování mezi autonomními systémy se využívá BGP (*Border Gateway Protocol*) protokol. Pro IPv4 sítě se využívá verze 4, označovaná jako BGP4, pro IPv6 existuje modifikovaná verze označovaná jako **BGP4+**.

7.15 Zařízení síťové vrstvy

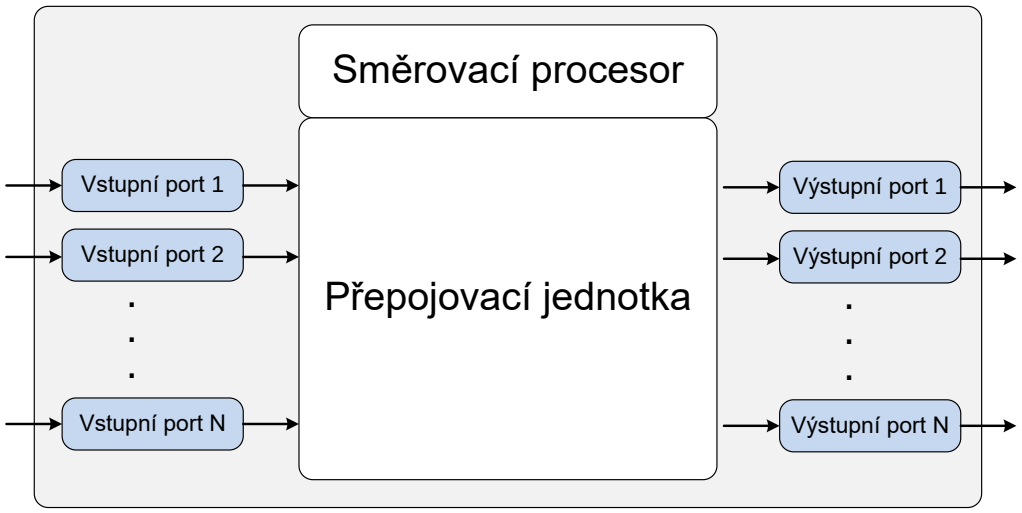
Základním zařízením síťové vrstvy je směrovač, který slouží především k propojení sítí. Směrovač pracuje zejména s pakety, které předává a doručuje podle obsahu jejich záhlaví (adresa IPv4 či IPv6 cíle). **Základní úlohou směrovače je** tedy **směrování.** Toto zařízení pracuje zpravidla se směrovacími protokoly, které mu umožňují zjišťovat směrovací informace od svých sousedů a následně vybudovat a udržovat směrovací tabulku.

Směrovač, resp. funkce směrování může být realizována primárně v hardware nebo v software. Hardwarový směrovač bývá často označován jako L3 přepínač, jak bylo uvedeno již v kap. 6.10, což může být matoucí. Směrovač může být realizován i pomocí vhodného software na běžné pracovní stanici, za předpokladu existence více síťových rozhraní a nižších požadavků na propustnost. Každý směrovač má typicky dvě a více síťových rozhraní, z nichž každé disponuje vlastní IP adresou. Směrovač pracuje vždy i s ICMP protokolem, což je situace, kdy je sám původcem zpráv (paketů s informacemi o chybě).

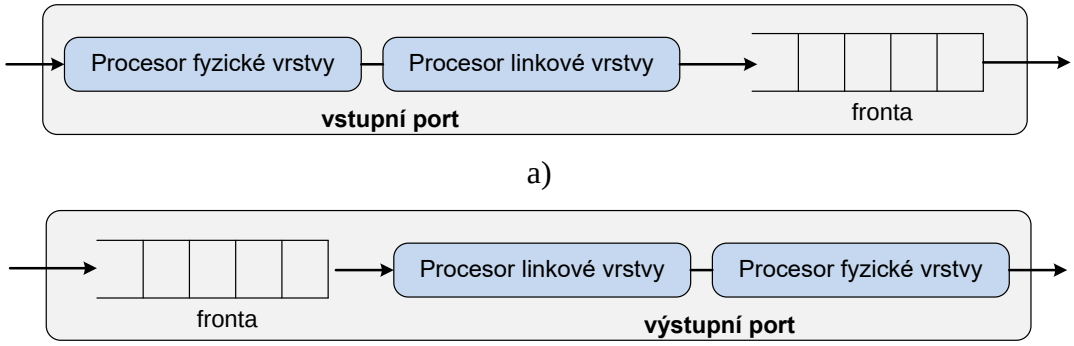
Směrovače běžně podporují i další mechanizmy jako je např. zajištění kvality služeb či určité bezpečnostní mechanizmy spočívající např. ve filtrování nežádoucího provozu.

Základní struktura směrovače je naznačena na **Obr. 7-38**. Vstupní porty slouží k přijetí paketu, výstupní porty pak k odeslání. V případě plně duplexní komunikace, jsou fyzicky vstupní a výstupní totožné, nicméně z hlediska směrování jsou to oddělené jednotky. Přepojovací jednotka se na základě řízení procesorem stará o funkci směrování. Tato část je pro funkci směrování klíčová a existuje velké množství technik, jak tento blok řešit, jejichž bližší popis je však nad rámec tohoto textu.

Každý vstupní port musí disponovat fyzickou vrstvou, spojovou vrstvou a typicky i frontou, jak je znázorněno na **Obr. 7-39**a, každý výstupní port (**Obr. 7-39**b) pak stejnými komponentami, pouze v opačném pořadí. Fronta slouží k ukládání požadavků či zpráv při přijetí nebo před odesláním do času, kdy bude možné provést zpracování či odeslání. Tvoří tak určitý vyrovnávací mechanizmus pro situace, kdy směrovač nestíhá zpracovávat všechny pakety ihned.



Obr. 7-38: Základní komponenty směrovače (blokové schéma)



Obr. 7-39: Základní bloková struktura portu směrovače: a) vstupního b) výstupního

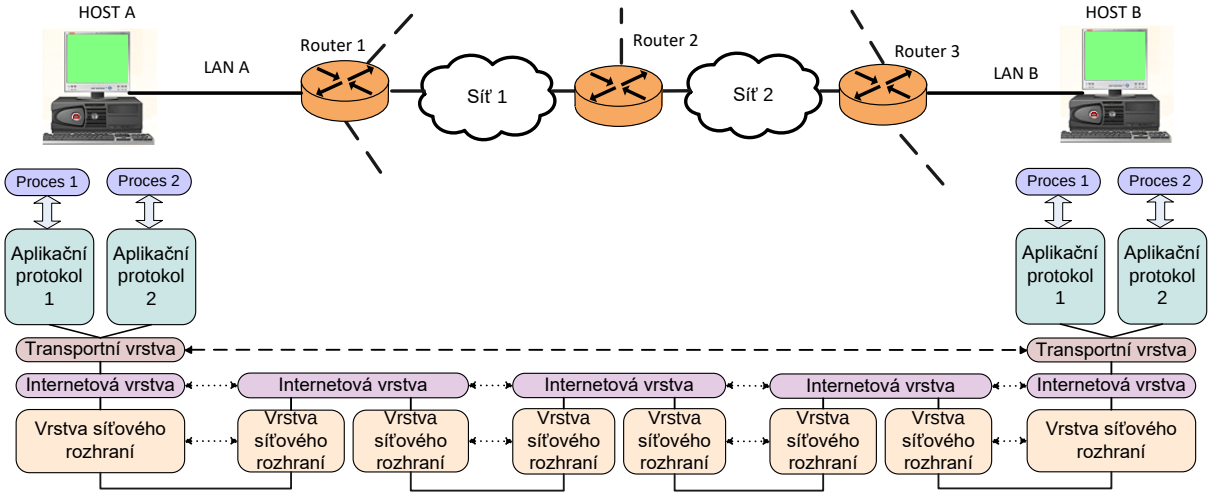
8 Transportní vrstva přenosových systémů

8.1 Služby transportní vrstvy

8.1.1 Komunikace procesů

Transportní vrstva se nachází nad síťovou vrstvou a poskytuje komunikační prostředky pro komunikaci procesů v modelu TCP/IP. Charakter komunikace je tedy již koncový (*end-to-end*). Nižší vrstvy, tvořící komunikační podsíť a především pak síťová vrstva se starají o doručení ke konkrétní stanici na vzdálené síti. Na této stanici však zpravidla běží více procesů a ty je třeba umět rozlišit. Data mají být doručena jednomu konkrétnímu procesu a zajištění této funkce je jedním z primárních úkolů transportní vrstvy. Situace je demonstrována na **Obr. 8-1**, kde jsou pro jednoduchost na každé stanici pouze dva procesy a jim odpovídající dva aplikační protokoly. Transportní vrstva tedy umožňuje vytvořit a rozlišit více komunikačních spojení mezi dvěma koncovými body či mezi jedním a více jinými body.

V případě, že bychom neuvažovali síťový model TCP/IP, ale model ISO/OSI, transportní vrstva by především zajišťovala spojení mezi dvěma relačními entitami, princip by ale zůstal stejný, tedy rozlišení koncových jednotek.



Obr. 8-1: Ukázka komunikace na transportní úrovni z hlediska identifikace procesů koncové komunikace

8.1.2 Adresování na transportní vrstvě

Z hlediska organizace komunikace existuje více možných variant, nicméně nejčastější je využití systému klient-server, tak jak byl popsán v kap. 3.11. Proces žádající nějaké služby z běžné stanice je nazýván klient a partnerský proces na straně vzdálené stanice, který služby

poskytuje, pak server. Důležité jsou z tohoto pohledu celkem čtyři adresy, které je třeba umět rozlišit:

lokální host (stanice) lokální proces (aplikace)

vzdálený host (stanice) vzdálený proces (aplikace)

O adresy stanic se, jak již bylo uvedeno, stará síťová vrstva. Adresy procesů jsou pak v kompetenci vrstvy transportní. Oba nejvýznamnější protokoly transportní vrstvy (UDP, viz kap. 8.2, i TCP, viz kap. 8.3) používají stejnou adresaci, na základě tzv. portů. Přenášená jednotka (paket) dorazí na základě síťové adresy (IPv4 či IPv6) do konkrétního počítače, je však třeba ještě nějakým způsobem odlišit, kterému aplikačnímu protokolu a následně aplikaci (www prohlížeč, emailový klient, ftp server,…) se mají přenášená data předat, viz **Obr. 8-1**. K tomuto účelu jsou určeny právě **porty** (16-bitové číslo). Z pohledu odesílající stanice je lokální proces identifikován zdrojovým portem a vzdálený proces pak cílovým portem.

Porty se dělí do třech základních skupin, viz **Tab. 9**, níže je pak uvedeno několik málo příkladů z nejdůležitější skupiny, viz **Tab. 10**.

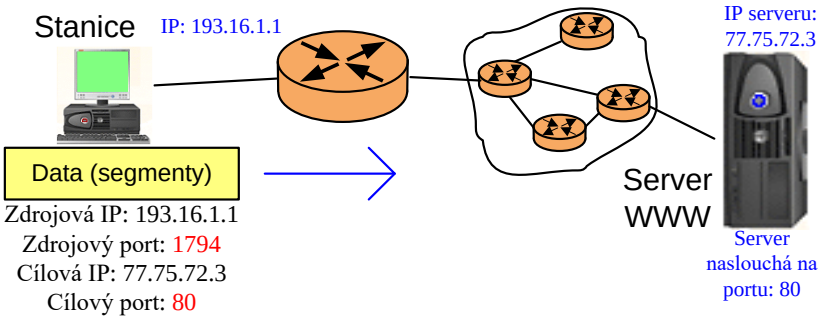
Tab. 9: Základní dělení portů

Rozsah čísel portů	Označení portů	Využití
0 – 1023	Známé (*well-known*)	Vyhrazeno pro dobře známé aplikace, číslo portu zpravidla na straně serveru
1024 – 49151	Registrované (*registered*)	Pro méně používané aplikace nebo pro porty na straně klienta při komunikaci; jejich použití je registrováno u organizace IANA
49152 – 65535	Soukromé a dynamické (*private and dynamic*)	Dynamicky přiřazované čísla portů na straně klientské aplikace

Tab. 10: Vybrané well-known porty

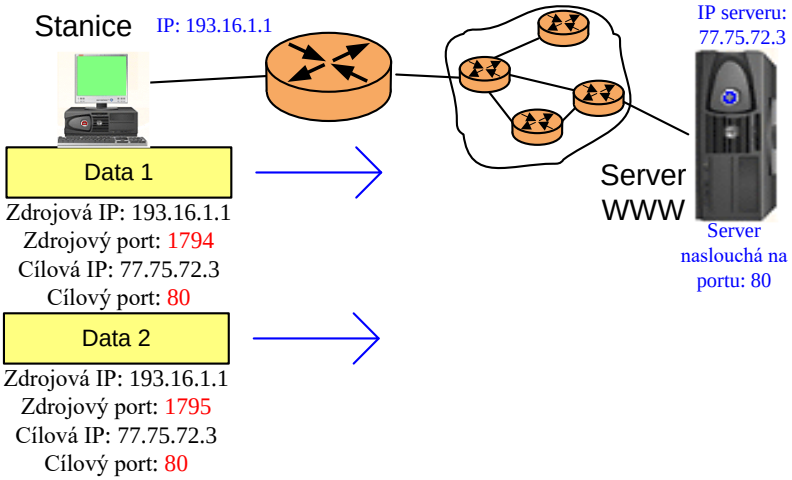
Číslo portu	Transportní protokol	Aplikační protokol
20	tcp	ftp – data
21	tcp	ftp – řízení
23	tcp	telnet
25	tcp/udp	smtp
53	tcp/udp	dns
67	udp	dhcp server
68	udp	dhcp klient
80	tcp/udp	http
443	tcp/udp	https

Jak bylo uvedeno výše, port slouží k odlišení konkrétního aplikačního protokolu, resp. přímo aplikace v počítači. Jestliže tedy máme např. spuštěný Internetový prohlížeč a připojíme se na libovolný standardně nakonfigurovaný webový server, může pár transportních adres vypadat např. tak, jak vyplývá z **Obr. 8-2**. Port na straně serveru je TCP/80, jak vyplývá i z **Tab. 10**. Port na straně stanice je náhodný, z určitého rozsahu používaného prohlížečem. Tyto porty jsou vybírány zpravidla z rozsahu označovaného jako „registrovaný“.



Obr. 8-2: Transportní adresy v případě komunikace webový prohlížeč – webový server

Jestliže na tom stejném počítači spustíme prohlížeč dvakrát a ve stejnou chvíli se připojíme na webový server z každého z nich, je nezbytné tyto dva prohlížeče nějakým způsobem odlišit. Jejich komunikace totiž probíhá paralelně. V souladu s výše uvedeným k tomu dojde na základě transportní adresy. Na straně serveru zůstává port jeden, proto se číslo portu změní na straně stanice. Situaci ilustruje **Obr. 8-3**. V tomto případě existují dvě nezávislá transportní spojení mezi danou stanicí a webovým serverem, které jsou odlišeny pouze hodnotou portu na straně stanice, konkrétně TCP/1794 a TCP/1795.



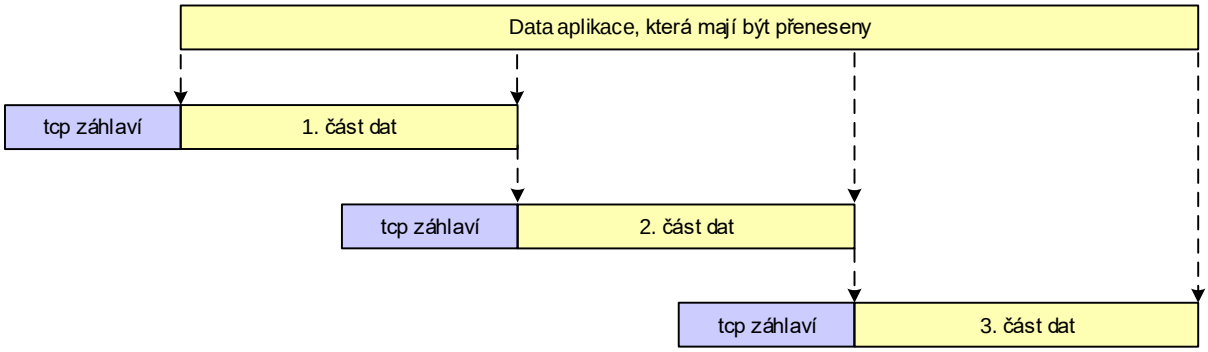
Obr. 8-3: Transportní adresy v případě dvou paralelních komunikací s jedním serverem

V souvislosti s předcházejícími příklady je vhodné zmínit pojem ***socket***, někdy též označovaný jako *Internetový socket*. Je tak označována kombinace IP adresy a portu a slouží k identifikaci koncového bodu komunikace, celá relace je pak definována dvěma sockety, tj. tím na straně odesílatele a na straně příjemce. Kombinace zdrojového a cílového *socketu* je

vždy jedinečná, tj. nikdy neexistují zároveň dvě probíhající komunikace, které by měly všechny čtyři hodnoty stejné. To je vidět i z příkladu uvedeného na **Obr. 8-3**, kde se dvě paralelní spojení liší pouze v jedné ze čtyř hodnot.

8.1.3 Zapouzdřování dat

Některé aplikace přenášejí velké množství dat a je značně nepraktické či dokonce nemožné posílat všechna tato data v jednom kuse. Daleko výhodnější je data rozdělit na menší části a tyto části pak přenášet sítí samostatně. Tomuto rozdělení se říká v TCP/IP sadě segmentace a je schematicky naznačeno na **Obr. 8-4**. Připomeňme, že jednotka na úrovni transportní vrstvy se nejčastěji nazývá *segment* (TCP) nebo *datagram* (UDP).



Obr. 8-4: Segmentace aplikačních dat na úrovni transportní vrstvy v sadě TCP/IP

Jak je patrné z **Obr. 8-4**, transportní vrstva u každé části dat přidává svoje záhlaví, což bývá běžně nazýváno jako zapouzdřování (*encapsulation*). Tato operace je vždy prováděna u odesílatele a opačná operace, rozbalení či odpouzdření (*decapsulation*), je pak prováděno až konečným příjemcem. V TCP/IP platí, že když nějaký aplikační protokol chce odeslat data, předává transportní vrstvě informaci o obou socketech (a případně i další potřebné parametry, dle vybraného transportního protokolu). Transportní protokoly mají různá záhlaví, podle toho, jaké další funkce poskytují, avšak všechny obsahují informaci o transportních adresách, tj. portech.

Segmenty opatřené záhlavím jsou následně předány síťové vrstvě, zapouzdřeny IP záhlavím a jako pakety odeslány. Na straně příjemce jsou segmenty předány transportní vrstvě, zbaveny záhlaví a na základě portu předány konkrétní aplikaci.

