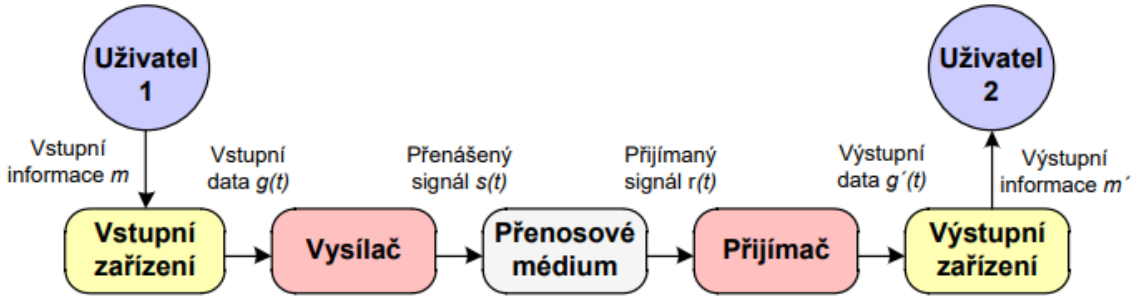
1 Technika sítí a protokolů - komunikační modely, způsob přenosu informace, základní struktura sítí, typy sítí, architektura komunikace systémů

1.1 Komunikační modely

- Základním účelem komunikace je vzájemná výměna informací mezi dvěma uživateli (zdrojem a spotřebičem). Lze ji rozdělit na **dva typy komunikací**:
  - **komunikace uvnitř sít**í včetně dohledu nad touto komunikací, tj. určitá servisní část komunikace
  - **komunikace mezi koncovými uživateli**, tj. komunikace nad sítěmi
- Základní pojmy
  - **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 je mezi dvěma uživateli vyměňována informace nazvaná „*m*“. Informace *m* je pomocí vstupního zařízení reprezentována jako data *g(t)*, ve formě časově proměnlivého signálu. V tomto okamžiku 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´*“. Komunikací uvnitř sítě se rozumí část od vstupního k výstupnímu přenosovému zařízení, zatímco pro uživatele by se tato část měla jevit jako zcela transparentní.



Zjednodušené blokové schéma datové komunikace (pro jednoduchost přenos pouze jednosměrně)

- Komunikační sítě slouží vždy k tomu, aby pomocí nich mohli spolu komunikovat koncoví uživatelé, v případě použití počítačů pak vzájemně komunikují odpovídající si (partnerské) procesy, které běží každý na jednom z komunikujících počítačů (uvažujeme-li pouze komunikaci bod-bod). Základním předpokladem pro komunikaci uživatelů je definice rozhraní mezi uživatelem a sítí. **Rozhraní** musí definovat strukturu a formát předávaných uživatelských a řídicích dat.

Zjednodušené schéma komunikace mezi procesy pracujícími na samostatných počítačích propojených obecnou sítí

- Tři základní úkoly pro přenos informací spočívají v:
    - vlastní přenos informace – 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 – 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 – mezi přijímačem a vysílačem musí existovat určité formy časového sjednocení, tak, aby bylo možno rozeznat jednotlivé signálové elementy.
    - Formátování zpráv – unifikace způsobu sestavení obsahu zprávy, tak aby si odpovídající části v komunikačním modelu „rozuměly“.
    - Adresování a směrování – jednoznačný způsob určení cíle a nalezení cesty k němu.
- Komunikační řetězec zpravidla umožňuje:
    - Vícenásobné využití přenosových systémů – komunikační řetězec je sdílen více uživateli, případně více procesy.
    - Řízení systému – 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 – zajištění, aby nedocházelo k zahlcení systému nadměrným množstvím přenášené informace.
    - Ochranu zpráv – posílaná data může přijímat pouze zvolený příjemce.

1.2 Způsoby přenosu informace

- 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:

1.2.1 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. Důležitá vlastnost – nutnost sestavit spojení před vlastním přenosem informace (rezervace prostředků a kapacit pro následný přenos) Tato operace posunuje dobu zahájení přenosu (může trvat cca 10 sekund). Z hlediska nákladů „drahé“ spojení – platí za celou dobu sestaveného spojení, i když nedochází k přenosu informace po celou dobu. Převážně pro přenos hovorových signálů (klasická pevná i mobilní telefonní síť), kde neexistoval donedávna jiný způsob.

1.3 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. 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. Výhoda – vždy zatěžována pouze ta část sítě, kterou se právě daná zpráva přenáší.