Nejběžnější protokoly transportní vrstvy, tedy TCP a UDP, neprovádí segmentaci stejným způsobem. TCP přidává do záhlaví pořadové číslo odesílaného bajtu (*sequence number*), takže pokud segmenty dorazí k příjemci v jiném pořadí, je možné je opětovně seřadit. UDP samo toto neumožňuje, a proto dává smysl nepoužívat tento protokol k segmentaci. Rozdělování dat na menší části je v tomto případě možné provést na nižší vrstvě (fragmentace paketu, viz. kap. 7.8).

8.1.4 Multiplexování a demultiplexování v transportní vrstvě

Transportní vrstva představuje jeden z multiplexních a demultiplexních nástrojů v rámci TCP/IP. K multiplexování dochází tehdy, jestliže se v jednom bodě střetávají požadavky z různých zdrojů a tyto mají být nějakým způsobem obslouženy. Demultiplexování je pak již pouze opačný proces.

V případě TCP/IP sady je transportní vrstva zodpovědná za vyřizování požadavků pocházejících od různých protokolů aplikační vrstvy, jak bylo znázorněno již na **Obr. 3-12** či na **Obr. 8-1**. Požadavky jednotlivých aplikačních protokolů musí být řazeny do fronty a postupně vyřizovány. K tomuto multiplexování dochází u každého z transportních protokolů a z **Obr. 3-12** je zřejmé, že k dalšímu sdružování pak logicky dochází i na úrovni síťové vrstvy.

8.1.5 Řízení přenosu v transportní vrstvě

Do problematiky řízení přenosu v transportní vrstvě patří především mechanizmy zajišťující:

- **řízení toku dat** (*flow control*) – spočívající především ve způsobu organizace komunikace mezi koncovými body, realizaci front a vyrovnávacích pamětí.
- **řízení chybových stavů** (*error control*) – vyžadující především číslování přenášených jednotek či dat a potvrzování jejich úspěšného přenosu. Řízení chybových stavů a řízení toku dat je typicky kombinováno v rámci techniky posuvného okna.
- **předcházení zahlcení** (*congestion control*) – je primárně řešeno opět pomocí techniky posuvného okna a následného nastavení dalších parametrů, např. pravidel pro opakovaný přenos či potvrzování přenosů.

Se všemi těmito mechanizmy jsme se již v nějaké formě mohli potkat i na síťové či spojové vrstvě.

Základní rozdíl oproti spojovým technikám řízení přenosu je, že řízení na transportní vrstvě má koncový charakter. Na spojové vrstvě je řešeno řízení vždy jednotlivě na konkrétní síti či trase, zatímco transportní vrstva řeší celý přenosový řetězec dohromady. Navíc platí, že ne každá spojová technologie tyto mechanizmy obsahuje.

Síťová vrstva již má z určitého pohledu i koncový charakter, avšak běžně v rámci sady TCP/IP má jen velmi omezené možnosti řízení, které jsou vysvětleny v rámci kapitoly o protokolu ICMP (kap. 7.13), či ICMPv6 (kap. 7.14.10). Tyto mechanizmy představují zejména prostředky oznamování chybových stavů, nikoliv mechanizmy řízení přenosu.

Z výše uvedených důvodů jsou mechanizmy řízení přenosu na transportní vrstvě z pohledu spolehlivosti celé komunikace klíčové. V rámci kapitoly o transportní vrstvě již nebudou v obecné rovině znovu dílčí mechanizmy rozebírány. K jejich připomenutí je možné nahlédnout do příslušných kapitol spojové a síťové vrstvy. U dvou nejvýznamnějších protokolů transportní vrstvy bude pojednáno o tom, jak k řešení této důležité problematiky přistupují.

8.1.6 Charakter poskytovaných služeb

Transportní vrstva může stejně jako dvě nižší vrstvy poskytovat služby dvou základních typů:

- **bez spojení** (*connectionless*) – aplikace potřebuje pouze rozdělit data do bloků pro transportní úroveň přiměřené velikosti a od transportní vrstvy vyžaduje pouze sekvenční odesílání těchto jednotek. Během přenosu těchto jednotek přes sítě může dojít ke změně pořadí či ztrátám. U této služby není možné implementovat mechanizmy řízení toku, řízení chybových stavů či předcházet zahlcení. Existují však aplikace, kterým tento

způsob postačuje, více viz kap. o aplikační vrstvě. Zřejmou výhodou je malá režie komunikace. Typickým zástupcem transportní vrstvy, který poskytuje služby bez spojení je protokol UDP.

- **se spojením** (*connection-oriented*) – koncové strany komunikace před vlastním přenosem navazují spojení a pouze pokud je spojení navázáno, je možné začít přenášet data, resp. segmenty. Typicky pak dochází k potvrzování úspěšnosti přenosu či opakovanému přenosu v případě chyb, úpravě rychlosti apod., tedy ke všem technikám označovaným souhrnně jako řízení přenosu v transportní vrstvě. Po provedení přenosu je spojení ukončeno. Odlišnost služby se spojením na transportní a síťové vrstvě je následující: Na rozdíl od síťové vrstvy se již transportní vrstva nezabývá fyzickými trasami paketů v síti. Služba se spojením na síťové vrstvě typicky vyžaduje spolupráci směrovačů či jiných přepojovacích prvků k vytvoření (virtuální) přenosové trasy, po které pak budou pakety přenášeny. Na transportní vrstvě jsme již nad tyto záležitosti do určité míry povzneseni, zabýváme se pouze koncovým charakterem komunikace. Pozn.: I nad síťovou vrstvou bez spojení může vzniknout transportní služba se spojením. Typickým příkladem transportního protokolu poskytujícího služby se spojením je TCP.

Všechny protokoly transportní vrstvy pracují jako plně duplexní, tzn., že umožňují komunikovat obousměrně v jeden časový okamžik. Transportní protokoly, resp. zejména protokol TCP, jsou ve skutečnosti skupinou protokolů a mechanizmů, které jsou TCP protokolem zastřešeny. Jsou to např. mechanizmy *Stop-and-wait, Go-back-N* či *Selective Repeat* a technika klouzavého okna, která byly již popsány v rámci spojové vrstvy.

8.1.7 Network and Port Address Translation

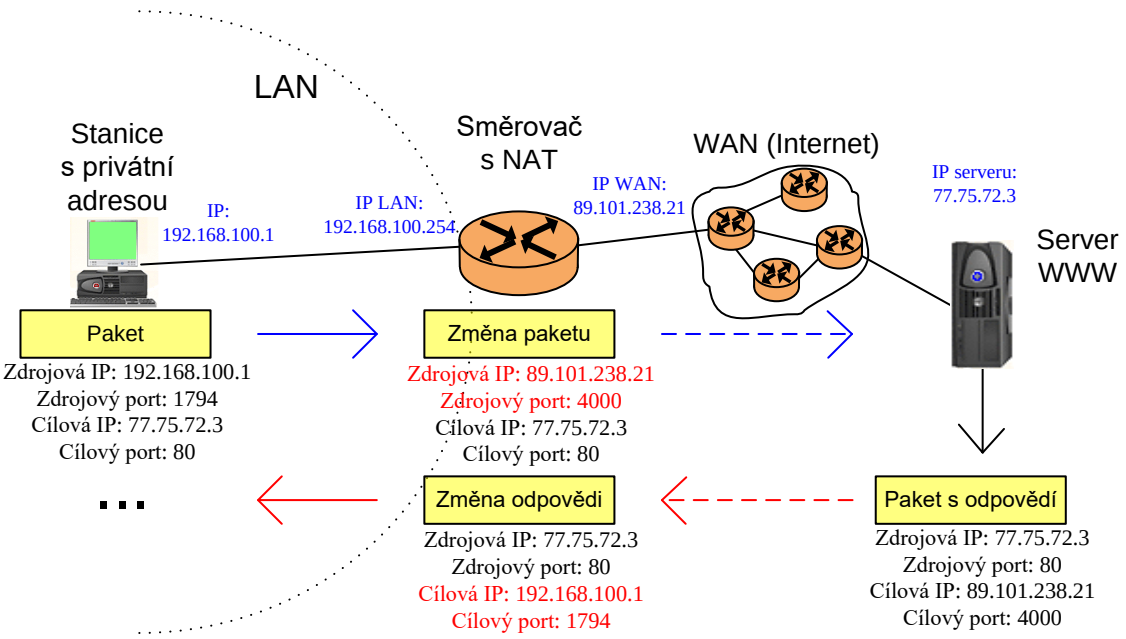
Komunikace v TCP/IP sítích vždy probíhá nejen za pomoci IP adres, ale i transportních adres (portů, viz kap. 8.1.2). V kap. 7.11 byla diskutována technika síťové úrovně, NAT. Ta spočívá v tom, že se, typicky na směrovači, zaměňuje cílová nebo zdrojová IP adresa v rámci přenášených paketů. Nejběžnější varianta pak je ta, kdy se překládá mezi privátními a veřejnými IP adresami. Privátní adresy jsou využity v rámci lokální sítě a veřejné adresy (nebo pouze jedna adresa) pak na rozhraní směrovače směrem do Internetu, obdobně, jak bylo znázorněno na **Obr. 7-29**. Na tomto obrázku byla znázorněna pouze jedna stanice v rámci lokální sítě. Při této konfiguraci je možné provádět převod 1:1, tj. pouze zaměnit jednu IP adresu za jinou a ničím dalším se nezabývat. V situaci, kdy na lokální síti bude více stanic, je však situace složitější. V zásadě máme dvě možnosti, jak provoz jednotlivých stanic při překladu odlišit:

- **překlad n:n**, u kterého musí počet adres na lokální síti (privátních) odpovídat počtu (veřejných) adres, které má směrovač k dispozici na svém odchozím portu. To je v praxi velmi málo časté, jelikož tato technika vyžaduje vysoký počet veřejných adres.
- **překlad se záměnou adres transportní úrovně**. Směrovač zaměňuje všechny vnitřní IP adresy na nižší počet veřejných (typicky pouze jednu) a aby byl schopen rozlišit provozy jednotlivých stanic, zasahuje i do transportních adres. Toto řešení je v praxi běžné a výrazným způsobem šetří adresní prostor.

Situace z na **Obr. 7-29** je znovu ilustrována na **Obr. 8-5**, tentokrát však již i s využitím překladu transportních adres. Stanice v lokální síti s adresou 192.168.100.1 se snaží o spojení s www serverem s IP adresou 77.75.72.3. Na straně stanice je vybrán port, např. 1794. Tedy máme kombinaci IP adresy a portu - 192.168.100.1:1794. Paket s žádostí o spojení dorazí na

směrovač s funkcí NAT a ten změní zdrojovou IP adresu v paketu na svoji (veřejnou), např. 89.101.238.21 a číslo portu přidělí např. takovým způsobem, že má pro stanici 192.168.100.1 vyhrazené porty 4000 – 7999 a přidělí první volný. Směrovač si uloží překladový záznam do své tabulky. Paket odchází ze směrovače do Internetu a má již jako zdrojovou adresu a port kombinaci 89.101.238.21:4000. Pokud na směrovač dorazí odpověď na kombinaci 89.101.238.21:4000, podívá se do své tabulky a na základě záznamu předá paket do vnitřní sítě na kombinaci 192.168.100.1:1794.

Název této techniky se v literatuře různí. Nejčastěji se můžeme setkat s pouze názvem NAT, který však striktně vzato pokrývá pouze překlad na úrovni síťové vrstvy. Pokud do názvu zakomponujeme i slovo port, bude pojmenování přesnější. Např. *Network and Port Address Translation* (NPAT), či *Network Address Port Translation* (NAPT), přesněji vyjadřuje, co je v rámci techniky skutečně prováděno.



Obr. 8-5: Technika NAT ve variantě Source NAT a s využitím záměny transportních adres za účelem identifikace jednotlivých koncových stanic směrovačem

8.2 User Datagram Protocol (UDP)

8.2.1 Úvod do protokolu UDP

UDP je jednoduchý transportní protokol umožňující nespojovaný (*connectionless*) a nespolehlivý (nepotvrzovaný, *unreliable*) přenos dat – v anglické literatuře označováno jako „*best effort*“. Jednotlivým přenášeným jednotkám se při použití protokolu UDP říká obvykle datagramy. Pokud je potřeba potvrzovat doručení, musí to být řešeno na úrovni aplikačního protokolu. Oproti síťové vrstvě s IP protokolem umí UDP navíc provádět přenos mezi konkrétními procesy. Ty jsou rozlišeny na základě transportních adres – portů.

Právě v jednoduchosti je však největší síla protokolu. Minimum funkcí obnáší také minimální režii přenosu a minimální zpoždění. Protokol je vhodný především na přenos krátkých zpráv, u nichž není tak kritické, pokud dojde občas k nějakému selhání. Pro přenos jednoho krátkého dotazu a následné odpovědi mohou dostačovat pouze dva datagramy, zatímco u protokolu TCP, jak uvidíme později, je to minimálně 9 jednotek.

8.2.2 Datagram protokolu UDP

Záhlaví UDP je maximálně jednoduché, jak je patrné z **Obr. 8-6**, sestává pouze ze čtyř 16bitových polí, což je dohromady 8 B, za kterými ihned následují data aplikace.

Bity 0-15	16-31
Zdrojový port	**Cílový port**
Celková délka	**Kontrolní součet**
Data aplikace	

Obr. 8-6: Záhlaví UDP protokolu

- **Zdrojový port** (*source port*) – hodnota indikuje port na straně odesílatele datagramu. Pokud je odesílatel klientem, je port vybrán z rozsahu registrovaných či dynamických portů (viz kap. 8.1.2). Pokud je odesílatel server, je číslo portu zpravidla dáno dle typu služby (viz **Tab. 10**).
- **Cílový port** (*destination port*) – hodnota indikuje port na straně příjemce datagramu, není zpravidla shodná se zdrojovým. Obdobně jako zdrojový port vychází především z toho, zda je odesílatel klient či server.
- **Celková délka** (*total length*) – hodnota reprezentuje délku celého datagramu včetně záhlaví, v bajtech.
- **Kontrolní součet** (*checksum*) – pole užito k detekci základních chyb na transportní úrovni. Určitou ochranu vůči chybám tedy UDP sice obsahuje, ale ve srovnání s mechanizmy protokolu TCP je prakticky zanedbatelná. UDP kontrolní součet je počítán nejen z UDP záhlaví a datové části, ale i části IP záhlaví paketu, do kterého bude UDP datagram později zapouzdřen, tzv. pseudozáhlaví. To v praxi vede k tomu, že funkce protokolů UDP a IP nelze zcela oddělit.

8.2.3 Služby protokolu UDP

Základní služby poskytované protokolem UDP jsou:

- **Komunikace proces-proces** – pomocí socketových adres, resp. zejména portů.
- **Přenos dat bez spojení** – každý datagram je přenášen jako samostatná jednotka, obdobně jako je tomu běžně na síťové vrstvě u paketového přenosu. Datagramy nejsou žádným způsobem číslovány, což je patrné i z formátu záhlaví. Před vlastním přenosem neprobíhá žádné navazování spojení či testování dostupnosti adresáta.
- **Žádné řízení toku dat, řízení proti zahlcení či řízení chybových stavů** – vysílač UDP datagramů může potenciálně zahltit příjemce či síť, v rámci UDP protokolu neexistují mechanizmy na řešení těchto problémů. S výjimkou kontrolního součtu v záhlaví

neobsahuje UDP ani žádné mechanizmy řízení chyb, chybových stavů či řízení přenosu jednotek.

- **Zapouzdřování a odpouzdřování dat** – tedy služba vytváření jednotek transportní úrovně na straně vysílače a následně oddělení záhlaví na straně příjemce, ale pouze tehdy, pokud není detekována žádná chyba.
- **Frontování, multiplexování a demultiplexování** – UDP protokol definuje vstupní i výstupní fronty odděleně pro jednotlivé aplikace, tj. dle adres portů. Fronty umožňují řadit požadavky a následně provádět multiplexování či demultiplexování těchto požadavků v rámci transportní vrstvy a předání do síťové vrstvy.

8.2.4 Příklady využití protokolu UDP

Při volbě transportního protokolu je třeba nalézt určité optimum. Jednoduchý a rychlý protokol je pro spoustu aplikací dostatečný či dokonce výhodný. Některé složitější mechanizmy mohou např. zbytečně zatěžovat mezilehlé uzly, trasy či konkrétní aplikace. Významným faktorem může být i zpoždění, jelikož musíme vzít vždy v potaz hodnotu RTT, která může řádově dosahovat i stovek milisekund.

Klasickým příkladem využití protokolu UDP jsou jednoduché služby typu dotaz – odpověď, např. často i *Domain Name System* (DNS), o kterém bude pojednáno v rámci aplikační vrstvy. U tohoto protokolu se klient dotazuje na IP adresy, které odpovídají běžně používaným jmenným názvům (tzv. mapování). Dotazy i odpovědi představují pouze krátké zprávy. Pokud během přenosu dojde ke ztrátě, po určité době si klientská aplikace vyžádá informaci opakovaně. Pozn.: DNS umí však využít i TCP protokol.

Dalším klasickým příkladem přenosu využívajícího UDP je technika VoIP (*Voice over IP*). Při telekomunikačním přenosu není až tak důležité, aby byl doručen úplně každý datagram. Zpravidla totiž obsahuje jen malou část slova a jeho velikost by zbytečně narostla, kdyby měl každý z nich obsahovat mnoha-položkové záhlaví protokolu. Stejně tak je zbytečné, aby se přenášel datagram opakovaně, protože čekání na opakovaný přenos by komunikaci jen zpomalilo. Pozn.: Nicméně u této aplikace je důležité pořadí jednotlivých datagramů, což však musí být řešeno až na aplikační úrovni, popř. nějakou mezi-vrstvou, která se postará o řešení těchto situací.

Současné použití UDP protokolu je výrazně širší a můžeme se s ním setkat v určitých případech např. i u přenosu větších dat (přenos webových stránek a HTTP/3). Podrobnější popis této problematiky je však nad rámec tohoto textu.

8.3 Transmission Control Protocol (TCP)

8.3.1 Služby protokolu TCP

Základní služby poskytované protokolem TCP jsou:

- **Komunikace proces-proces** – stejně jako UDP, pomocí socketových adres, resp. zejména portů.
- **Přenos toku dat** – odlišný koncept od UDP, kde probíhá přenos samostatných datagramů. TCP vytváří dojem propojení komunikujících procesů okruhem, kterým je možné přenášet tok bajtů. Pro přenos síťovou vrstvou je třeba vytvářet jednotky, které

jsou nazývány jako segmenty. Segmenty, resp. přenášené bajty jsou určitým způsobem číslovány, což je patrné i z formátu záhlaví. To umožňuje seskládat data do správného pořadí, pokud při přenosu došlo ke změnám.