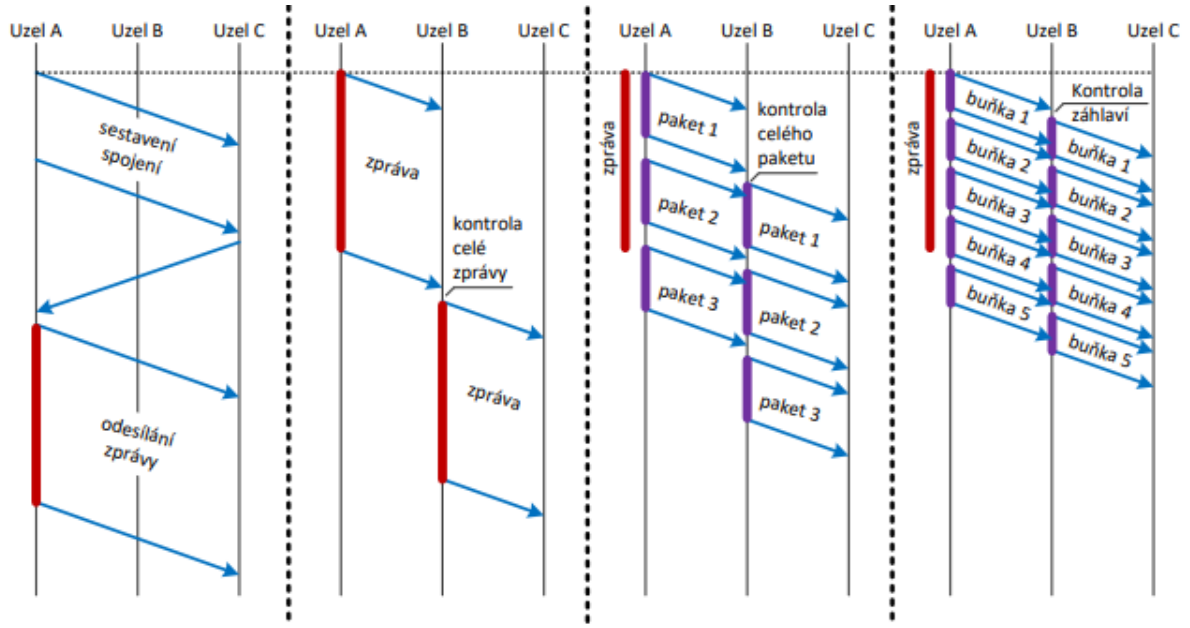
1.3.1 Komutace paketů (packet switching)

- Obdobné vlastnosti jako komutace zpráv. Rozdílem je, že pokud je zpráva dlouhá, tak je rozdělena na bloky dat – pakety o definované maximální délce. Sítí pak přenášeny jednotlivé pakety, obdobně jako v předcházejícím případě zprávy. 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.

1.3.2 Komutace buněk (cell switching)

- Zpravidla rozdělení na menší jednotky s přesně definovanou délkou. Při přenosu se provádí kontrola pouze u záhlaví buňky (či rámce). Proto jen velmi malé zdržení přenášené jednotky v uzlu. Uživatel pak musí provádět veškeré kontroly přenesených dat samostatně. Možné využít k přenosu řečového signálu i klasických dat (např. u ATM technologií)Výhoda oproti komutaci okruhů – úspora prostředků sítě, jelikož pro daný přenos je zpravidla blokována jen nezbytně nutná kapacita. Výhoda oproti komutaci zpráv a paketů – rychlejší odezva. Nevýhoda – fixní velikost přenášené jednotky.



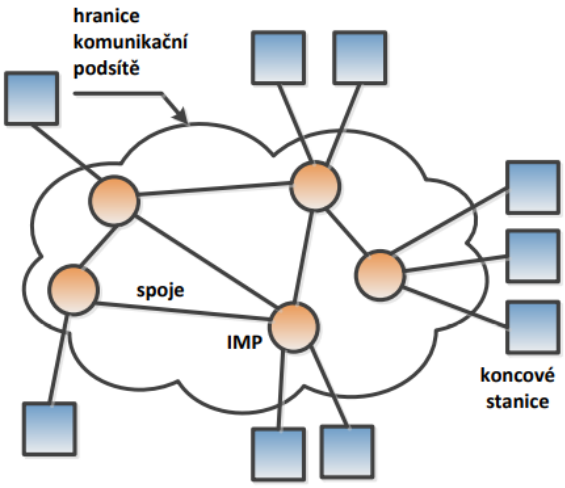
a) komutace okruhů b) komutace zpráv c) komutace paketů d) komutace buněk

Obr. 3-3: Časové posloupnosti jednotlivých metod přenosu informace

1.4 Základní struktura sítí

1.4.1 Základní prvky sítě

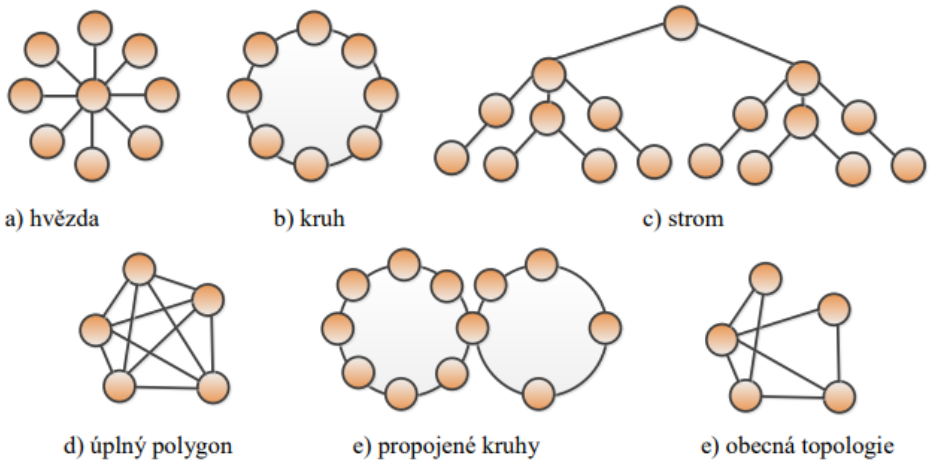
- Každá komunikační síť má vždy dvě základní komponenty, které **slouží k propojení koncových stanic**.
- **Spoje** představují technické prostředky umožňující přenos zpráv mezi dvěma místy bez ohledu na druh použitých prostředků a druh přenosů. Propojují přepojovací prvky mezi sebou, případně s koncovými uzly. V některých případech mluvíme o **okruzích, (telekomunikačních) kanálech nebo linkách.**
- **Přepojovací prvky** jsou specializované spínané systémy, případně počítače sloužící k propojení dvou nebo více spojů. Základní úlohou tohoto prvku je, u přijatých dat vybrat příslušný výstupní spoj, po kterém jsou data poslána dále. Přepojovacím prvkem pro datové přenosy je např. **IMP** (Interface Message Processor), což je předchůdce dnešních směrovačů, dále se můžeme potkat s pojmenováním jako např. datová ústředna (data switching exchange), mezilehlý systém (intermediate system) a uzel přepojování paketů (packet switch node). Názvy zde uvedené jsou si rovnocenné, neboť obecná terminologie není sjednocena. Veškerý přenos mezi hostitelskými počítači prochází přes IMP.



Obr. 3-4: Základní struktura sítě

1.4.2 Architektura a topologie sítí

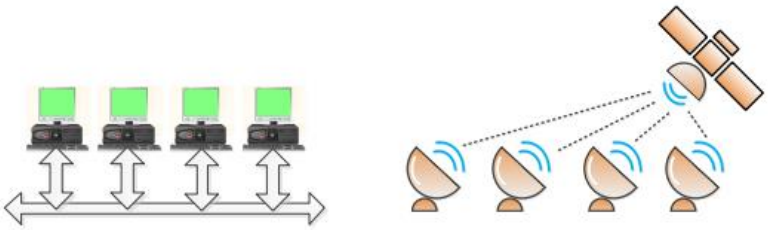
- Z hlediska topologie sítí existují dva základní způsoby spojení uzlů sítě.
- **Dvoubodové spoje** (point-to-point),
    - Tvořeny řadou spojů, z nichž každý propojuje koncovou stanici s přepojovacím uzlem nebo tyto uzly navzájem.
    - Informace vyměňována nepřímo. Možné struktury
        - hvězda
        - kruh
        - strom
        - úplný polygon (úplné propojení, tj. každý přímo s každým dalším, full mesh)
        - propojené kruhy
        - obecná topologie (neúplný polygon, partial mesh).



Obr. 3-5: 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. 3-6: Topologie sítí založených na všesměrovém vysílání

1.5 Typy sítí (dle velikosti)

- Nejběžnější 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.

1.5.1 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, PDA, tablety, ale i počítače nebo např. scannery či tiskárny. Typicky se jedná o bezdrátové (popř. i drátové) technologie s dosahem v řádu jednotek metrů, jako jsou např. USB, Firewire, Bluetooth nebo IrDA (Infrared Data Association). Příkladem je např. přenos dat mezi dvěma telefony nebo telefonem a počítačem.

1.5.2 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 a s rychlostmi 100 Mbit/s, 1 Gbit/s, ojediněle i 10 Gbit/s. Dřívější LAN sítě využívají také kruhové nebo sběrnicové provedení s rychlostmi 10 až 100 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 nebo Gigabit Ethernet (dříve Ethernet nebo Token Ring).

1.5.3 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

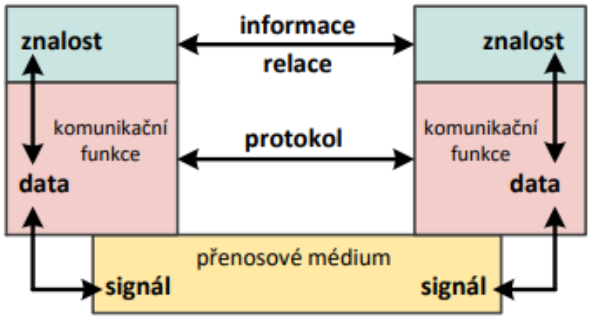
jsou využívány více subjekty. Zpoždění v těchto sítích je podobně jako u LAN sítí velmi nízké, přibližně na úrovni 100 µs až 10 ms.

1.5.4 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), 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.

1.6 Architektura komunikace systémů

- Přenos informace mezi komunikujícími stranami se děje dle dohodnutých pravidel – **protokolu** (a samozřejmě přes přenosové médium)
- V každém komunikačním systému pak existují dva způsoby komunikace:



Obr. 3-7: Základní model komunikace – principiální schéma

1.6.1 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. Obecně tento postup umožňuje připravovat znalosti tak, aby mohly být odeslány přes přenosové médium.

1.6.2 Horizontální komunikace

- Probíhá na dvou odpovídajících si úrovních, mezi nimiž musí existovat forma „společné řeči“. Na každé dvojici vzájemně si odpovídajících úrovní komunikačního systému musí být odpovídající protokol. 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.