- **Plně duplexní přenos dat** – je možné a běžně, že strany komunikují oběma směry zároveň.
- **Multiplexování a demultiplexování** – TCP protokol, stejně jako UDP protokol, definuje vstupní i výstupní fronty odděleně pro jednotlivé aplikace, tj. dle adres portů. Fronty umožňují řadit požadavky a následně provádět multiplex či demultiplex těchto požadavků v rámci transportní vrstvy.
- **Spojově orientovaná služba** – pokud mají být odesílána a přijímána data, musí být nejdříve uskutečněno navázání spojení. Po provedení výměny dat je každé spojení ukončeno. Tato spojení jsou pouze virtuální na úrovni transportní vrstvy. Účelem je zjištění dostupnosti druhé strany komunikace, ověření ochoty komunikovat s námi a také případné nastavení spojení.
- **Spolehlivý přenos dat** – TCP používá potvrzovací mechanizmy, které umožňují ověřit, že došlo k úspěšnému přenosu.

8.3.2 Vlastnosti protokolu TCP

Základní vlastnosti, které odlišují TCP od UDP, a které umožňují poskytovat služby popsané v předcházející kapitole, jsou:

- **Číslovací systém** – je založen na číslování odesílaných a potvrzovaných bajtů. V protokolu TCP tedy nejsou číslovány segmenty jako celky. Jelikož komunikace je obousměrná, v rámci jedné plně duplexní komunikace se vyskytuje celkem čtvero číslování (odeslané bajty jedné strany, odeslané bajty druhé strany, bajty potvrzované jednou stranou, bajty potvrzované druhou stranou).
- **Řízení toku dat** – které je založeno především na práci s velikostí okna a souvisejících mechanizmech.
- **Řízení chybových stavů** – TCP poskytuje spolehlivou službu a k tomu účelu musí obsahovat mechanizmy sledování chyb a řízení způsobů reakce na tyto chyby.
- **Řízení stavů zahlcení** – TCP dokáže pružně reagovat nejen na zahlcení na straně příjemce, ale i na zahlcení v síti. Podstatou řešení je možnost regulovat množství a rychlost odesílaných dat.

8.3.3 Segment protokolu TCP

Vzhledem k tomu, jaké funkce má protokol TCP zajišťovat, je záhlaví jednotky tohoto protokolu (segmentu) výrazně obsáhlejší než u protokolu UDP, viz **Obr. 8-7**, význam jednotlivých položek je objasněn níže.

- **Zdrojový port** (*source port*) – hodnota indikuje port na straně odesílatele segmentu, obdobně jako u UDP.
- **Cílový port** (*destination port*) – hodnota indikuje port na straně příjemce segmentu, opět obdobně jako u UDP.

- **Pořadové číslo odesílaného bajtu** (*sequence number* – **SEQ**) – pole číslování odesílaných bajtů; pole obsahuje pořadové číslo prvního z odesílaných bajtů v daném segmentu.
- **Pořadové číslo potvrzovaného bajtu** (*acknowledgment number* – **ACK**) – jestliže komunikace probíhá obousměrně (což je většina případů), tak strana, která odesílá data, má možnost v rámci záhlaví těchto dat potvrdit přijetí dat od protistrany. Uvádí se hodnota dalšího očekávaného bajtu, tj. např. když poslední správně přijatý bajt je číslován jako 100, pole obsahuje hodnotu 101.
- **Délka záhlaví** (*header length*) – délka celého záhlaví. Pole musí být uvedeno, jelikož položka „Volitelné položky záhlaví“ má proměnnou délku (0 – 40 bajtů). Jednotkou jsou řádky TCP záhlaví (32 bitů = 4 bajty).
- **Příznakové bity** (*flags*) – mohou být různě kombinovány k dosažení funkcí řízení toku, navázání či ukončení spojení apod. Význam jejich nastavení na „1“ je:
  - **URG** (*urgent*) – segment nese naléhavá data.
  - **ACK** (*acknowledgment*) – indikuje, že hodnota uvedená v poli potvrzovaného bajtu je platná (tj. že segment zároveň i potvrzuje dřívější přijetí dat z druhé strany).
  - **PSH** (*push function*) – signalizuje, že data mají být ihned po přijetí předána aplikaci a nemá se čekat na přijetí dalších segmentů.
  - **RST** (*reset the connection*) – pro řešení situace s duplikáty navazovacích segmentů, k odmítnutí spojení.
  - **SYN** (*synchronize sequence numbers*) – odesílatel začíná novou sekvenci číslování bajtů, využíváno při navazování spojení, viz dále.
  - **FIN** (*terminate the connection*) – odesílatel ukončil přenos dat, využíváno při uzavírání spojení, viz dále.
- **Délka okna** (*window size*) – vyjadřuje maximální počet bajtů, které může vysílač odeslat, aniž by čekal na potvrzení od přijímače. Hodnota se může podle potřeby měnit.

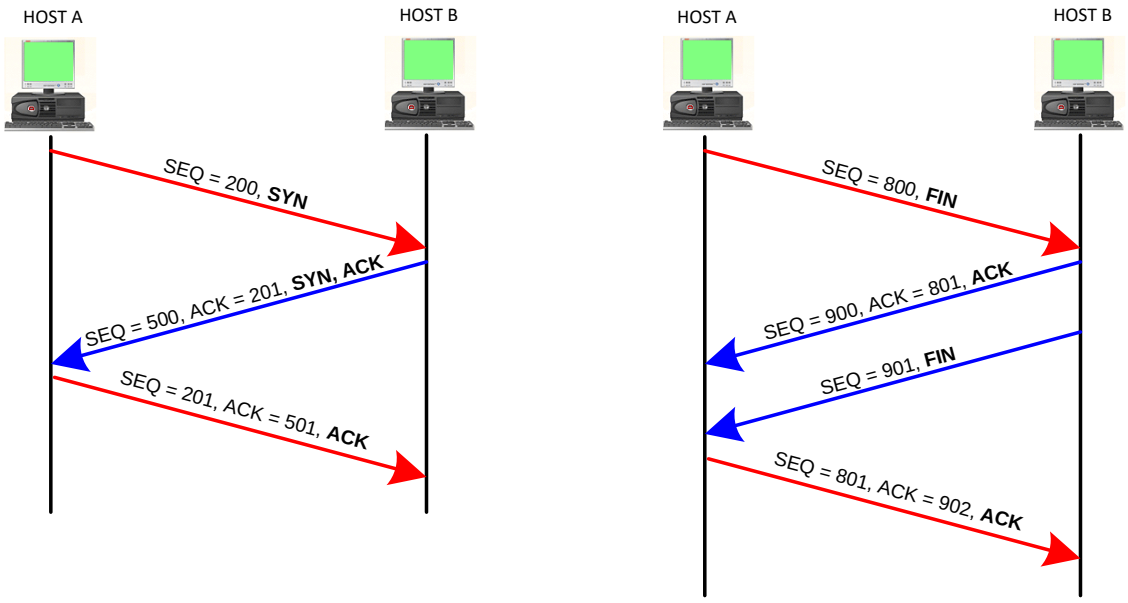
<table>
<tr><th colspan="8">Bity 0-15</th><th>16-31</th></tr>
<tr><td colspan="8">Zdrojový port</td><td>Cílový port</td></tr>
<tr><td colspan="9">Pořadové číslo odesílaného bajtu</td></tr>
<tr><td colspan="9">Pořadové číslo potvrzovaného bajtu</td></tr>
<tr><td>Délka záhlaví</td><td>Rezerva</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>Délka okna</td></tr>
<tr><td colspan="8">Kontrolní součet</td><td>Ukazatel naléhavých dat</td></tr>
<tr><td colspan="9">Volitelné položky záhlaví</td></tr>
<tr><td colspan="9">Data aplikace</td></tr>
</table>

Obr. 8-7: Struktura TCP záhlaví

- **Kontrolní součet** (*TCP checksum*) – obdobné k UDP kontrolnímu součtů. Umožňuje určitou kontrolu bezchybnosti přenosu na transportní úrovni. K výpočtu je použito TCP záhlaví, data a část záhlaví IP protokolu (pseudozáhlaví).
- **Ukazatel naléhavých dat** (*urgent pointer*) – pole vyplněno jen když je příznakový bit URG nastaven na „1“.
- **Volitelné položky záhlaví** (*options*) – pole nemusí být přítomna, jejich délku lze odvodit z celkové délky záhlaví uvedené v příslušné pozici. Jednotlivými možnostmi se v rámci tohoto textu nebudeme zabývat.

8.3.4 Navazování a ukončování spojení u protokolu TCP

Protokol TCP vytváří virtuální okruh mezi komunikujícími procesy. Jak již bylo uvedeno, protokol TCP před vlastním přenosem dat nejdříve naváže spojení, poté teprve přenáší data a nakonec spojení ukončí. Průběh navázání spojení je schematicky naznačen na **Obr. 8-8** a), ukončení spojení pak na **Obr. 8-8** b).



a) **b)**

Obr. 8-8: Průběh spojení TCP a) navázání, b) ukončení

V obrázku jsou tučně označeny významné příznakové bity, které jsou nastaveny na „1“. Navázání spojení začíná tak, že strana, která spojení iniciuje (v obrázku je to Host A), zašle segment s příznakem SYN (požadována synchronizace číslování přenášených bajtů ve směru A–>B) a v poli SEQ toto číslo nastaví. Druhá strana odpoví segmentem s nastaveným příznakem ACK (synchronizace OK, resp. potvrzení o obdržení segmentu) a zároveň taktéž bude chtít synchronizovat (nastaví SYN bit) číslování pro přenos dat v opačném směru. Toto prvotní číslo nastaví taktéž do SEQ. Jestliže Host A zahájil číslování hodnotou 200, Host B umístí do pole potvrzovaného bajtu (ACK) hodnotu dalšího očekávaného bajtu – 201. Všechny tyto údaje jsou odeslány v jednom segmentu. Host A po přijetí této zprávy následně ví, že Host B je dostupný a ochotný komunikovat. Aby dokončil proces navazování spojení, odešle třetí segment, ve kterém nastaví příznak ACK (synchronizace pro zpětný přenos OK). Do pole potvrzovaného bajtu dá číslo 501, čímž potvrdí příjem bajtu číslo 500 (číslování

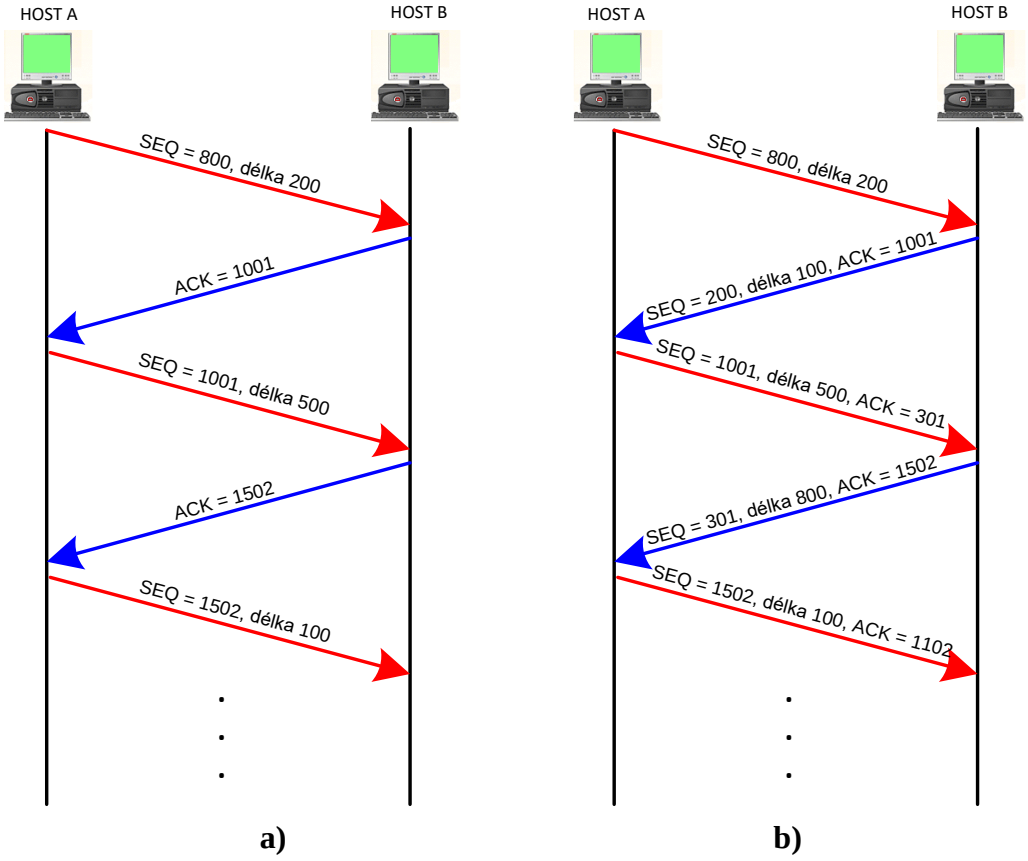
Hosta B začalo např. od 500). V rámci třetí zprávy je opět i číslo SEQ, které vždy vzroste alespoň o 1. Vzniklá jednotka se opět odešle a spojení je nyní navázáno. Tomuto způsobu navázání obousměrného **spojení** se říká **three-way handshake** (třícestné podání rukou). **Zkrácený zápis této komunikace, s kterým je možné se potkat, zní: [SYN] > [SYN, ACK] > [ACK]**.

Princip **ukončení** spojení je podobný, na obrázku je naznačen nejobecnější způsob, tzv. **four-way handshake** (čtyřcestné podání rukou), zkráceně zapsáno jako **[FIN] > [ACK], [FIN] > [ACK]**. Spojení je v tomto případě v každém směru ukončováno zvlášť. Existuje však i zkrácená verze ukončení spojení, three-way handshake, tj. zprávy: [FIN] > [FIN, ACK] > [ACK], které spočívá v tom, že zprávy FIN a ACK ze strany Hosta B jsou sloučeny do jedné.

Pozn.: Do navazování a ukončování spojení může jedna z komunikujících stran promluvit ještě bitem RST, který by měl způsobit znovu-navázání spojení. Podrobnější analýza je však nad rámec tohoto textu.

8.3.5 Průběh komunikace u protokolu TCP

Komunikace, tj. výměna zpráv, může v případě protokolu TCP probíhat až po navázání spojení, popsaném v předcházející kapitole. Po ukončení výměny zpráv dojde k uzavření spojení, jak bylo popsáno tamtéž. Obecně může při použití protokolu TCP probíhat komunikace jednosměrně nebo obousměrně. V praxi se však jednosměrná komunikace vyskytuje mnohem méně než obousměrná. Příklady obou typů je možné nalézt na **Obr. 8-9a** a **Obr. 8-9b**.



a) **b)**

Obr. 8-9: Příklad **a)** jednosměrné a **b)** obousměrné komunikace s využitím protokolu TCP

Jak již bylo uvedeno, bajty zprávy jsou při komunikaci číslovány. Hodnota prvního bajtu se uvede do pole SEQ a dále je důležitá délka přijatých dat, na základě které přijímací stanice vyšle potvrzující zprávou ACK, jejíž hodnota je nastavena na v pořadí další očekávaný bajt. Na **Obr. 8-9a** se přenáší data pouze ve směru od Hosta A k Hostu B, Host B tedy vždy data jen potvrzuje. Na **Obr. 8-9b** se přenáší data obousměrně, což znamená, že existují dvě nezávislá pořadová čísla prvního odesílaného bajtu (SEQ), pro každý směr komunikace zvlášť a taktéž dvě nezávislá čísla potvrzovaného bajtu (ACK).

V rámci obou obrázků je komunikace ukázána v nejjednodušší formě, kdy se strany ve vysílání segmentů střídají vždy pouze po jednom odeslaném segmentu. To ve své podstatě reprezentuje metodu *stop-and-wait*, která je, jak již bylo popsáno v kap. 6.7.2, velmi neefektivní. V praxi proto počty po sobě odeslaných segmentů, resp. dat, a způsob potvrzení závisí zejména na mechanizmu velikosti okna, který je stručně popsán v následující kapitole.

8.3.6 Velikost okna u protokolu TCP a návaznost na řízení provozu

Protokol TCP poskytuje mechanizmy pro řízení toku dat, čímž se napomáhá celkové spolehlivosti přenosu. Záhlaví TCP obsahuje pole *délka okna*, které definuje kolik je povoleno odeslat bajtů bez čekání na potvrzení. Toto pole umožňuje, aby přijímač nastavil, kolik mu vysílač může maximálně odeslat bajtů, aniž by dostal zpátky potvrzení a povolení k odesílání dalších dat. Jelikož přenos je zpravidla plně duplexní, okna jsou vždy dvě a nemusí být stejně velká. Nedochází tak ke zbytečnému zahlcení a zahazování dat, které přijímač nestihne zpracovat.

Problematika fungování tohoto mechanizmu (nazýván technika posuvného okna) již byla principiálně popsána v rámci spojové vrstvy. Důležité je vědět, že tento mechanizmus slouží k řízení toku dat, chybových stavů i zahlcení. Detailní specifikace fungování v rámci protokolu TCP je však již nad rámec tohoto textu.

8.3.7 Příklady využití protokolu TCP

TCP je, oproti jednoduchému a rychlému protokolu UDP, poměrně robustní protokol, s čímž pak souvisí významná režie přenosu, která je nezanedbatelná, zejména při přenosu relativně malých objemů dat. TCP poskytuje mnoho funkcí, které pak již nemusí být řešeny na aplikační úrovni.

TCP je využíváno velice často např. u protokolu HTTP (*HyperText Transfer Protocol*), FTP (*File Transfer Protocol*) či SMTP (*Simple Mail Transfer Protocol*). Protokol HTTP je primárně využíván pro přenos webových stránek, protokol FTP a jeho nástupci pro přenos souborů a SMTP je jeden z protokolů pro přenos elektronické pošty. Všechny tyto protokoly potřebují spolehlivou službu, kterou jim TCP nabízí. Není přípustné, aby se např. část souboru při přenosu ztratila, nebyl proveden pokus o nápravu či aplikace o tom nebyla informována. Je však třeba vzít v potaz, že TCP není jediným řešením a v současné době se setkáváme poměrně často i s řešeními, které některé funkce podobné TCP protokolu implementují nad UDP protokolem. Příkladem je např. protokol **QUIC** (*Quick UDP Internet Connections*), jehož popis je však nad rámec tohoto textu.