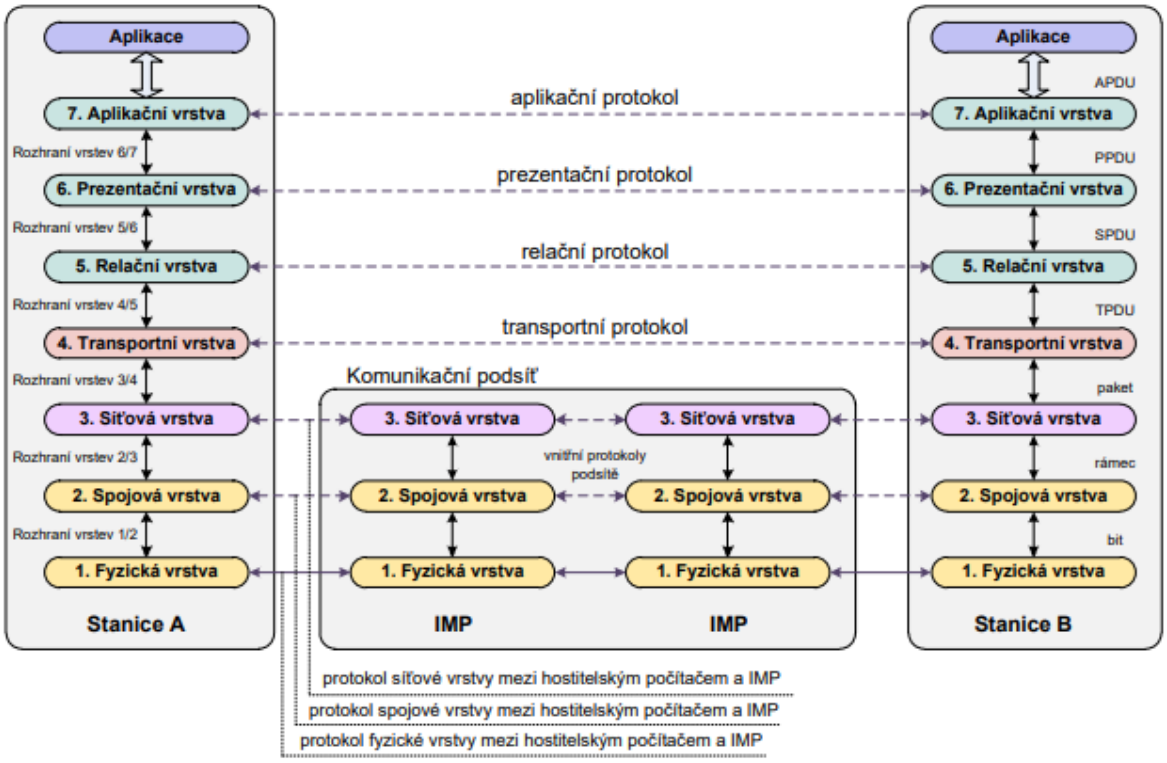
2 Základní popis referenčního modelu ISO/OSI a srovnání s TCP/IP.

- Síťová komunikace byla již od začátků svého budování založena na vrstvách, tak jak je popsáno v této kapitole. 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.
- Úlohou referenčního modelu je především poskytovat společnou základnu pro další aktivity. V terminologii OSI se zařízení, která jsou schopna vykonávat zpracování a přenos informace (počítače, periferní zařízení, …) označují jako **reálné systémy**. Prvek, který v takovémto reálném systému vykonává zpracování údajů pro určitou aplikaci, se nazývá **aplikační proces**
- 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 vrstva** (nesprávně linková)
    3. **síťová vrstva**
    4. **transportní vrstva**
    5. **relační vrstva**
    6. **prezentační vrstva**
    7. **aplikační vrstva**

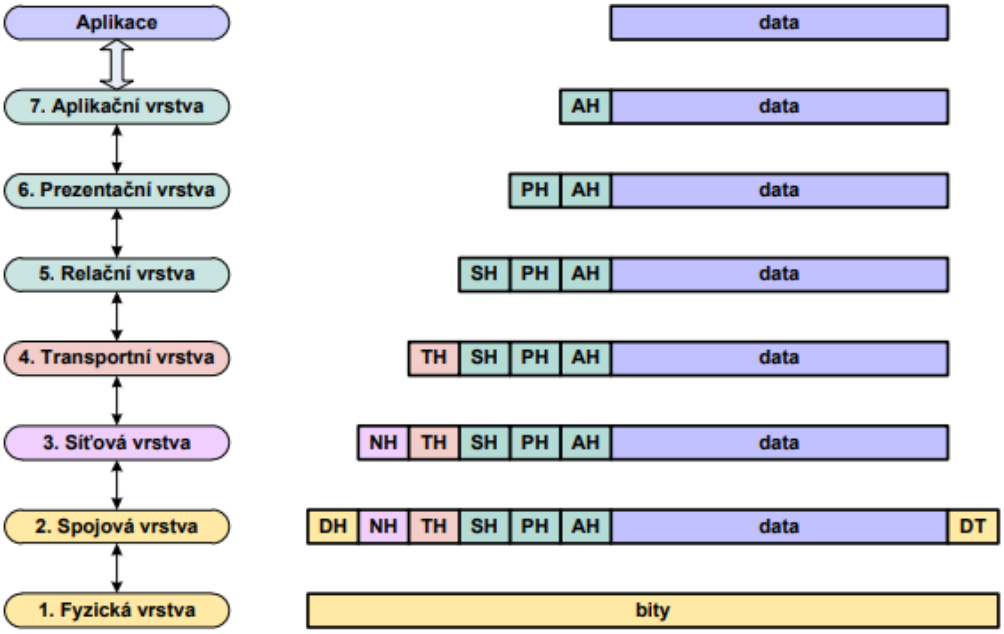
- Implementace vrstev může být softwarová nebo hardwarová. Nejčastější je situace, kdy **nejnižší dvě** vrstvy jsou zejména hardwarové a všechny **vyšší vrstvy jsou již zejména softwarové**.
- Vrstvy 1-4 lze shrnout do označení „**poskytovatelé transportní služby**“, vrstvy 5-7 jako „**uživatelé transportní služby**“.
- Častější je však rozdělení na skupinu vrstev 1-3 a 4-7. 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.
- Zkratky:
    - **IMP** (Interface Message Processor) přepínací bod mezilehlé sítě v době ARPANETu2 ,
    - **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.

- 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.



Obr. 3-12: Architektura sítě založené na modelu ISO/OSI

- 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.
- Zkratky:
  - 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. 3-13: Znázornění tvorby PDU v jednotlivých vrstvách

2.1 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.

2.2 Aplikační vrstva (Application Layer)

- Aplikační vrstva především zpřístupňuje informačním systémům prostředí OSI. Protokoly aplikační vrstvy (7.) zajišťují komunikaci mezi aplikačními procesy ve spojení se správními funkcemi operačního systému, jež tyto procesy podporují. Aplikační vrstva poskytuje aplikačním procesům přístup ke komunikačním prostředkům. Lze říci, že u mnohých systémů je aplikační vrstva pouhým rozšířením běžných operačních systémů na požadavky síťového prostředí. Aplikační vrstva je funkčně velmi rozsáhlá. Aplikační vrstva poskytuje aplikacím např. následující služby: *přenos zpráv, identifikace komunikujících partnerů, zjištění informací o připravenosti komunikujícího partnera, dohoda o mechanizmech ochrany zpráv, ověření přípustnosti komunikovaných parametrů, určení přijatelné kvality poskytovaných služeb, zajištění synchronizace spolupracujících stran, výběr způsobu dialogu, dohoda o omezeních týkající se syntaxe zpráv* apod. K zabezpečení těchto služeb jsou samozřejmě využívány i služby nižších vrstev.
- Mezi nejznámější síťové aplikace patří např.:
    - **File transfer, access and management (FTAM**) pro přenos souborů nebo textu,
    - **Elektronická pošta** pro přenos elektronických dopisů,
    - **Virtual Terminal System**, tj. vzdálený přístup ke strojům,
    - **Remote Database Access** pro přístup do vzdálených databázových systémů.
- Tyto aplikace jsou však dnes využívány výhradně ve formě definované nad TCP/IP sadou.

2.3 Prezentační vrstva (Presentation Layer)

- Prezentační vrstva především koordinuje kódování a syntaxi vyměňovaných dat.
- Úlohou této vrstvy (6.) je příprava služeb pro aplikační vrstvu k interpretaci vyměňovaných dat (tyto vrstvy spolu těsně souvisí). Připravuje výběr ze standardních prezentací a jejich interpretací tak, aby komunikující aplikační procesy svoje data nejdříve transformovala do společného standardního formátu, přenesla je a nakonec je transformovala zpět. To má význam pouze tehdy, existuje-li shoda v syntaxi a sémantice. Standardní prezentace obsahuje informace, které popisují datové struktury a příkazy pro manipulaci s nimi.
- Existují tři verze syntaxe:
    - **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.
- Prezentační vrstva poskytuje aplikační vrstvě nezávislost na použité syntaxi (prezentaci) dat prostřednictvím služby **transformace** syntaxe do/z přenosové podoby. Tato transformace může spočívat např. v převodu kódů a abeced nebo modifikaci uspořádání dat. Podle modelu OSI je prezentační vrstva jedinou v modelu, která může zasahovat přímo do uživatelských dat. Mezi funkce, které může prezentační vrstva také poskytovat, patří i **komprese** nebo dekomprese datové části, dále **šifrování** nebo dešifrování.
- Služba transformace, která je připravena pro aplikační vrstvu, umožňuje uživatelským procesům komunikovat v heterogenním otevřeném prostředí bez nutnosti nepřiměřených požadavků na transformace.
- Od relační vrstvy vyžaduje vrstva č. 6 vytváření, provádění, dohled nad a ukončování logických spojení v této vrstvě – **relace** (Session). Během vytváření (session) se provádí inicializace protokolu mezi dvěma prezentačními pracovními jednotkami. Zřejmé je, že od relační vrstvy vyžaduje také vlastní přenos dat.
- Prezentační vrstva tedy provádí zejména tyto **úkony**:

- **Transformace kódování,**
- **Šifrování,**
- **Komprese.**

2.4 Relační vrstva (Session Layer)

- Relační vrstva především poskytuje informačním systémům nástroje pro řízení a synchronizaci jejich dialogu.
- Úkolem této vrstvy (5.) je podpora komunikace mezi spolupracujícími aplikačními procesy – organizovat a synchronizovat dialog, řídit výměnu dat mezi nimi. Služby můžeme rozdělit na vazbové služby pro dva aplikační procesy včetně služeb správy relace a na relační služby přenosu dat pro kontrolu výměny dat a synchronizaci.
- Prezentační entita může využívat i více než jedno relační spojení. Relační vrstva poskytuje prezentační následující **služby**: *vytváření a zavírání relačního spojení (relace), různý způsob přenosu zpráv (např. normální, spěšný), případně pozdržený přenos zpráv, řízení interakce, synchronizace relačního spojení, oznamování výjimečných stavů*.
- **Funkce** obsažené v relační vrstvě promítají relační spojení do transportního nebo i více transportních spojení. Obecně může během jedné relace vzniknout a zaniknout více spojení, případně může existovat i více transportních spojení najednou, které reprezentují jednu relaci.
- Při vytváření relace (navázání spojení) jsou vyšším vrstvám k dispozici prostředky pro navázání relace mezi dvěma aplikačními procesy. Během fáze přenosu dat může dojít k chybám. V takovém případě může rozpoznání a korekce chyb, které nebyly zachyceny některou nižší vrstvou, být opraveno na této úrovni, aniž by musela znovu proběhnout fáze navazování spojení.
- Od transportní vrstvy očekává vrstva relační možnost navázání spojení, provedení přenosu a jeho ukončení. Běžný tok dat musí proběhnout správně a bez chyb. V případě výskytu transportní vrstvou nekorigovatelných chyb o tom musí být relační vrstva informována.
- **Úkoly relační vrstvy** lze vyjádřit v následujících bodech:
    - **Řízení dialogu mezi aplikačními protokoly,**
    - **Synchronizace.**