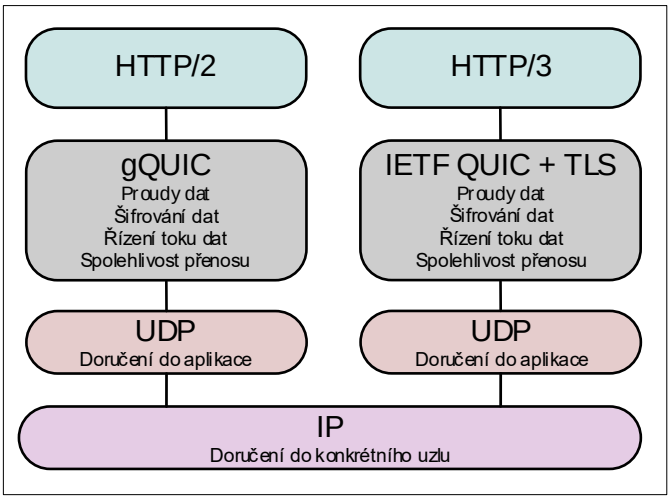
8.4 Protokol QUIC

8.4.1 Google QUIC

QUIC je v původní podobě zkratka protokolu s názvem *Quick UDP Internet Connection*, které byl vyvinut a nasazen v prohlížeči Chrome společností Google v roce 2013. Protokol v této verzi byl úzce spjat s fungováním HTTP protokolu a hlavním cílem jeho návrhu bylo, jak je možné vytušit z názvu, zrychlení komunikace mezi klientem a serverem, především při komunikaci HTTP protokolem.[35] Jak taktéž vyplývá z názvu, jedná se o protokol, který využívá ke svému fungování UDP protokol. Vytváří se tak další vrstva, která svou funkcionalitou pokrývá funkce principiálně známé z TCP protokolu a tedy řazené tradičně do transportní vrstvy. Výhodou je, že tato druhá transportní vrstva není součástí operačního systému, nýbrž konkrétní aplikace a je možné ji pružněji vyvíjet či i více přizpůsobovat danému použití.

8.4.2 Tvorba protokolu IETF QUIC

Již od roku 2016 běží snahy o standardizaci protokolu QUIC pod organizací IETF, která však v době psaní této kapitoly (8/2020) stále ještě probíhá.[36] Navrhovaná verze protokolu QUIC je označována buď pouze jako QUIC (nejedná se o zkratku) či jako IETF QUIC. Protokoly IETF QUIC a Google QUIC nejsou vzájemně kompatibilní. Zatímco protokol gQUIC byl vytvářen především pro HTTP/2, IETF QUIC je koncipován pro použití s protokolem HTTP/3 a s celou řadou dalších odlišností. Protokol IETF QUIC je úzce svázán s TLS (Transport Layer Security) protokolem, který se používá velmi často pro kryptografické zabezpečení komunikace v počítačových sítích, jak zachyceno na **Obr. 8-10**.



Obr. 8-10: Umístnění dvou základních variant protokolu QUIC do vrstvového modelu s uvedením základních úkolů jednotlivých vrstev z pohledu aplikace (HTTP)

[35] Tento protokol bývá dnes někdy označován jako gQUIC (*Google QUIC*).

[36] Viz odkaz: https://datatracker.ietf.org/doc/draft-ietf-quic-transport/?include_text=1 , kde je možné v době psaní tohoto textu nalézt 29. pracovní verzi připravovaného standardu QUIC.

8.4.3 Základní vlastnosti protokolu IETF QUIC

Protokol QUIC je v mnoha vlastnostech podobný TCP. Podobně jako TCP pracuje se spojením, které se vytváří, udržuje a ukončuje. Navíc umí během tohoto spojení přenášet více nezávislých proudů dat, označovaných jako „stream“ a rozdělovaných v rámci přenášených jednotek do podčástí označovaných jako rámec (*frame*). U protokolu QUIC je možné řídit tok dat nejen na úrovni celého spojení, ale i jednotlivých proudů.

Další základní vlastností protokolu QUIC je snaha o snížení času, který souvisí s režií navazování spojení před skutečným zahájením přenosu dat. U TCP protokolu je úvodní zpoždění před zahájením přenosu rovno dvojnásobku doby RTT (2 RTT), k čemuž musíme zpravidla následně přičíst dobu ustavení zabezpečeného (šifrovaného) kanálu, minimálně 1 RTT). Protokol QUIC při navazování spojení spojuje dohromady jak samotné navazování spojení, tak ustavení kryptografického přenosového kanálu. To vede k tomu, že po uplynutí doby 1 RTT, již může začít i přenos dat. Navíc v situaci, kdy se navazuje nedávno používané spojení znovu, počítá protokol QUIC i s tzv. přenosem dat po 0 RTT zpoždění, kdy se přenáší data již s prvním odeslaným paketem, který je primárně signálem o tom, že se obnovuje dřívější spojení s dřívějšími parametry. To může dále efektivitu zlepšit a snížit celkové zpoždění přenosu především u krátkých spojení, kde je přenášeno málo paketů.[37].

Další významnou vlastností protokolu QUIC je schopnost vyrovnat se se změnou adres na síťové či transportní vrstvě během existence spojení. To je dáno především díky tomu, že každé spojení je identifikováno nejen pomocí dvou adres socketů, ale i pomocí tzv. *Connection* ID. To je platné i v případě, že se např. změní adresa portu u odesilatele paketu.

Se zkušebními implementacemi protokolu QUIC se můžeme setkat především v prohlížečích Chrome a Firefox a u některých implementací webových serverů, avšak zpravidla je nutné tyto funkce nejdříve aktivovat v nastavení (stav k 8/2020).[38]

8.5 Další protokoly transportní vrstvy, zařízení transportní vrstvy

Celkově lze konstatovat, že TCP protokol je v současné době využíván spíše častěji než UDP, resp. u vyššího počtu běžných aplikačních protokolů. Nicméně tyto dva protokoly nejsou jediné, které jsou na transportní vrstvě definovány. Existuje dále např. SCTP (*Stream Control Transmission Protocol*), RSVP (*Resource Reservation Protocol*), RUDP (*Reliable User Datagram Protocol*), které se typicky snaží nějakým způsobem kombinovat vlastnosti TCP a UDP či přidávat nové specifické funkce. Použití těchto protokolů je ale spíše okrajové, a proto se jimi nebudeme dále zabývat.

Přímo na transportní vrstvě obvykle žádné síťové zařízení nepracuje. Jediným příkladem jsou stavové firewally, které slouží k filtrování provozu a dokáží sledovat stavy jednotlivých spojení na transportní vrstvě (typicky u TCP). Problematika bezpečnosti je však nad rámec tohoto textu, takže se těmito zařízeními (či velice často spíše software) nebudeme zabývat.

[37] Tento přístup však přináší i jisté bezpečnostní problémy, které jsou však nad rámec tohoto textu.
[38] Viz https://http3-explained.haxx.se/en/proc-status

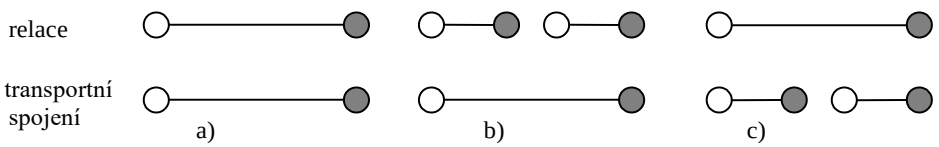
9 Relační vrstva přenosových systémů

Relační vrstva je pátou vrstvou modelu ISO/OSI. V rámci modelu TCP/IP není přímo definována a její služby, pokud jsou vyžadovány, spadají především do působnosti aplikační vrstvy, částečně pak i do transportní vrstvy. Z tohoto důvodu se budeme samostatné relační vrstvě věnovat jen minimálně.

Účelem relační vrstvy je poskytnout prostředky pro spolupráci vyšších vrstev (prezentačních), synchronizaci jejich dialogu a řízení výměny dat na této úrovni. Hlavní význam relační vrstvy lze jinými slovy popsat jako poskytování síťových služeb aplikaci. Proto u systémů, kde je jednodušší síťová struktura nebo kde je dostatečná transportní služba, nemusí tato vrstva existovat vůbec (případ TCP/IP).

Termín relace je překladem z anglického slova *session*. Relační vrstva by měla poskytovat následující služby:

- **Zřízení spojení** – vytvoření nástavby transportního spojení. Jsou možné různé režimy vzájemného promítnutí transportního a relačního spojení, viz **Obr. 9-1**, typické je však 1:1, tj. že jedno relační spojení je realizováno jedním transportním spojením. Doba života těchto dvou spojení však může být rozdílná. Možná je i situace, že jedno transportní spojení podporuje více relačních spojení či naopak.
- **Přenos dat** – spočívající nejen v přenášení dat, ale i v zajištění mechanizmů proti zahlcení daty. Spojení v rámci relační vrstvy, v rámci kterých probíhá přenos dat, mohou být simplexní, poloduplexní či plně duplexní. Relační vrstva umožňuje zpravidla rozlišovat i priority jednotlivých dat.
- **Uvolnění spojení** – buď byl skončen přenos dat a může být tak ukončeno i spojení nebo došlo k zadání požadavku na ukončení spojení bez ohledu na to, zda byl přenos dokončen či nikoliv.
- **Garantovaná služba** – podržení a zkompletování přijatých dat na úrovni relační vrstvy a jejich následné předání prezentační vrstvě (na základě požadavku).
- **Synchronizace relačního spojení** – definice a identifikace určitých synchronizačních bodů. Nastavení dohodnutého výchozího stavu relačního spojení při ztrátě synchronizace na nižší úrovni. Důležité je především udržení dialogu při ztrátě dat na transportní úrovni.
- **Řízení relační vrstvy** – spuštění, přerušení či rušení aktivit relační vrstvy, což z hlediska implementace znamená práci s procesy. Do této problematiky spadá i řešení výjimečných stavů.



Obr. 9-1: Možnosti promítnutí relačního spojení do transportního a) vazba 1:1, b) jedno transportní spojení poskytuje služby více relačním spojením, c) jedna relace využívá více transportních spojení během své existence.

10 Prezentační vrstva přenosových systémů

Prezentační vrstva představuje taktéž vrstvu, která je samostatně definována pouze v rámci modelu ISO/OSI. V rámci TCP/IP spadá tato problematika do působnosti aplikační vrstvy. Na rozdíl od relační vrstvy však platí, že prezentační funkce jsou i v TCP/IP často nezbytné a jsou tam proto řešeny.

Prezentační vrstva byla stručně popsána již v rámci kap. 3.9.2, kde bylo uvedeno, že hlavní funkce jsou:

- **prezentace dat,**
- **komprese dat,**
- **šifrování dat.**

10.1 Prezentace dat

Každý počítač může mít svoji vlastní vnitřní reprezentaci dat, tj. způsob, jak jsou jednotlivé symboly, znaky apod. ukládány. Mají-li si rozumět dva počítače, je nutno zabezpečit při jejich komunikaci provádění změn přenášených dat, aniž by se změnil význam přenášené informace. Data se tak převádí do tzv. vnější reprezentace dat.

Prezentační vrstva umožňuje překlenutí rozdílů v reprezentaci dat a činností jednotlivých entit aplikační úrovně. Problematikou prezentační vrstvy se zabývá ve své podstatě prakticky každý aplikační protokol. Veškerá data protějšího procesu jsou místnímu procesu prezentována v takové formě, kterou běžně používá. Jedná se tedy o změnu syntaxe při zachování sémantiky (významu). Důvodem používání prezentační vrstvy je především snaha o porozumění si počítačů, resp. operačních systémů. K tomu je třeba definice vnější typové reprezentace a převody mezi abecedami či formáty (pro tiskárny, zobrazovací jednotky, soubory), transformace příkazů s povely podle vlastností jejich příjemce (změna syntaxe, ale i složitější úpravy).

Přenášená data se mohou obecně vyskytovat ve třech syntaktických verzích:

- **syntaxe vysílače** – ta, která je interně používána vysílací aplikační entitou,
- **syntaxe přijímače** – ta, která je interně použita přijímací entitou,
- **přenosová syntaxe** – dojednaná syntaxe, která se používá na přenosové trase. Tato verze je referenční (prezentační) a je důležitá zejména proto, aby existovalo co nejméně různých typů převodu. Bylo by možné provádět převody i bez této verze, systém by byl ale složitější. Je možné, aby se verze příjemce či odesílatele shodovala s verzí přenosovou. V tom případě by k transformaci docházelo pouze na jedné straně komunikace.

10.1.1 Abstract Syntax Notation One (ASN.1)

ASN.1 představuje jeden ze standardních způsobů reprezentace přenášených informací, což mohou být data nebo např. řídicí pokyny. ASN.1 se používá v telekomunikacích a v různých protokolech počítačových sítí. Základní vlastností je abstrakce, která umožňuje

zápis ve formě určené pro přenos sítí, nezávisle na specifickém přístupu konkrétní platformy. ASN.1 je standard více organizací (ISO, ITU a další) vedený nejnověji pod označením X.680.

Každý datový objekt je vyjádřen pomocí tří polí – tzv. TLV (*type, length, value*) kódování.

- **Typ** – indikace konkrétní typu,
- **Délka** – délka kódované hodnoty,
- **Hodnota** – vlastní hodnota délky specifikované v předcházejícím poli.

ASN.1 nikterak neomezuje vlastní implementaci kódování a dekódování, ve skutečnosti existuje celá řada kódovacích sad, z nichž některá je pak při vlastním přenosu použita. Některé aplikační protokoly pak ASN.1 využívají především k tomu, aby popsali svoji jednotku PDU, která je využita k přenosu na aplikační úrovni. Jako příklad uveďme protokol SNMP (*Simple Network Management Protocol*), který slouží především jako prostředek k řízení sítě.

Výstupem ASN.1 bude např. posloupnost bitů, u které určitý počet bitů bude reprezentovat zvolený typ, určitý počet bitů délku dále uvedené hodnoty a následně počet bitů daný předcházející veličinu bude reprezentovat právě konkrétní hodnotu.

Jiný přístup než ASN.1 představuje např. tzv. **textový režim**, který je využíván u několika klasických Internetových aplikačních protokolů, jako je HTTP, FTP či SMTP. V tomto případě jsou přenášeny předem dohodnuté textové značky daného typu a vlastní hodnoty.

10.2 Komprese dat

Výměnu zpráv mezi procesy zprostředkovává komunikační okruh. Pro dosažení maximální propustnosti komunikačního okruhu je žádoucí předkládat zprávu pro přenos pokud možno v minimální formě. To znamená zachovat její informační obsah, ale provést její kódování tak, aby byla minimalizována její délka. Úsporu přenosové kapacity může poskytovat služba komprese dat, která spadá taktéž do prezentační vrstvy.

Základní kódy dat typicky reprezentují jednotlivé znaky zprávy určitým počtem bitů (např. 8), takže v rámci kódu je povoleno až 256 různých znaků či symbolů. Každý znak na vstupu kodéru (blok provádějící kódování) je pak na výstupu reprezentován právě jako 8 bitů. Toto kódování je však vzhledem k tomu, že pravděpodobnost výskytu různých znaků není stejná, neefektivní.

Základní metodou, která je využívána při kompresi dat, je různá délka reprezentace daného symbolu a to dle pravděpodobnosti jeho výskytu. Čím častější daný symbol je, tím více se vyplatí reprezentovat ho menším počtem bitů než ostatní méně pravděpodobné či časté symboly. Dále se k úspoře využívají také možné vazby mezi symboly, které umožňují kódovat určité kombinace jako by se jednalo o jeden znak.

Příklady základních druhů kódování sloužících ke kompresi dat jsou:

- **Huffmanovo kódování** – kóduje každý symbol nezávisle na ostatních znacích přítomných ve zprávě. Huffmanovo kódování odpovídá principiálně Morseově abecedě, kde prvek s největší pravděpodobností v anglické abecedě (e), má nejkratší délku určenou pro vysílání, tj. pouze tečka. U Huffmanova kódování tedy platí, že prvky

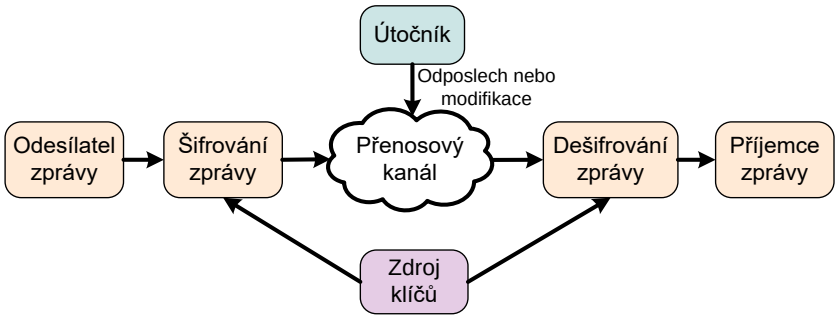
abecedy mají různé délky (různý počet bitů). Počet použitých bitů u nejméně pravděpodobného znaku je závislý na počtu možných symbolů v systému.

- **Aritmetické kódování** – je založeno na reprezentaci znaků a zejména skupin znaků reálnými čísly z rozsahu 0 až 1.
- **LZW (*Lempel–Ziv–Welch*) kódování** – představuje reprezentanta tabulkové metody, kde obě strany komunikace znají určité převodní tabulky. Často se opět využívá společná reprezentace pravděpodobných kombinací znaků, což vede ke značným úsporám.

S možností komprese dat se můžeme setkat v TCP/IP sadě u některých aplikačních protokolů, např. protokolu SSH (*Secure Shell*). Komprese dat má význam zejména u pomalých přenosových tras, kde časová náročnost provedení komprese bude malá vzhledem k úspoře času vlastního přenosu.

10.3 Šifrování dat

Šifrování (kryptografie) představuje nauku o metodách utajení obsahu (přenášené) zprávy, a to transformací do podoby, která je čitelná pouze vlastníkovi speciální znalosti. Šifrování může sloužit ale i k autentizaci nebo jiným účelům. Opakem šifrování je dešifrování. Existují dva základní způsoby šifrovaní, asymetrické a symetrické (které se liší v přístupu k šifrovacím klíčům, nicméně jejich popis je již nad rámec tohoto textu). Obecné blokové schéma kryptografického systému je možné nalézt na **Obr. 10-1**. Z obrázku je patrné, že v tomto případě předpokládáme, že útočník má přístup pouze do přenosové části, a proto musíme před vstupem do kanálu zprávu šifrovat a následně na straně příjemce dešifrovat.