2.5 Transportní vrstva (Transport Layer)

- Transportní vrstva především zvyšuje kvalitu spojů na požadovanou úroveň.
- Úlohou této vrstvy (4.) je příprava univerzální transportní obsluhy pro nejbližší vyšší vrstvu (relační) a správa pomocných prostředků nejbližší nižší vrstvy (síťové). Transportní vrstva osvobozuje svoji vyšší vrstvu (to jsou pracovní jednotky relační vrstvy) od nutnosti **určení optimální cesty, kontroly toku dat, problematiky přetížení či chyb na této úrovni** atd.
- Všechny protokoly, které jsou v této (a vyšší) vrstvě definovány, mají **koncový charakter** (end-to-end). Protokoly podřízených vrstev mohou být deklarovány mezi uzly, které jsou na cestě při přenosu zprávy od zdroje k příjemci. Protokoly vrstev 4 až 7 pracují vždy v koncových systémech, kterých se spojení týká. V rámci konstrukce síťové komunikace tedy musí existovat nějaké rozhraní, které bude realizovat přechod mezi specifickými vlastnostmi fyzických podsítí a aplikačními protokoly. A tím je právě transportní vrstva. **Transportní vrstva do určité míry vyrovnává rozdílné vlastnosti mezilehlých přenosových sítí a provádí koncové řízení**. Služby transportních datových jednotek jsou zprostředkovány od jednoho konce transportního spojení ke druhému pomocí prostředků nižších vrstev.
- Transportní vrstva poskytuje tyto služby: *transportní služba se spojením a bez spojení*. U transportní vrstvy se spojením můžeme rozlišit **tři fáze – navázání spojení, přenos dat, ukončení spojení**.
- Při navazovací fázi se provádějí zejména tyto funkce: *získání nejvhodnějšího síťového spojení, rozhodnutí o potřebě nebo vhodnosti multiplexování nebo rozvětvení s cílem optimálního využití síťového spojení, určení optimální délky jednotek na úrovni transportní vrstvy*.
- Ve fázi přenosu dat se jedná zejména o funkce: *uspořádání pořadí datových jednotek, segmentace a zřetězení (seskládání), multiplexování nebo rozvětvení, řízení toku dat, detekce a oprava chyb, identifikace transportních spojení*.

- V poslední fázi (ukončení spojení) se vykonávají tyto funkce: *oznámení důvodu ukončení a identifikace příslušného (ukončovaného) transportního spojení*.
- Mezi transportními entitami může být navázáno i **více transportních spojení**. Aby bylo možné odlišovat koncové body transportního spojení, rozlišují se pomocí identifikátorů (porty). Transportní spojení poskytují **duplexní** přenos mezi dvojicí transportních adres. Kvalita služeb, které budou při transportním spojení poskytovány, může být uživatelem zvolena a udržována během trvání tohoto spojení.
- **Kvalita služeb** transportního spojení **závisí na třídě služby**, kterou relační entity požadují. Třídy se rozlišují pomocí kombinace parametrů, jako jsou propustnost, zpoždění přenosu a chybovost. Nastavená kvalita je pak platná pro celé trvání transportního spojení.
- Části zpráv musí být odeslány na místo určení ve stejném pořadí, v jakém byly ze zdroje předány. Změna pořadí v rámci přenosu zpráv musí být u cíle řešena správným zřetězením.
- **Od síťové vrstvy očekává transportní úroveň primárně sestavení optimální cesty v síti**. Optimálnost se přitom řídí různými kritérii, která jsou dána sítí, její strukturou a způsobem propojení.
- **Funkce transportní vrstvy** v bodech jsou:
    - **Adresování konkrétní služby,**
    - **Segmentace a znovu-skládání dat,**
    - **Řízení spojení mezi komunikujícími aplikačními protokoly,**
    - **Řízení toku dat,**
    - **Řízení chybových stavů.**

2.6 Síťová vrstva (Network Layer)

- Síťová vrstva především směruje tok dat organizovaný do paketů.
- Úkolem síťové vrstvy (3.) je poskytnout síťové spojení prostřednictvím vytváření funkčních a procedurálních prostředků pro výměnu datových služeb mezi dvěma transportními jednotkami. Síťová vrstva je **zodpovědná za vlastní komunikaci v komplexní síti** (na základě logických adres), **směrování a přenos datových jednotek** (nejčastěji označovaných jako datagramů) od vysílače k přijímači. Utváří tak nezávislost transportní vrstvy na funkcích směrování a propojování (routing) – to je její základní **služba**. Mezi další služby patří síťové adresování, zahajování, průběh a uzavírání síťových spojů, identifikace koncových bodů síťového spojení, různé priority přenosu, oznamování vzniklých chyb, uspořádání pořadí přenášených fragmentů v paketech a další.
- Síťová služba může být v rámci OSI modelu buď **se spojením**, nebo **bez spojení**. V síťové vrstvě jsou sdruženy **funkce, které umožňují překlenout rozdílné vlastnosti dílčích úseků** trasy, které mohou být založeny na různých technologiích. Výsledkem je **konzistentní služba**.
- **Funkce směrování** se dá přirovnat k tomu, že síťová vrstva „vlastní mapu sítě“ nebo její určité části, pokud je celá síť příliš rozsáhlá. Pomáhá přitom vyhledávat partnery, kteří jsou určeni pouze logickými adresami.
- Aby se komunikace mohla uskutečnit, **od spojové vrstvy** očekává síťová vrstva **přípravu logických kanálů** mezi dvěma nebo více síťovými pracovními jednotkami. Datové kanály vrstvy č. 2 jsou vytvářeny a rušeny síťovými pracovními jednotkami. Z tohoto důvodu je pro síťovou vrstvu nezbytné disponovat **mechanizmem mapování logických adres na skutečné** (fyzické).
- **Úkoly síťové vrstvy** lze shrnout zejména do:
    - **Logické adresování,**
    - **Směrování mezi jednotlivými sítěmi.**

2.7 Spojová vrstva (Data Link Layer)

- Spojová vrstva především mění prostý tok bitů na spolehlivou cestu přenosu datových bloků – rámců.
- Spojová (2.) vrstva připravuje funkční a procedurální prostředky k vytvoření, udržení a rušení datových spojů mezi dvěma nebo více síťovými pracovními jednotkami. Spojová vrstva vždy propojuje dva přímo sousedící komunikační systémy, přičemž počet spojení na této vrstvě může být vyšší než jedno - datové spojení je tvořeno jedním nebo více reálnými nebo virtuálními datovými obvody. Tato spojení vznikají a zanikají **dynamicky**, tj. dle potřeby.

- Spojová vrstva musí umožňovat **tyto operace**: *zahajování, udržování a uzavírání vytvořených spojení, rozvětvení datových spojení, formátování rámců, identifikace koncových bodů spojení, seřazování přenášených rámců, detekce a oprava chyb, oznamování chyb, které není schopna opravit. Dále řízení po fyzických okruzích, synchronizace, fyzická adresace.*
- 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*). Podvrstva LLC poskytuje rozhraní mezi konkrétním přenosovým prostředkem a síťovou vrstvou, podvrstva MAC poskytuje služby specifické pro daný přenosový prostředek.
- Implementace spojové vrstvy je závislá na druhu sítě (technologie a s ní související topologie).
- Popis problematiky **spojové vrstvy lze shrnout** do pojmů:
    - **Vytváření rámců,**
    - **Adresování v rámci dané sítě,**
    - **Řízení toku dat,**
    - **Řízení chybových stavů,**
    - **Přístupové metody ke sdílenému médiu.**

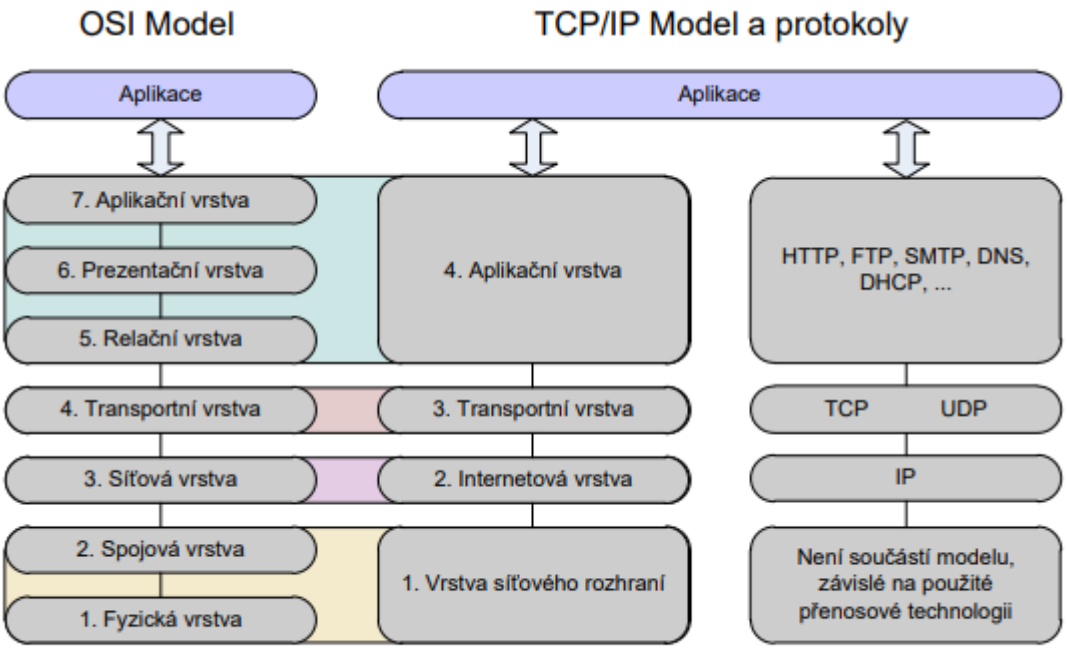
2.8 Fyzická vrstva (Physical Layer)