Obr. 10-1: Blokové schéma kryptografického systému

Šifrování představuje taktéž metodu kódování dat, které může být obecně prováděno i na jiných vrstvách TCP/IP či ISO/OSI. V rámci TCP/IP se šifrováním často potkáváme ve formě protokolů TLS/SSL (*Transport Layer Security / Secure Socket Layer*). Tyto kryptografické protokoly umožňující zabezpečenou komunikaci. Zabezpečení je prováděno nad transportní úrovní, pro svou komplexnost se řadí do aplikační vrstvy, formálně se však jedná o prezentační vrstvu. Z výše uvedených důvodů jsou šifrovány pouze data aplikační vrstvy, nikoliv další údaje či záhlaví nižších vrstev. Šifrovaný kanál má koncový charakter, jelikož zašifrovaná data jsou přenášena přes celý komunikační řetězec od původního zdroje až k příjemci, bez zásahu mezilehlých prvků či uzlů.

Detailní popis protokolů TLS/SSL je nad rámec textu, uveďme si však, že jsou využívány např. u protokolu HTTPS (zabezpečená verze HTTP protokolu).

11 Aplikační vrstva přenosových systémů

11.1 Úvod do aplikační vrstvy

Aplikační vrstva zahrnuje ty komunikační funkce, které jsou specifické pro konkrétní aplikační procesy. Funkce aplikační vrstvy jsou proto závislé na dané aplikaci. Funkce mohou realizovat lokální nebo vzdálené programy. Vybrané funkce aplikační úrovně v obecné rovině jsou např.:

- Identifikace účastníků komunikace,
- Zjištění dostupnosti účastníka komunikace,
- Umožnění přístupu k požadovaným zdrojům,
- Stanovení metod pro opravu chyby, potvrzování,
- Bezpečnost dat,
- Management sítě.

V nejjednodušším pojetí je aplikační vrstva okno do OSI systému pro uživatelské procesy. V případě TCP/IP zahrnuje aplikační vrstva i relační a zejména prezentační funkce a služby, jak bylo popsáno v předcházejících kapitolách. Obecně byla již aplikační vrstva popsána v rámci kap. 3.9.2 (ISO/OSI) a kap. 3.10.5 (TCP/IP), kde byly uvedeny i příklady nejznámějších a nejdůležitějších protokolů:

- **HTTP** (*Hypertext Transfer Protocol*), základní přenosový protokol ve WWW (*World Wide Web*) prostředí,
- **FTP** (*File Transfer Protocol*), protokol zejména pro přenos souborů,
- **SMTP** (*Simple Mail Transfer Protocol*), hlavní protokol pro přenos elektronické pošty,
- **DNS** (*Domain Name System*), protokol pro práci se jmennými názvy (adresami) v celém Internetu,
- **DHCP** (*Dynamic Host Configuration Protocol*), protokol pro centralizovanou správu IP adres na lokální síti,
- **TELNET** (*Telecommunication network protocol*), protokol pro vzdálený terminálový přístup k jinému systému.

V rámci aplikační vrstvy existují stovky dalších protokolů, my se však budeme v rámci tohoto textu věnovat více pouze šestici výše uvedených protokolů a také se stručně zmíníme o protokolech, které slouží k realizaci **VoIP** (Voice over IP) komunikace.

11.2 Dynamic Host Configuration Protocol (DHCP)

11.2.1 Základní vlastnosti DHCP

DHCP je aplikační protokol pro dynamické nastavení parametrů sítě, primárně u koncových stanic na lokální síti. Tyto základní parametry jsou především IP adresa, maska sítě, výchozí brána a případně pak DNS servery a další parametry. Protokol funguje na principu klient-server. Podstatou je, že DHCP server tyto parametry sítě stanici na určitou dobu propůjčuje. Po této době musí stanice žádat o adresu a další parametry znovu. DHCP server u každého klienta eviduje IP adresu a čas, do kdy ji klient smí používat (doba zapůjčení, *lease time*).

DHCPv4 protokol (verze pro IPv4 sítě) je rozšířením staršího BOOTP protokolu, který přiděloval IP adresy na neomezenou dobu. DHCP je s BOOTP obousměrně kompatibilní. To znamená, že DHCP klienti dovedou získat nastavení z BOOTP serveru a DHCP server může přidělit IP adresu BOOTP klientovi (zde je třeba opatrnosti, protože BOOTP klient bude jednou přidělenou IP adresu používat už navždy).

DHCP je aplikační protokol, přestože primárně slouží síťové vrstvě. Využívá jednoduchého transportního protokolu UDP, klient komunikuje na UDP portu 68, server naslouchá na UDP portu 67.

DHCP a jeho dynamická konfigurace parametrů sítě **přináší několik výhod**:

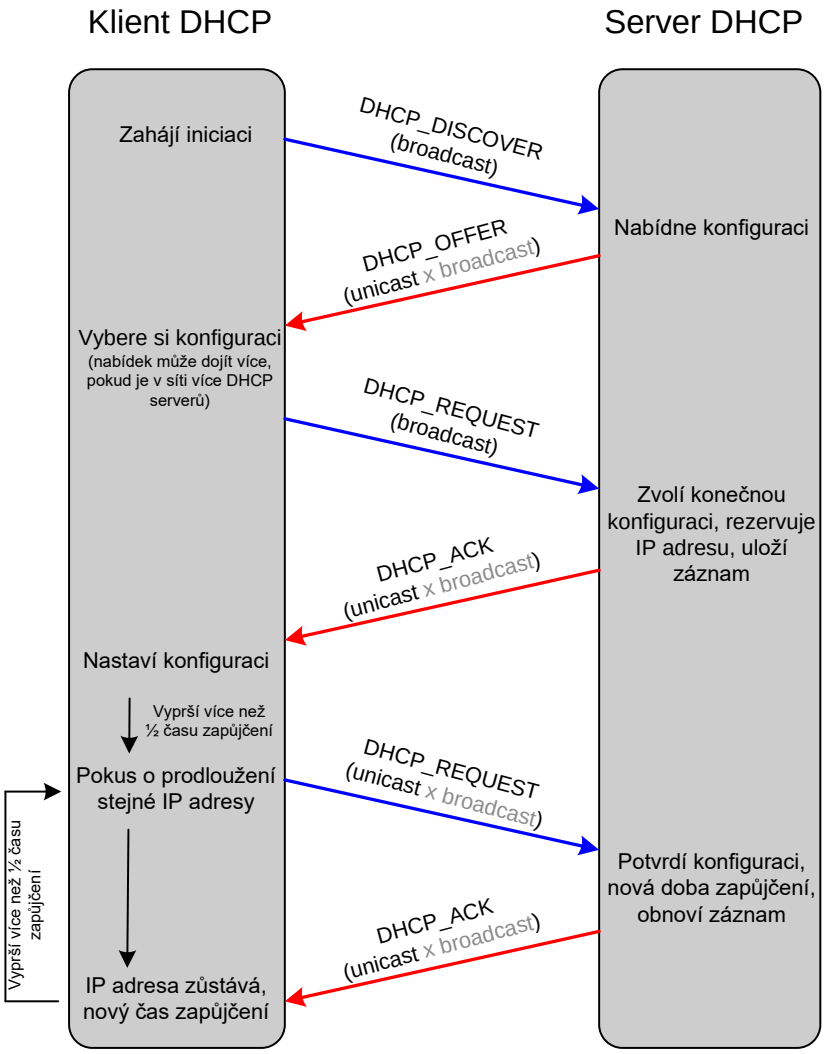
- Jednodušší správa a šetření adresního prostoru,
- Zaručuje, že se na síti nevyskytnou dvě stejné IP adresy (tzv. konflikt IP adres), což např. u ruční konfigurace parametrů sítě na každé stanici nelze snadno zaručit,
- Správce sítě může snadno „přečíslovat“ celou síť nebo změnit vlastnosti sítě s minimálním zásahem do práce uživatelů,
- Uživatelé si na stanicích v souvislosti s připojením k síti nemusí nic nastavovat, pouze musí mít povolené využití služeb DHCP. Tento protokol je standardní součástí všech operačních systémů a je ve výchozí konfiguraci povolen. Protokol umožňuje stanici na libovolné síti, kde je k dispozici DHCP server, získat potřebné parametry pro další komunikaci. Tato vlastnost je v praxi nejvýznamnější.

11.2.2 Princip činnosti DHCP

Na **Obr. 11-1** je naznačen průběh komunikace při použití protokolu DHCP. V dalším popisu budeme diskutovat pouze IP adresu, DHCP server však, jak již bylo uvedeno, poskytuje více síťových parametrů[39].

Stanice po připojení do sítě neví, kde se nachází z hlediska adresního prostoru a jakou IP adresu může využívat, aby byla schopná odesílat pakety. DHCP klient proto vyšle DHCP_DISCOVER zprávu. Tím se snaží kontaktovat DHCP server, o kterém však neví, zda na síti je a případně jakou má adresu. Proto je zpráva odeslána všesměrově, aby byla obdržena všemi uzly na dané síti, včetně DHCP serveru.

[39] Názvy zpráv odpovídají DHCPv4. Definováno je i DHCPv6, u kterého jsou principy fungování obdobné, názvy zpráv jsou však mírně odlišné. Nicméně v rámci sady IPv6 je primárně použit pro konfiguraci adres jiný mechanizmus, tzv. bezestavová automatická konfigurace, jejíž popis je však již nad rámec tohoto textu.



Obr. 11-1: Ukázka činnosti DHCP protokolu

Pokud není klient na seznamu zakázaných hostů (zpravidla identifikovaných dle fyzických adres), DHCP server odpoví zprávou DHCP_OFFER s nabídkou IP adresy. Zpráva může být odeslána již konkrétní stanici (*unicast*) nebo taktéž všesměrově všem stanicím podle nastavení určitého příznakového bitu v předchozí zprávě *discover*.

Klient si z nabídek (teoreticky od několika DHCP serverů v rámci sítě) vybere jednu IP adresu a o tu požádá paketem DHCP_REQUEST (přenášena opět všesměrově, tak aby se o výběru potenciálně dozvěděly všechny přítomné DHCP servery).

Server, který adresu nabízel, klientovi vzápětí potvrdí volbu odpovědí DHCP_ACK. Až jakmile klient obdrží zprávu DHCP_ACK, může IP adresu a zbylá nastavení používat a začít tak standardně komunikovat.

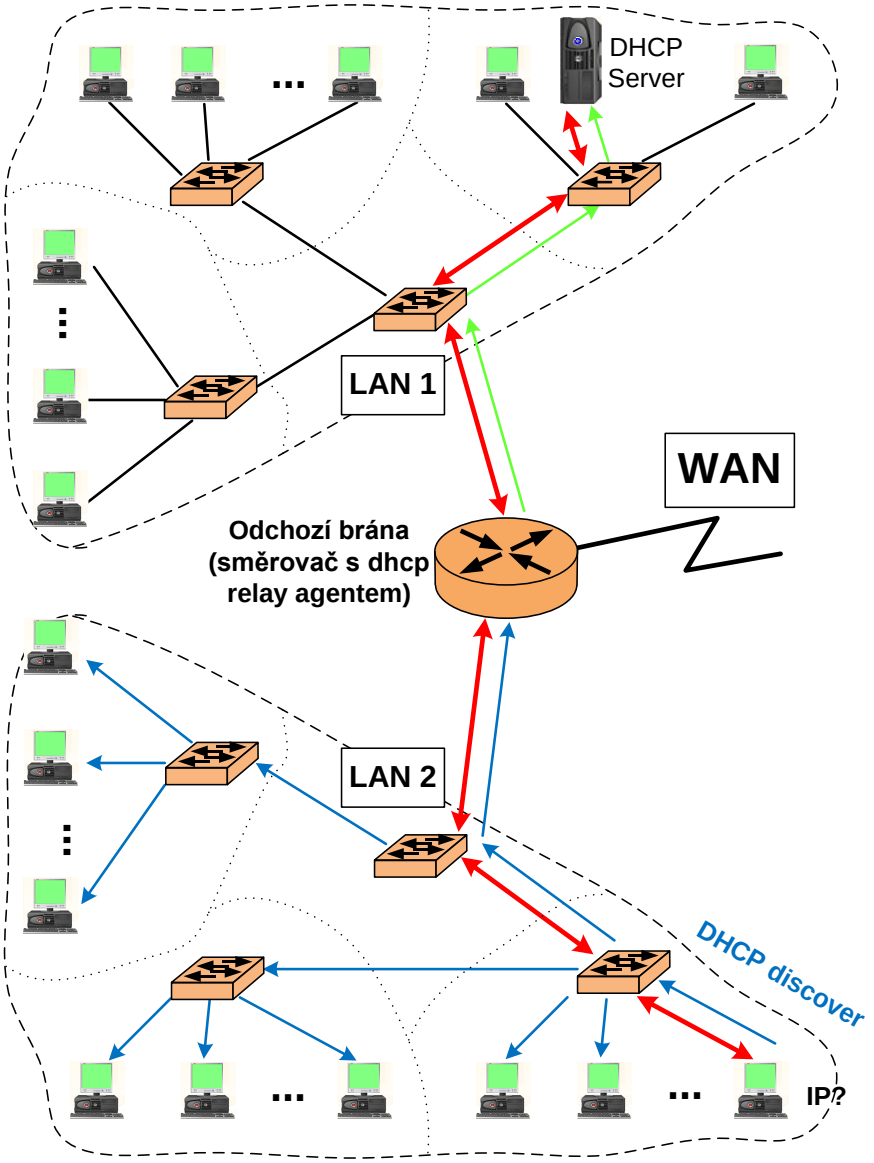
Z hlediska pružnosti systému je IP adresa přidělována jen na určitou dobu. Klient musí před uplynutím doby zapůjčení uvedené v DHCP_ACK obnovit svoji konfiguraci. To lze obecně provést stejným způsobem, jako bylo uvedeno výše. Nicméně běžně se provádí zkrácená verze komunikace, která začíná zasláním DHCP_REQUEST zprávy (běžně již konkrétnímu serveru) a server následně odpoví DHCP_ACK s novou dobou zapůjčení.

Pokud lhůta uplyne, aniž by klient dostal nové potvrzení, nesmí IP adresu dále používat. Protokol DHCP definuje ještě další typy zpráv, např. DHCP_NAK pro případy, kdy server

zamítne požadavek klienta nebo DHCP_RELEASE, která umožňuje klientovi vzdát se přidělené konfigurace (např. před korektním vypnutím systému). Další zpráva, DHCP_INFORM, slouží klientovi jako žádost o další informace, server pak požadované informace zasílá ve zprávě DHCP_ACK.

Protokol definuje i roli tzv. ***DHCP relay agenta*** (předávací agent). Používá se v situaci, kdy existují dvě nebo více sítí oddělené směrovačem a z důvodů efektivity není v každé síti samostatný DHCP server. Standardně se všesměrové DHCP zprávy nedostanou vně sítě, kde není DHCP server, a proto bez další úpravy automatická konfigurace adresy pomocí tohoto protokolu selže.

V takovém případě správce na směrovači zapne *relay agenta* a nastaví jej tak, aby všesměrové (*broadcast*) DHCP dotazy ze sítí bez DHCP serveru přeposílal do té sítě, která ho má. Agent k přeposílanému dotazu přidá informaci o síti, z které dotaz pochází, aby DHCP server věděl, ze kterého adresního rozsahu má klientovi adresu přiřadit.



Obr. 11-2: DHCP relay agent – příklad využití

Možná situace je naznačena na **Obr. 11-2**. Na obrázku jsou naznačeny dvě lokální sítě – LAN1 a LAN2, přičemž pouze v LAN1 se nachází DHCP server. Jestliže se některá ze stanic v LAN2 pokusí využít služeb protokolu DHCP a rozešle tedy všesměrově zprávu DHCP_DISCOVER, obdrží ji všechny stanice v LAN2, včetně směrovače (vyznačeno modře). V síti bez DHCP *relay agenta* by zpráva zůstala nezodpovězena, v tomto případě ji však *relay agent* přepošle DHCP serveru do LAN1 (vyznačeno zeleně). Povšimněte si, že tato zpráva již není v LAN1 šířena všesměrově, ale cíleně přímo DHCP serveru (*unicast*). Další komunikace stanice s DHCP serverem je pak vyznačena červeně. Existence *relay agenta* je pro stanici transparentní.

11.3 Domain Name System (DNS)

11.3.1 Motivace existence jmenného systému

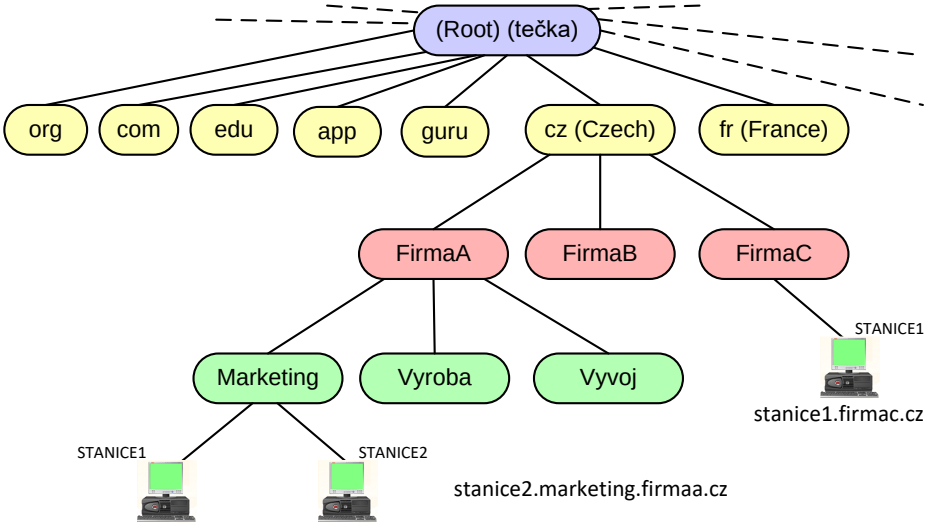
Jak již bylo uvedeno, k identifikaci koncových/transportních uzlů v Internetu slouží adresy síťové vrstvy (IPv4 či IPv6 adresy). Kdyby neexistoval systém DNS, musel by uživatel při požadavku na komunikaci např. s www nebo ftp servery zadávat jejich IP adresu. Pro běžného uživatele je prakticky nemožné pamatovat si IP adresy všech používaných serverů. To však nyní pomiňme, jistě by bylo možné nalézt systém, jak tento problém nějak vyřešit. Co však v situaci, kdyby se IP adresa serveru změnila? K tomu může dojít velmi snadno. Např. dojde k přečíslování stanic v síti anebo se celá síť, kde se nachází i uvažovaný server, přesune k jinému poskytovateli internetového připojení – dojde k fyzickému přestěhování nebo využití alternativní nabídky připojení. V tomto případě může dojít i ke změně IP adresy serveru a jeho uživatelé se to musí nějakým způsobem dozvědět, aby mohli jeho služeb i nadále využívat.

Systém DNS od těchto potíží síťové vrstvy uživatele (i stroje) osvobozuje. Zavádí jmenný systém, který můžeme chápat jako určitý systém odkazů na skutečné adresy serverů a stanic. Jestliže dojde ke změně IP adresy, stačí upravit záznam ve jmenném prostoru DNS serveru. Uživatelé se při pokusu o komunikaci se serverem díky systému DNS dozví o platné IP adrese serveru a komunikace může bez problémů proběhnout.

11.3.2 Základní popis protokolu DNS

DNS je **aplikační protokol** využívající transportní porty UDP/53 i TCP/53. Jak bylo uvedeno dříve, IP adresy představují abstrakci na úrovni síťové vrstvy. Větší počet těchto adres je však pro člověka jen těžko zapamatovatelný. **DNS** tak **vytváří** ještě **vyšší úroveň abstrakce**, konkrétně na aplikační úrovni. **Síťovým IP adresám je přiřazeno** relativně snadno zapamatovatelné **jméno** (DNS název). Výjimka z tohoto systému je zřejmá – samotný DNS server musí být zadán IP adresou, aby bylo možné s ním komunikovat. DNS primárně zajišťuje (do značné míry decentralizovaným způsobem) překlad jména hostitele (počítače) na jeho IP adresu a naopak (*reverse mapping*).

Systém je založen **na principu klient – server**, jedná se o distribuovanou datovou službu. Server DNS není pouze jeden, jsou organizovány hierarchicky, stejně jako jsou hierarchicky tvořeny názvy domén, viz **Obr. 11-3**. Vazby mezi jmény počítačů a IP adresami jsou uloženy v **DNS databázi**, která je celosvětově distribuována. Základní jednotkou systému je tzv. jmenný server (*name server*), často nazývaný DNS server, či rekurzivní resolver, viz kap. 11.3.6.



Obr. 11-3: Hierarchie DNS systému

11.3.3 Domény a doménová jména

Počítače jsou organizovány v **hierarchii domén,** viz **Obr. 11-3**. Doménou je skupina počítačů, které jsou v nějakém vztahu vůči sobě (buď tvoří nějakou organizační jednotku, nebo jsou geograficky blízko sebe). Např. doména .edu (jedna z tzv. *top-level domain* = **TLD**) je vyhrazena pro americké univerzity, naproti tomu .cz sdružuje počítače patřící (či registrované) do České republiky. V doméně se mohou vyskytovat jak koncové počítače, tak **subdomény** (FirmaA.cz), které se opět mohou dělit (FirmaA.cz –> Marketing, Vyroba, Vyvoj), kvůli lepší údržbě a snazší symbolické identifikaci. Doménové jméno se vždy vyhodnocuje **zprava doleva**, od nejvyšší úrovně (root) po nejnižší.

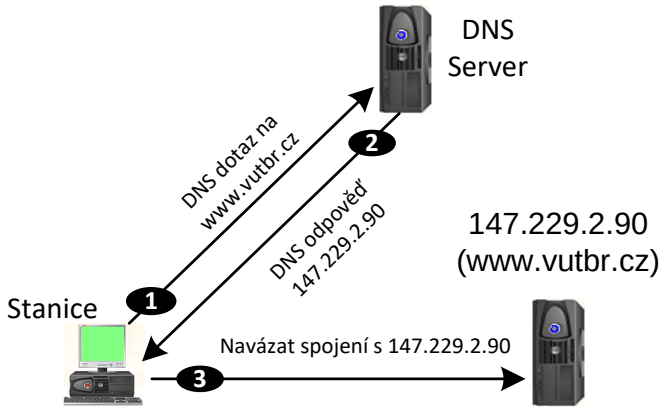
Každý uživatel Internetu dnes považuje za samozřejmé, že doménový název je tvořen několika řetězci znaků oddělených tečkami. V základním systému DNS platí, že celková délka jména může být **maximálně 255 znaků** a jeden dílčí řetězec maximálně 63 znaky. Takto dlouhá jména se však používají jen v ojedinělých případech, většinou jsou řetězce dlouhé 4 až 10 znaků (*utko, vutbr, seznam* atd.), s výjimkou domén nejvyšší úrovně, kde jsou řetězce nejčastěji dlouhé 2 až 4 znaky (*cz, int, eu, arpa* atd.). Dle první historické specifikace DNS (RFC 1034) jsou povolena **pouze písmena** (bez diakritiky a nezáleží, zda velká nebo malá), **číslice a pomlčka**. Platilo navíc pravidlo, že pomlčka nemůže být na začátku ani na konci řetězce. Dobrá implementace DNS je však schopná zvládnout libovolné 8 bitové znaky (RFC 2181), tj. prakticky libovolné symboly. Existují i rozšíření, která umožňují používat v systému DNS další znaky, zpravidla z národních abeced. Řeč je o systému IDN, jehož popis je však již nad rámec tohoto textu.

11.3.4 Základní princip komunikace v systému DNS

Ilustraci o **průběhu komunikace** s využitím DNS serveru poskytuje **Obr. 11-4**, v rámci kterého je znázorněn pouze základní princip a fungování z pohledu koncové stanice.

Stanice, která chce komunikovat s *www.vutbr.cz* vyšle dotaz na DNS server (*name server*), ten se podívá do svých záznamů a v odpovědi zašle IP adresu. Stanice pak již může přímo kontaktovat požadovaný stroj. Pokud DNS server nenalezne potřebný záznam ve své

paměti, kontaktuje nadřazený DNS server, tzv. kořenový (root) DNS server, více viz kap. 11.3.6. Tato komunikace je však klientovi systému DNS skryta a probíhá přímo mezi servery DNS.



Obr. 11-4: Komunikace s využitím DNS serveru (zjednodušeno)

11.3.5 Resolver

Ve skutečnosti je fungování překladu mírně složitější, než bylo popsáno výše. V rámci operačního systémů existuje tzv. ***stub resolver***. Je to klient, který zprostředkovává stanici (všem aplikacím – ftp, www, telnet atd.) případné dotazy na rekurzivní DNS servery. Než resolver kontaktuje DNS server, vždy prověří, zda není požadovaný překlad definován staticky[40], případně zda daný překlad již nebyl v nedávné době proveden a není uložen v lokální dočasné paměti (*cache*). Dotazování na DNS server je zpravidla vícenásobné i při jediném překladu. Vysvětlení je podáno na následujícím příkladu.

Mějme stanici, která se nachází v doméně *utko.feec.vutbr.cz*. Uživatel prostřednictvím své aplikace požádá o překlad www.mit.edu. Jestliže se v daném počítači nenalezne odpovídající záznam v souboru statických překladů ani v dočasné paměti, resolver se pokusí o překlad odesláním dotazu na DNS server. Jelikož se však počítač nachází v doméně *utko.feec.vutbr.cz*, resolver nejdříve vyzkouší, zda uživatelem zadaný řetězec „www.mit.edu“ není lokálním názvem, platným v rámci dané domény. První dotaz o překlad tedy obsahuje požadavek na překlad „www.mit.edu.utko.feec.vutbr.cz“. Server DNS (v tomto konkrétním případě) odpoví, že takové jméno se v jeho zóně nenachází. Dalším krokem resolveru je pokus o překlad v doméně vyšší úrovně, tedy *feec.vutbr.cz*. Další dotaz je tedy na překlad „www.mit.edu.feec.vutbr.cz“, v případě neúspěchu následuje ještě „www.mit.edu.vutbr.cz“. Jestliže ani na této (druhé) úrovni domény neuspěje, resolver předpokládá, že stanice s názvem „www.mit.edu“ je mimo danou doménu a zažádá čistě o překlad DNS názvu „www.mit.edu“. V tomto případě se již zpravidla dočká požadované odpovědi ve formě IP adresy. Jak tuto informaci DNS server získal, je uživateli skryto. Vysvětlení viz kap. 11.3.6.

Systému, popsaného v tomto příkladu, lze využít k usnadnění vzájemné komunikace v rámci domény. Jestliže budeme mít ve výše uvedené doméně dva počítače, jeden s celým názvem *teko.utko.feec.vutbr.cz* a druhý s názvem *adela.utko.feec.vutbr.cz*, lze s těmito

[40] V případě, že je DNS překlad definován staticky, překlad přes DNS server se neprovádí. K uložení záznamů o statickém překladu slouží soubor *hosts*, který je možné nalézt (ve Windows) ve windows složce /system32/drivers/etc. Obdobný soubor existuje i v jiných operačních systémech. Ruční zásahy do tohoto souboru lze však doporučit pouze v odůvodněných případech.

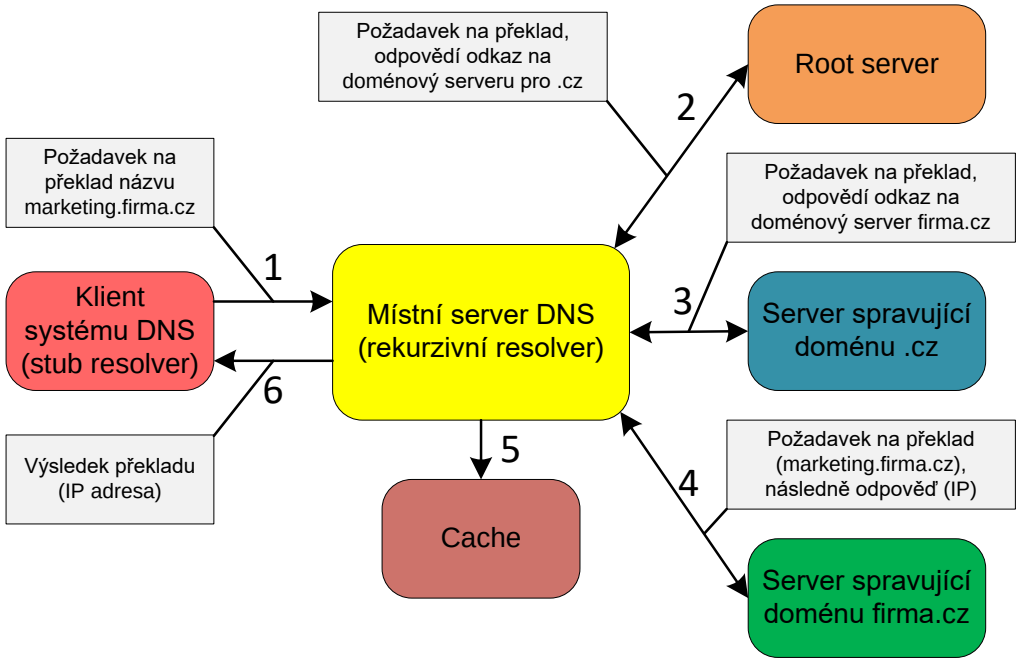
počítači (servery) komunikovat pouze pomocí názvů, platných v rámci domény, tedy *teko* a *adela*.

Na DNS serveru se také vyskytuje resolver, kterému se říká ***rekurzivní resolver***. Tento pracuje podobně jako ten na běžné stanici. Pracuje s lokální databází (a *cache*) a v případě, že při vyřizování dotazu od klienta nenalezne odpovídající záznam, kontaktuje vzdálené DNS servery, viz kap. 11.3.6. Jestliže získá odpověď, přepošle ji jako odezvu na původní dotaz klienta.

11.3.6 Hierarchie DNS serverů – kořenové DNS servery

Kořenové (*root*) DNS servery obsluhují root doménu. Tyto servery jsou využívány běžnými rekurzivními DNS servery k přesměrování na jiné (místní) doménové servery. Způsob, jakým jsou využívány kořenové DNS servery, je patrný z **Obr. 11-5**. Popis následuje.

V prvním bodě **(1)** stub resolver zformuluje dotaz a zašle ho na místní DNS server (rekurzivní resolver). DNS server zkontroluje svoje záznamy a v případě, že nalezne hledaný překlad, odpověď odešle. K tomu však v tomto příkladu nedošlo. V dalším kroku **(2)** proto místní DNS server kontaktuje některý z kořenových serverů s dotazem na překlad, přičemž bude zpravidla odkázán na doménový server nejvyšší domény[41], v příkladu .cz. Situace se opakuje, místní server kontaktuje doménový server s dotazem na překlad **(3)** a ten, pokud není sám schopen překladu, provede odkázání na doménový server druhé úrovně, který spravuje konkrétní subdoménu. Pokud tento DNS server již disponuje požadovanou informací, provede se překlad **(4)**, může však i v tomto kroku dojít na odkázání se na jiný (více zanořený) DNS server. Po obdržení informace o překladu si tento záznam místní DNS server uloží do dočasné paměti **(5)** a poskytne ji klientovi **(6)**, který celý proces inicioval.



Obr. 11-5: Systém DNS z pohledu místního serveru DNS – využití dalších DNS serverů

[41] Kořenový DNS server zná adresy všech DNS serverů nejvyšší úrovně a je schopný na ně tázající se DNS server odkázat. To je hlavní úloha kořenových serverů.

Kořenové DNS servery obsluhují nejvyšší (root) DNS zónu a jejich úkolem je především přesměrovávat dotazy na DNS servery do jednotlivých TLD domén (.com, .cz, .int, …). Jedná se o 13 serverů provozovaných 12 organizacemi. Mimo jiné společností VeriSign, některými americkými univerzitami, NASA, ICANN, RIPE NCC, americkou armádou apod. Jejich fyzické umístění je často v USA (první kopie), spíše jsou ale tyto servery distribuovány do více lokalit – ve skutečnosti serverů není 13, ale více než 1100.[42] Určitá skupina DNS serverů se vzhledem ke svým uživatelům tváří jako jeden server, je distribuována. Rekurzivní resolver komunikuje zpravidla s tím DNS serverem, který je mu z pohledu směrovacích protokolů nejblíže (*anycast*). Jiný uživatel umístěný v jiné lokalitě komunikuje s fyzicky jiným serverem, než první uživatel, přestože oba používají např. služeb serveru *M*. DNS název serverů je vždy `X.root-servers.net`, kde *X* je písmeno označující daný server, tedy *A* až *M*. Nejvíce serverů v Internetu je *L*, které provozuje organizace ICANN (viz kap. 7.5.2).

11.3.7 Typy DNS záznamů

Systém DNS je hojně využíván nejen pro adresy webových serverů, ale i pro další služby. DNS záznamy, tj. informace (nejen) o doménových jménech a odpovídajících IP adresách, jsou na serverech uloženy v záznamech nazývaných věty **RR** (*Resource Records*).

Vybrané typy vět RR jsou

- **A** (*A host address*) – 32bitová IPv4 adresa, slouží jako výsledek překladu jména na IP adresu.
- **MX** (*Mail eXchanger*) – speciální záznam pro emailové servery domény.
- **NS** (*Authoritative name server*) – doménové jméno DNS serveru, který je autoritou pro danou doménu, používá se k přesměrování na jiný DNS server.
- **AAAA** (*IPv6 address*) – 128bitová IPv6 adresa, slouží jako výsledek překladu.
- **CNAME** (*Canonical name for an alias*) – doménové aliasy (další možné názvy domény, které odkazují na stejný stroj).

Z popisu je zřejmé, že DNS systém rozlišuje např. záznamy s IPv4 adresou (A) a IPv6 adresou (AAAA). V případě, kdy resolver (DNS klient) požaduje od serveru získání informací (zjednodušeně viz **Obr. 11-4** a podrobně viz **Obr. 11-5**), pak ve skutečnosti požaduje od DNS serveru nějakou konkrétní větu RR. Typem požadavku lze tedy rozlišovat, jaký záznam server poskytne. Stanice pak mohou dotazovat i více typů, k tomu jsou však nutné samostatné dotazy. Pokud tedy stanice chce u serveru www.server.cz zjistit odpovídající IPv4 i IPv6 adresu, pošle dotaz typu A a i dotaz typu AAAA. Od DNS serveru dostane dvě odpovědi. Ne každá doména však má v databázi všechny typy záznamů, proto v některé z odpovědí může být uvedeno, že záznam neexistuje.

[42] Mapa umístění veřejně známých kořenových serverů je k dispozici na http://www.root-servers.org/ .

11.3.8 Registrace domén

V této kapitole se budeme zabývat registrací doménových jmen. Můžeme se rozhodnout náš webový server s IP adresou např. 217.11.251.230 nazývat např. www.mujserver.cz[43]. Aby se na něj však mohli lidé snadno připojovat přes název www.mujserver.cz, musí být tento název registrován v doméně, konkrétně v tomto případě v doméně .cz[44].

Je zřejmé, že registrace domén musí mít z technického hlediska určitá pravidla a musí existovat organizace, která bude systém zastřešovat. Celosvětově je to organizace IANA, o níž byla řeč v kap. 7.5.2. Pokud se posuneme do českých poměrů, tak narazíme na zájmové sdružení CZ.NIC, které v rámci ČR zastřešuje právě oblast registrace domén (mimo jiné). Sdružení zabezpečuje provoz domény .cz a provozuje registr doménových jmen. CZ.NIC samo o sobě domény zákazníkům neregistruje, to provádějí jednotliví registrátoři – členové sdružení (kterých je cca 45)[45]. Celkově již bylo v doméně .cz zaregistrováno více než 1,1 miliónu domén (stav k 6/2014).

Pro registraci domény je potřeba zaregistrovat kontakt (jména, organizace) a sady (vlastních) jmenných serverů, kam bude doménové jméno delegováno. Registrace domény je možné ověřit ve službě WHOIS (http://www.nic.cz/whois/).

11.4 Telnet

Telnet je klient-server protokol umožňující emulaci telekomunikačního terminálu (někdy označovaného jako virtuální terminál) v síti TCP/IP. Byl poprvé standardizován již v roce 1980 a tvoří tak jeden z klasických protokolů sady. Využíván je port TCP/23 (na straně serveru). Umožňuje přihlášení za účelem interaktivní práce, administraci či konfiguraci na jiném počítači (server) ze vzdáleného počítače (klient). Při přihlášení je obvykle požadována kombinace přihlašovací jméno (*login***)** a heslo.

Z dnešního pohledu je nevýhodné, že tyto údaje jsou sítí přenášeny **nešifrovaně**, a protokol proto není považován za bezpečný. Proto se v současné době nedoporučuje tento protokol používat jinde než v rámci důvěryhodné lokální sítě. Z těchto důvodů může být protokol v některých sítích administrativně blokován. Jeho plnou náhradou je protokol **SSH** (*Secure Shell*), jehož popis je však nad rámec tohoto textu.

Protokol Telnet obsahuje **podvrstvu NVT (***Network Virtual Terminal*), tedy síťový virtuální terminál. NVT se zabývá **prezentací dat**, což je funkce prezentační vrstvy (viz kap. 10.1) tj. v jaký bajt se má změnit např. znak "A", aby na druhém konci síťového spojení byl interpretován opět jako "A", či jaký příkaz protokolu Telnet se má vygenerovat při určité události.

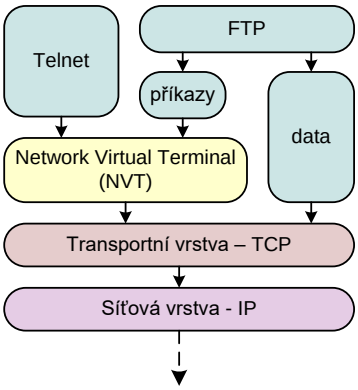
Protokol NVT je použit v omezené míře pro prezentaci dat v mnoha dalších protokolech, jako jsou např. FTP, poštovní POP3 či SMTP a HTTP. Protokol NVT jakožto určitá **prezentační vrstva** je znázorněn na **Obr. 11-6**.

---

[43] Na základě pravidel registrace a především zákonů dané země je možnost registrace určitých názvů omezena (firmy, ochranné známky, obchodní značky, slavná jména, …). U „volných" domén samozřejmě funguje přístup „kdo dřív přijde, ten dřív mele".

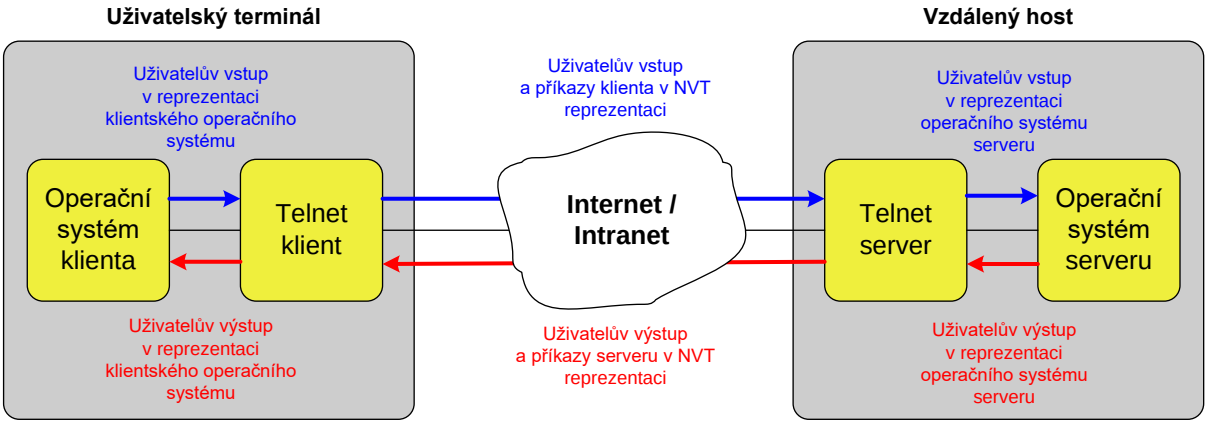
[44] Teoreticky by samozřejmě bylo možné na všech počítačích nastavit statický překlad DNS, tj. že po zadání www.mujserver.cz se má prohlížeč připojit na IP adresu 217.11.251.230, to však není příliš praktické. Ještě méně praktické pak je nemít DNS název vůbec.

[45] Aktuální seznam je k dispozici na http://www.nic.cz/whois/registrars/



Obr. 11-6: Zařazení protokolu NVT v hierarchii TCP/IP (FTP využívá NVT jen pro přenos příkazů, nikoliv dat)

Na **Obr. 11-7** je znázorněna základní filozofie práce protokolu Telnet. Klient je na jednom počítači a potřebuje pracovat i na druhém (vzdáleném) počítači. Klient spustí program Telnet s parametrem vzdáleného počítače, na který se chce připojit. Program Telnet naváže spojení protokolem TCP na port 23 vzdáleného počítače. Na tomto portu čeká server protokolu Telnet (program telnetd = *telnet daemon*). Následně probíhá komunikace tak, jak je naznačeno v obrázku.

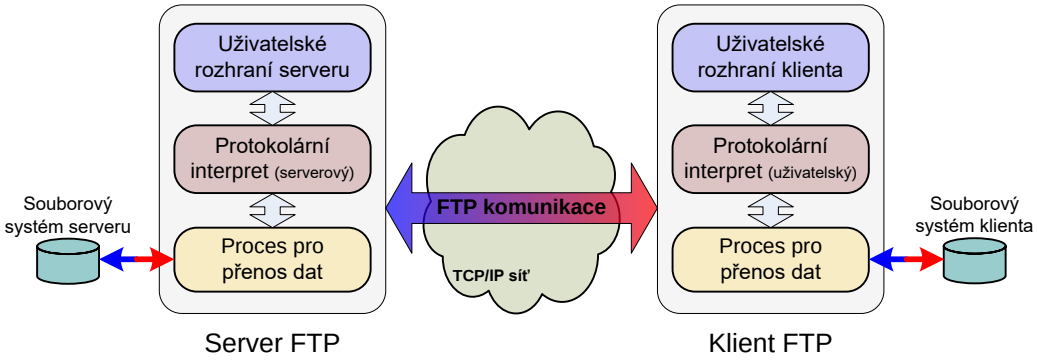


Obr. 11-7: Komunikace prostřednictvím telnetu s využitím NVT

11.5 Přenos souborů a protokol FTP

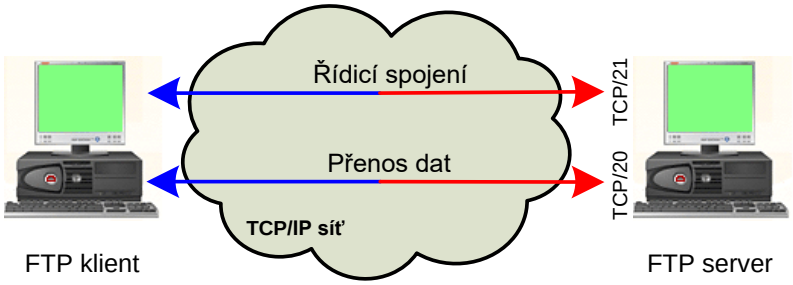
11.5.1 Základní popis protokolu

FTP (*File Transfer Protocol*) je taktéž standardním aplikačním protokolem, který slouží pro přenos souborů mezi uzly v sítích TCP/IP, na kterých mohou obecně běžet různé operační systémy. Je to jeden z nejstarších protokolů, byl definován poprvé taktéž již v roce 1980. Základní schéma systému FTP je možné nalézt na **Obr. 11-8**.



Obr. 11-8: Základní schéma systému FTP

FTP pracuje na principu **klient-server**. Unikátní je v tom, že pro přenos řídicích příkazů a přenos dat používá **dvě oddělená transportní spojení**, implicitně porty: **TCP/20 pro data a TCP/21 pro řízení** (porty na straně serveru), viz **Obr. 11-9**. Data jsou přenášena spolehlivou cestou (spojení TCP), takže data nejsou ohrožena z pohledu ztrát. Avšak z dnešního pohledu je i u tohoto protokolu nutné podotknout, že podstatnou nevýhodou je, že přenos probíhá bez jakéhokoliv kryptografického zabezpečení, které by zabraňovalo třetí osobě v odposlechu nebo změně dat.



Obr. 11-9: Způsob FTP komunikace mezi klientem a serverem

Přihlášení klienta na server může probíhat dvěma způsoby:

- **anonymně** – server nepožaduje žádnou autentizaci uživatele, resp. jako login je uvedeno klíčové slovo„anonymous“ a jako heslo zpravidla libovolná emailová adresa.
- **Zadáním konkrétního přihlašovacího jména a hesla** – oba tyto údaje se zasílají sítí zcela otevřeně a kdokoliv si je tak může zachytit.

Protokol FTP je **stavový**, tj. server klienta registruje (*user aware*) a pamatuje si o něm řadu informací, např.:

- ve kterém adresáři na serveru se momentálně nachází,
- jaký režim přenosu je nastaven, viz dále.

Řídicí spojení (*Control connection*) se serverem **navazuje vždy klient** a toto spojení je udržováno po celou dobu trvání relace. Toto spojení je jednoúčelové a slouží výlučně k výměně příkazů a odpovědí mezi oběma stranami.

Pro datové přenosy se navazují nová **datová spojení**, která jsou **jednorázová**. Okamžitě po přenesení souboru, výpisu adresáře apod. se toto datové spojení ukončí.

11.5.2 Základy komunikace klienta se serverem

Klient posílá po řídicím spojení FTP příkazy (*FTP commands*), k interpretaci příkazů se využívá NVT protokol, jak bylo uvedeno v kap. 11.4. Obvyklé jsou dva tvary zpráv:

`PŘÍKAZ` nebo `PŘÍKAZ parametry`

Daný `PŘÍKAZ` jsou 3 až 4 textové znaky identifikující konkrétní FTP příkaz (pokyn či operaci). Seznam těchto příkazů je poměrně dlouhý.

Server posílá po řídicím spojení FTP odpovědi (*FTP replies*), rozlišované pomocí tří číselného kódu.

Komunikace mezi klientem a serverem se tedy skládá z FTP příkazů posílaných klientem a FTP odpovědí zasílaných serverem. Na jeden FTP příkaz může server odpovědět i více FTP odpověďmi (např.: příkaz: „přenes soubor“, odpověď: „přenos zahájen“, [vlastní přenos], odpověď „přenos dokončen“), ale server až na výjimky neposílá odpovědi bez předchozího příkazu. Standardizované číselné kódy odpovědí byly zavedeny proto, aby software na straně klienta mohl rychle rozhodnout, zda a jak se poslaný příkaz provedl či proč se neprovedl. První číslice kódu odpovědi určuje to, jestli se požadovaná akce zdařila nebo ne, viz **Tab. 11**.

Tab. 11: Význam první číslice tří-číselného kódu v odpovědi FTP serveru

První číslice	Zpráva
1	Požadovaná akce byla úspěšně započata, čekej další odpověď
2	Příkaz úspěšně proveden, může být zadán další
3	Příkaz byl přijat, server očekává další (používá se o u sekvence souvisejících příkazů, např. přihlášení na server)
4	Akce se nezdařila, ale existuje možnost pokusit se provést příkaz znovu (např. dočasná nedostupnost)
5	Příkaz nebyl proveden, např. není podporován, uživatel není oprávněn apod.

Druhá a třetí číslice přibližuje více druh chyby nebo odpovědi. Např. kód 231 značí, že uživatel byl korektně odhlášen a služba ftp tak byla ukončena.

11.5.3 Pracovní režimy vzniku datového spojení

FTP podporuje dva režimy otevírání datového spojení.

V **aktivním režimu** otevře klient náhodný port (>1023) a pošle serveru přes řídicí spojení jeho číslo. Klient na portu naslouchá a očekává *navázání datového spojení* serverem. Tento režim je výchozí, nicméně může být problematický, pokud uživatel či jeho síť využívá nějaké filtrování nevyžádaného provozu, či je za NATem apod.

V **pasivním režimu** probíhá založení datového spojení přesně naopak, tj. server otevírá náhodný port (>1023) a prostřednictvím řídicího kanálu vybídne klienta, aby navázal spojení. Tento režim je výhodný, pokud je FTP klient „schován“ ve vnitřní síti s privátní adresou a server tak není schopen se přímo na jeho port připojit. Pokud má být použit pasivní režim, musí se na tom klient a server nejdříve dohodnout na řídicím spojení.

11.6 WWW a protokol HTTP

11.6.1 Stručná historie vzniku a současnost WWW

V roce 1989 byla napsána první kapitola webu ve švýcarském výzkumném středisku CERN v Ženevě, které bylo v té době jako jedno z mála připojeno k rodícímu se Internetu. V roce 1989 Tim Berners-Lee definoval hypertextový systém pro CERN. O rok později napsal Tim Berners-Lee program pro tvorbu primitivních hypertextových stránek a systém běžící na *jediném* počítači nazval "**World-Wide Web**". Dále už následují roky nepřetržitého růstu. V roce 1992 existuje na světě okolo padesáti webových serverů. Vznikají první grafické prohlížeče, které postupně nahrazují textové. Vzniká *Mosaic* a architektura *Mozilla,* na které jsou postaveny prakticky všechny dnes nejpoužívanější prohlížeče. Je ustaveno W3C (WWW Consortium), které má dohlížet a schvalovat standardy na Internetu. V roce 1995 je na světě skoro 100 000 webových serverů, v současné době a vzhledem k masivní virtualizaci je těžké servery posčítat, avšak podle hrubých odhadů se toto číslo pohybuje v řádu desítek miliónů.

Uživatelé vyžadují stránky, které jim poskytnou interaktivní přístup k informacím. Webové stránky se dnes nepoužívají jen k prezentaci dokumentů na Internetu. Web, který byl původně vyvinut jako pomůcka pro sdílení výsledků vědeckého výzkumu po celém světě, se dnes stává spíše novou platformou, na které mohou běžet nejrůznější aplikace. Na Internetu existují obrovské virtuální obchodní domy, rezervační systémy, bankovní systémy, podnikové informační systémy, prohledávací katalogy a mnohé jiné aplikace. Prohlížeče se stávají novou platformou pro aplikace, které nejsou závislé na konkrétním operačním systému.

11.6.2 Technologie kolem WWW

Zpočátku si **W**orld-**W**ide **W**eb vystačil pouze se třemi technologiemi:

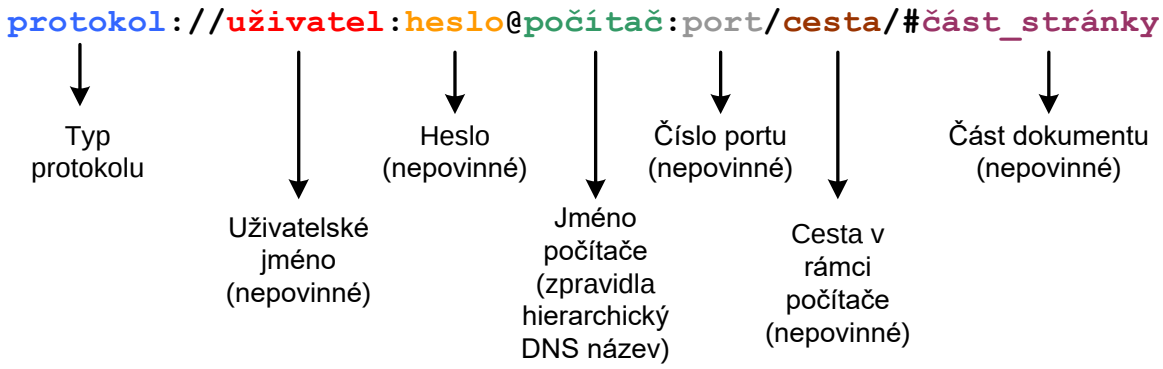
- **jazyk HTML** (*HyperText Markup Language*) – značkovací jazyk, který slouží k zápisu základních webových stránek. Všechny verze tohoto jazyka jsou zpětně kompatibilní.
- **protokol HTTP** (*HyperText Transfer Protocol*) – zajišťuje přenos HTML stránek a případně dalších dat mezi WWW serverem a prohlížečem na straně uživatele.
- **URL** (*Uniform Resource Locator*) – každý objekt na webu má jedinečnou URL adresu, která slouží k jeho jednoznačné identifikaci a umožňuje tak vytvoření odkazů mezi objekty.

Z dnešního pohledu spojení těchto tří technik již nabízí málo, vlastně lze „jen“ prohlížet statické elektronické dokumenty, které jsou propojeny elektronickými odkazy. V současné době se již požaduje víc, např. určitá míra interaktivity a dynamiky stránek. Stránky musí umět reagovat na požadavky uživatelů a zpřístupňovat informace, které se v čase mění. Toho principiálně docílíme tak, že každý požadavek zobrazení stránky vyvolá spuštění skriptu

(kódu), který zjistí všechny potřebné informace a složí z nich výslednou podobu stránky v HTML jazyku, kterou zašle prohlížeči.

11.6.3 URL (Uniform Resource Locator)

Zkratka URL znamená jednotný lokátor zdrojů a představuje jednoznačné síťové umístění nějakého zdroje nebo přímo dokumentu. Řetězec má přesně definovanou strukturu, některé části jsou volitelné, znázornění viz **Obr. 11-10**. Některé položky spolu souvisí, např. heslo není nikdy přítomno bez uživatelského jména. Adresa počítače vzniká hierarchicky podle umístění počítače v doméně, viz kap. o DNS (11.3). Položka port je nepovinná a prohlížeč její hodnotu doplní podle typu protokolu, implicitně např. u HTTP protokolu je to port 80, u https 443.



Obr. 11-10: Formát zápisu jednotného lokátoru zdroje (URL)

Příklady možných URL:

http://www.vutbr.cz/

https://is.vutbr.cz/

http://en.wikipedia.org/wiki/Domain_name_system#DNS_resolvers

11.6.4 Obecný popis protokolu HTTP

HTTP (*Hyper Text Transfer Protocol*) je Internetový (ASCII orientovaný) **aplikační protokol** určený původně pouze pro výměnu hypertextových dokumentů ve formátu HTML. Slouží ke komunikaci **mezi klientem** (zpravidla www prohlížeč) **a** WWW **serverem**. Definuje tvar dat, která jsou přenášena a taktéž formát dotazů a odpovědí komunikujících stran. Na straně serveru se používá standardně port **80**, přičemž nejčastěji se můžeme setkat s použitím protokolu TCP, avšak použití **UDP/80** taktéž není neobvyklé. V současné době jsou relevantní verze HTTP/1.1, HTTP/2 a připravované HTTP/3, které se dosti výrazně liší svými vlastnosti a chováním. Spolu s elektronickou poštou je patrně HTTP nejvíce používaným protokolem Internetu.

Protokol HTTP je používán i pro přenos dalších informací. Pomocí **rozšíření MIME** (*Multipurpose Internet Mail Extension*) umí HTTP (i email) **přenášet jakýkoli soubor**, používá se pro tzv. webové služby (spouštění vzdálených aplikací) a pomocí aplikačních bran zpřístupňuje i další protokoly, jako je např. FTP nebo SMTP.

HTTP používá stejně jako některé další aplikace tzv. jednotný lokátor prostředků (URL), viz kap. 11.6.3.

Standard HTTP obsahuje definici **číselných výsledkových kódů**, které tvoří odezvu na podnět klienta od serveru, podobně jako v případě FTP protokolu.

11.6.5 Činnost protokolu HTTP

Protokol funguje způsobem **dotaz – odpověď**. Uživatel pošle serveru dotaz ve formě čistého textu, obsahujícího označení požadovaného dokumentu, informace o schopnostech prohlížeče apod. Server poté odpoví pomocí několika řádků textu, které popisují výsledek dotazu (zda se dokument podařilo najít, jakého typu dokument je atd.), za kterými následují samotná data požadovaného dokumentu. **Původní** protokol **HTTP** (verze 1.0) **je** zcela **bezestavový**, což znamená, že server klienty žádným způsobem nerozlišuje a jejich jednotlivé dotazy bere jako zcela samostatné jednotky.

To, že klasické HTTP (do verze 1.0) neumí uchovávat stav komunikace, znamená, že přenosu každého objektu předchází navázání TCP spojení, po přenosu následuje ukončení spojení TCP (pokud je použito TCP). To je velmi nepraktické, jelikož navazování a ukončování spojení pak výrazně zpomaluje komunikaci. Ve verzi 1.1 je spojení perzistentní, což znamená, že během jednoho spojení je možné přenést postupně, více objektů, spojení je pak zpravidla po určité době nečinnosti ukončeno (ze strany serveru).

Existují i novější verze protokolu – HTTP/2 a HTTP/3, které mají upravené vlastnosti. Protokol HTTP ve verzi 1.1 a novější se používá i k celé řadě dalších účelů komunikace různých aplikací.

11.6.6 Vybrané metody protokolu HTTP

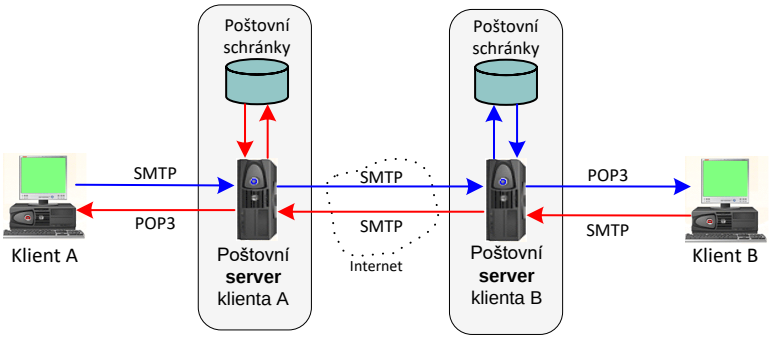
Metoda představuje způsob, pomocí kterého se specifikují požadavky klienta na webový server. Jsou to zejména:

- **OPTIONS** – tato metoda představuje dotaz na možnosti komunikace spojené s uvedenou URL.
- **GET –** *požadavek na poslání dokumentu* serveru specifikovaného pomocí URL. Tato metoda je prohlížeči využívána nejčastěji.
- **HEAD** – *požadavek na čtení záhlaví* www stránky specifikované pomocí URL. Metodu je možné použít k získání doplňkových informací o dokumentu. Tato metoda je hojně využívána vyhledávači.
- **PUT** – *požadavek na uložení posílaných dat* pod specifikovanou URL na server. Takto uložená data budou dostupná např. následnými dotazy GET.

11.7 Elektronická pošta a protokol SMTP

11.7.1 Schéma klasického způsobu přenosu emailů

Elektronická pošta je bezpochyby nejrozšířenější ze všech služeb, které Internet nabízí. Stejně jako většina základních služeb Internetu je založena na modelu klient – server. Schematické znázornění klasického přenosového systému je patrné z **Obr. 11-11**, popis následuje.



Obr. 11-11: Elektronická pošta v prostředí TCP/IP sítí

Klient A z **Obr. 11-11** chce odeslat email Klientu B, který je přítomen v jiné organizaci, přičemž každá organizace disponuje vlastním poštovním serverem. Klient A prostřednictvím svého **MUA** (*mail user agent*), tedy poštovního klienta odešle pomocí protokolu **SMTP** (*Simple Mail Transfer Protocol*, viz dále) na svůj poštovní server emailovou zprávu pro Klienta B. Ten ji po přijetí vyhodnotí a na základě emailové adresy adresáta (resp. pouze části za znakem @ [46]) určí, na který poštovní server má zprávu předat (server klienta B). Server B zprávu uloží do příslušné schránky, samozřejmě pouze za předpokladu, že se na něm tato schránka nachází. Klient B pak po připojení k poštovnímu serveru obdrží tuto zprávu např. pomocí protokolu **POP3** (*Post Office Protocol* verze 3, kterým se více zabývat nebudeme). Pro přenos mezi servery je využíván protokol SMTP. V obrázku je modře vyznačen běh zprávy od klienta A k B. Pokud klient B na tuto zprávu odpoví, průběh bude odpovídat červené čáře.

Část poštovního serveru, která se stará o přenos zpráv mezi servery, je někdy označována jako **MTA** (*Mail Transport Agent*), což poměrně dobře vystihuje její účel.

Část označovaná jako **MDA** (*Mail Delivery Agent*) tvoří program pro lokální doručování do uživatelských schránek v rámci jednoho serveru.

Poštovní server může být součástí vnitřní sítě určité organizace. V takovém případě na něm mají schránky pouze členové (zaměstnanci) této instituce. Velmi časté je však i použití veřejných poštovních serverů, které jsou dostupné všem zájemcům prostřednictvím Internetu, konkrétní příklady není třeba zmiňovat. Tyto servery jsou ve velké většině dostupné zdarma, z čehož samozřejmě vyplývají určitá omezení.

11.7.2 Formát zprávy elektronické pošty

Každá emailová zpráva se dělí na dvě základní části:

- **Záhlaví** (*header*) – přesně strukturované, obsahující mnohé položky, jejichž obsah je pevně dán. Obsahuje informace, podle kterých jsou jednotlivé zprávy odesílané, přenášené a doručované. Vybrané položky jsou popsány v **Tab. 12**.
- **Tělo zprávy** (*body*) – obsahuje vlastní text zprávy a z pohledu systému elektronické pošty není nikterak strukturované. Tuto část interpretuje až klientův poštovní agent. Od záhlaví je tělo odděleno jednoduše, pouze jedním nebo více prázdnými řádky.

[46] Je víc než dobře známo, že každá emailová adresa se skládá ze dvou částí, oddělených pomocí znaku „@“, v angličtině slovně jako „at“. Druhá část představuje adresu poštovního serveru a první je označení konkrétní schránky, která se na něm nachází. Emailový systém musí být schopen spolupráce se systémem DNS, který zajistí dohledání IP adresy serveru, kterému se má zpráva předat.

Tab. 12: Vybrané položky záhlaví emailové zprávy

Položka	Význam
Received	toto záhlaví připisuje na počátek e-mailu každý emailový server, kterým zpráva projde (může jich být po cestě i více). Při čtení od spodu nahoru lze zjistit celou cestu, kterou zpráva prošla.
From	Adresa odesílatele zprávy
Sender	kdo vyřizuje, např. sekretářka (nebo zde bývá i informace o konferenci, přes kterou zpráva přišla, pokud to tak je)
Reply-To	Odpověď zasílejte na (emailová adresa)
In-Reply-To	Odpověď na konkrétní zprávu
To	Adresát zprávy
Cc	Kopie zprávy (*carbon copy*)
Bcc	Utajená kopie, před odesláním (adresátovi v *To* či *Cc*) se toto záhlaví smaže (*background carbon copy*)
Subject	krátká charakteristika zprávy (předmět)

11.7.3 SMTP (Simple Mail Transfer Protocol)

Protokol SMTP je primární **standard** pro přenos emailů Internetem. Protokol je určen pro **spolehlivý přenos zpráv** elektronické pošty mezi dvěma stanicemi (resp. servery). Využívá port **TCP/25** na straně serveru. Jak je patrné z **Obr. 11-11**, SMTP se používá pro přenos zprávy od klienta na server a zejména pak mezi poštovními servery. Pro přístup uživatele do schránky se používají jiné protokoly (POP3, IMAP4).

Komunikace je založena na principu **klient – server**. Z toho vyplývá, že **poštovní server musí obsahovat obě tyto části**, aby mohl jednak komunikovat s uživatelem, kde je v roli serveru a zároveň kontaktovat další poštovní server, sám v roli klienta.

Vlastní protokol SMTP je poměrně jednoduchý, jednotlivé příkazy jsou textové v kódu ASCII, podobně jako u FTP a dalších protokolů. Klient vkládá do navázaného spojení čtyřznakové příkazy a server odpovídá stavovými kódy s textovým popisem, podobně jako u protokolu HTTP nebo FTP.

V současné době se převážně používá rozšířená verze, tedy **ESMTP** (*Extended SMTP*). Tato umožňuje mimo jiné např. přenos potvrzování o doručení emailové zprávy.

Z pohledu uživatele a nastavení jeho emailového klienta je podstatná role **SMTP serveru odchozí pošty**. Každý program fungující jako **poštovní klient vždy vyžaduje nastavit server odchozí pošty, pokud chce uživatel zprávy nejen přijímat, ale i odesílat**. Tento server vlastně doručuje zprávy v zastoupení za uživatele. Uživatelé jsou často nuceni mít nastaven v rámci sítě konkrétní server SMTP, přístup na jiné je záměrně blokován. To umožňuje poskytovateli připojení (nebo organizaci) snadněji odfiltrovat *spam* nebo jiné nežádoucí zprávy hned v zárodku, nastavovat různé politiky odesílání zpráv apod.

11.8 Vybrané protokoly realizující VoIP komunikaci

Pro uskutečnění **VoIP** (Voice over IP) hovoru, popř. přenosu jiných multimédií v reálném čase, existují standardizované protokoly, které pracují primárně na aplikační vrstvě TCP/IP. Využity jsou především protokoly **SIP** (*Session Initiation Protocol*), **RTP** (*Real-time Transport Protocol*) a **RTCP** (*RTP Control Protocol*). První z protokolů zajišťuje spojení účastníků hovoru. V případě, že je hovor spojen, tak přebírá komunikaci protokol RTP, který má na starosti přenos multimediálních dat, jako je zvuk (často hovor) nebo v jiných případech i video. RTCP protokol pak poskytuje určitou formu řízení přenosu RTP protokolem a možnost vyhodnocení dosahovaných parametrů přenosu. V dalším textu budeme primárně předpokládat použití těchto protokolů pro VoIP, tedy přenos hlasu, přestože použití těchto protokolů je obecnější.

SIP je kontrolní protokol, pracující na aplikační vrstvě. Je určen k vytváření a ukončování relací, jako jsou VoIP hovory nebo multimediální konference, mezi jedním nebo několika uživateli. Účastníci si pomocí SIP protokolu dohodnou například to, jaký kodek[47] budou při hovoru používat. Pro sestavení hovoru je nutné propojit obě komunikující strany. Jako příklad zpráv používaných u tohoto protokolu uveďme zprávu označovanou jako INVITE, která slouží ke kontaktování druhé strany a zprávu označovanou jako 200 OK, která znamená úspěšné spojení, resp. přijetí hovoru.

RTP je protokol, který umožňuje přenášet (multimediální) data v reálném čase. Sám o sobě nezaručuje doručení přenášených paketů, stejně jako UDP protokol, na kterém je RTP postaven. Každý RTP paket ale obsahuje pořadové číslo a také časové razítko, které pomáhají při rekonstrukci toku paketů na přijímající straně do správného pořadí. Záhlaví protokolu sestává z 12 bajtů a přenášená data jsou v jednom paketu zastoupena 160 bajty. Těmto 160 bajtům se říká chunky (*chunk*), což jsou zakódované kusy hlasových dat, pokrývající jen několik milisekund hlasového signálu.

RTCP je řídící protokol, který umožňuje sledovat multimediální přenos RTP protokolem, resp. vyhodnocovat jeho parametry a informovat o zjištěných skutečnostech druhou stranu hovoru. To je možné díky výměně dalších paketů mezi komunikující stranami, a to v omezené míře, aby nedošlo k zahlcení přenosové trasy. Díky tomu může VoIP aplikace v případě potřeby upravovat parametry přenosu, tak aby byla dosažena požadovaná kvalita služby (QoS = *Quality of Service*). Tato úprava může souviset např. se změnou používaného kodeku. RTCP je tedy vlastně servisním kanálem, který je využíván během multimediálního přenosu.

Všechny výše uvedené protokoly využívají typicky na transportní vrstvě protokol **UDP**. Použití tohoto protokolu je výhodné především kvůli jeho jednoduchosti a možnosti si nad jeho základními funkcemi vybudovat potřebné služby. TCP protokol je nevhodný především kvůli snaze vždy všechna data doručit i opakovaně, v případě ztrát či chyb při komunikaci. To při VoIP komunikaci nemusí být vždy žádoucí či potřebné z důvodu interaktivity komunikace a snahy o dosažení co nejmenšího zpoždění při přenosu.

VoIP služby zpravidla vyžadují nějaké infrastrukturní prvky (servery, brány), které komunikaci zprostředkovávají, jsou schopny konverze kodeků či formátu podle typu volané strany, popř. hovor účtují či provádí další nezbytné úkony (ukládání údajů do databáze).

[47] Kodek, je způsob, jakým jsou kódována a přenášena multimediální data, např. hovorový signál. Standardně používaných kodeků existuje velké množství a liší se svým určením a vlastnostmi. Některé kodeky jsou zaměřeny na co nejmenší potřebnou šířku pásma za cenu nízké kvality, některé naopak na co největší věrnost a tedy i kvalitu následné reprodukce. Podrobněji se této problematice v tohoto textu nebudeme věnovat.

Seznam použité literatury

[1] DOSTÁLEK, Libor a Alena KABELOVÁ. *Velký průvodce protokoly TCP/IP a systémem DNS*. 3. aktualiz. a rozš. vyd. Praha: Computer Press, 2002, xiv, 542 s. ISBN 80-7226-675-6.

[2] DOSTÁLEK, Libor et al. *Velký průvodce protokoly TCP/IP: bezpečnost*. 2. aktualiz. vyd. Praha: Computer Press, 2003, xvi, 571 s. ISBN 80-7226-849-x.

[3] PUŽMANOVÁ, Rita. *TCP/IP v kostce*. 1. vyd. České Budějovice: Kopp, 2004, 607 s. ISBN 80-7232-236-2.

[4] PUŽMANOVÁ, Rita. *Moderní komunikační sítě od A do Z*. 2. aktualiz. vyd. Brno: Computer Press, 2006, 430 s. ISBN 80-251-1278-0.

[5] PUŽMANOVÁ, Rita. *Širokopásmový Internet: přístupové a domácí sítě*. 1. vyd. Brno: Computer Press, 2004, 377 s. ISBN 80-251-0139-8.

[6] FRED, Halsall. *Computer networking and the internet*. Vyd. 1. Edinburg: Addison-Wesley, 2005, 803 s. ISBN 0-321-26358-8.

[7] STALLINGS, William. *Data and computer communications*. 8th ed. Upper Saddle River: Pearson Education, 2007, xviii, 878 s. ISBN 0-13-243310-9.

[8] OSTERLOH, Heather. *TCP/IP: kompletní průvodce : použitelný pro veškeré operační systémy*. Praha: SoftPress, 2003, 512 s. ISBN 80-86497-34-8.

[9] ODOM, Wendell a Tom KNOTT. *Networking basics: CCNA 1 companion guide*. Indianapolis: Cisco Press, 2006, xxx, 587 s. ISBN 1-58713-164-1.

[10] ODOM, Wendell a Rick MCDONALD. *Routers and routing basics: CCNA 2 companion guide*. Indianapolis: Cisco Press, 2007, xxviii, 473 s. ISBN 1-58713-166-8.

[11] LEWIS, Wayne. *Switching basics and intermediate routing: CCNA 3 companion guide*. Indianapolis: Cisco Press, 2006, xxiv, 307 s. ISBN 1-58713-170-6.

[12] REID, Allan. *WAN technologies: CCNA 4 companion guide*. Indianapolis: Cisco Press, 2007, xx, 254 s. ISBN 1-58713-172-2.

[13] DROMS, Ralph a Ted LEMON. *DHCP příručka administrátora*. Vyd. 1. Brno: Computer Press, 2004, 490 s. ISBN 80-251-0130-4.

[14] LOSHIN, Pete. *IPv6 theory, protocol, and practice*. 2nd ed. San Francisco: Morgan Kaufmann, 2004, xxiv, 536 s. ISBN 1-55860-810-9.

[15] STEVENS, Richard W, Bill FENNER a Andrew M RUDOFF. *Unix network programming*. 3rd ed. Boston: Addison-Wesley, 2004, xxiii, 991 s. ISBN 0-13-141155-1.

[16] STEVENS, W. *UNIX network programming*. 2nd ed. Upper Saddle River: Prentice Hall, 1999, xvii, 558 s. ISBN 0-13-081081-9.

[17] REILLY, David a Michael REILLY. *Java network programming and distributed computing*. Boston: Addison-Wesley, 2002, 464 s. ISBN 0-201-71037-4.

[18] MOLLIN, Richard A. *An introduction to cryptography*. 2nd ed. Boca Raton: Chapman & Hall/CRC, c2007, x, 413 s. ISBN 1-58488-618-8.

[19] TANENBAUM, Andrew S a Maarten VAN STEEN. *Distributed systems: principles and paradigms*. 2nd ed. Upper Saddle River: Pearson Education, 2007, xviii, 686 s. ISBN 0-13-239227-5.

[20] TANENBAUM, Andrew S. *Computer networks*. 4th ed. New Jersey: Prentice-Hall, c2003, xx, 891 s. ISBN 0-13-066102-3.

[21] TANENBAUM, S. *Modern operating systems*. Vyd. 1. New Jersey: Prentice-Hall, 2001, 950 s. ISBN 0-13-031358-0.

[22] SATRAPA, Pavel. *IPv6: internet protocol verze 6*. Vyd. 3. Praha: CZ.NIC, 2012, 409 s. ISBN 978-80-904248-4-5.

[23] CONLAN, Patrick J. *Cisco network professional´s advanced intenetworking guide*. Hoboken: Wiley Publishing, 2009, xxvii, 854 s. ISBN 978-0-470-38360-5.

[24] COLE, Eric, Ronald D KRUTZ a James W CONLEY. *Network security bible*. 2nd ed. Indianapolis: Wiley Publishing, 2009, xliv, 891 s. ISBN 978-0-470-50249-5.

[25] DAVIES, Joseph. *Understanding IPv6*. 2nd ed. Redmond: Microsoft Press, 2008, xlii, 556 s. ISBN 978-0-7356-2446-7.

[26] FOROUZAN, Behrouz A. *TCP/IP protocol suite*. 4th ed. Boston: McGraw-Hill Higher Education, 2010, xxxv, 979 s. ISBN 978-0-07-337604-2.

[27] PETERSON, Larry L a Bruce S DAVIE. *Computer networks: a systems approach*. 5th ed. Burlington: Morgan Kaufmann, 2011, xxxi, 884 s. ISBN 978-0-12-385059-1.

[28] KUROSE, James F a Keith W ROSS. *Computer networking: A Top-Down Approach*. 7th ed. (global edition). Harlow: Pearson Education Limited, 2017, 852 s. ISBN 978-1-292-15359-9.