- Fyzická vrstva především přenáší prostý tok bitů přenosovým médiem.
- Úkolem této vrstvy (1.) je uzpůsobení dat získaných od spojové vrstvy do podoby bitového toku, což je často elektrický signál s měnícími se napěťovými úrovněmi. Zajištění těchto funkcí obnáší přípravu funkčních, procedurálních, mechanických, elektrických a elektronických prostředků pro vytvoření, udržení a ukončení datových okruhů reálného nebo virtuálního typu mezi datovými koncovými a síťovými zařízeními.
- **Fyzické spojení** může být vytvořené formou propojení **datových okruhů**, přičemž datový okruh představuje **komunikační cestu** po fyzickém médiu mezi dvěma entitami **a prostředky potřebné pro provedení přenosu** bitů touto cestou. Fyzické spojení může umožňovat *plný duplex* nebo pouze *poloduplex*, může být *dvoubodové* nebo *mnohabodové*.
- Fyzická vrstva poskytuje kromě vytváření a ukončování fyzických spojení také další služby: *vytváření datových jednotek, identifikace datových okruhů, řazení bitů a oznamování poruchových stavů*. Kvalita fyzického spojení bývá popisována **chybovostí**.
- Jednoduše řečeno - bity musí projít médiem. **Hlavní funkcí fyzické vrstvy je tedy přenos bitů**. K tomu je musíme přizpůsobit, vyslat na médium a na vhodném místě převzít zpět do počítače, a to v čitelné formě. Fyzická vrstva se tedy musí vypořádávat s charakteristikami přenosových prostředků – **použité kódování dat, napěťové úrovně, rozhraní** apod.
- Stručně lze **záležitosti**, kterými se zabývá **fyzická vrstva**, popsat těmito body:
    - **Fyzické charakteristiky médií a rozhraní,**
    - **Reprezentace bitů,**
    - **Přenosová rychlost,**
    - **Synchronizace mezi vysílačem a příjemcem,**
    - **Přizpůsobení se charakteru kanálu a topologii sítě (bod-bod, vícebodová),**
    - **Přenosový režim z hlediska oboustrannosti komunikace.**

3 Základní popis síťového modelu TCP/IP a srovnání s ISO/OSI.

3.1 Vazba mezi RM OSI a modelem TCP/IP

- Soustava protokolů TCP/IP je původně „rivalem“ obecného sedmivrstvého referenčního modelu ISO/OSI. TCP/IP je obvykle chápáno jen jako 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 rela**tivně 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** 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 *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.



Obr. 3-14: Srovnání modelu RM ISO/OSI a TCP/IP

3.2 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.

3.3 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.

3.4 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 spolehlivost přenosu a také měnit nespojovaný charakter přenosu (v síťové vrstvě) na spojovaný. Přestože je transportní vrstva TCP/IP nejčastěji zajišťována právě protokolem TCP, není to zdaleka jediná možnost. Dalším používaným protokolem na úrovni transportní vrstvy je např. protokol **UDP (*User Datagram Protocol***), který na rozdíl od TCP nezajišťuje (mimo jiné) spolehlivost přenosu. Samozřejmě je využíván pouze aplikacemi, které si spolehlivost na úrovni transportní vrstvy nepřejí nebo nemohou dovolit, příkladem je přenos multimediálních dat v reálném čase.

3.5 Aplikační vrstva (Application Layer)

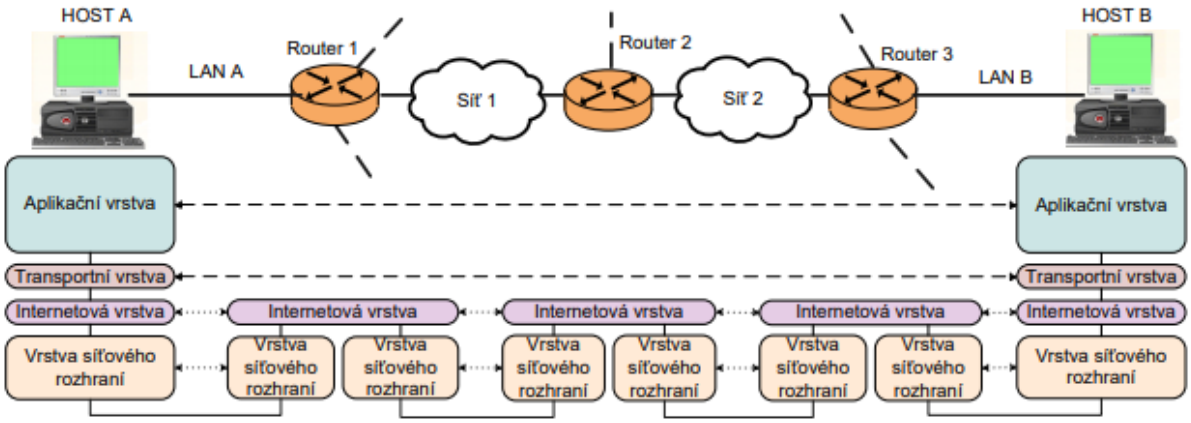
- Nejvyšší vrstva, jejími entitami jsou jednotlivé aplikační programy, které na rozdíl od referenčního modelu ISO/OSI komunikují přímo s transportní vrstvou. Případné prezentační a relační služby, které v modelu ISO/OSI zajišťují samostatné vrstvy, si zde musí jednotlivé aplikace realizovat samy (pokud 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.

3.6 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. 3-15 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.)



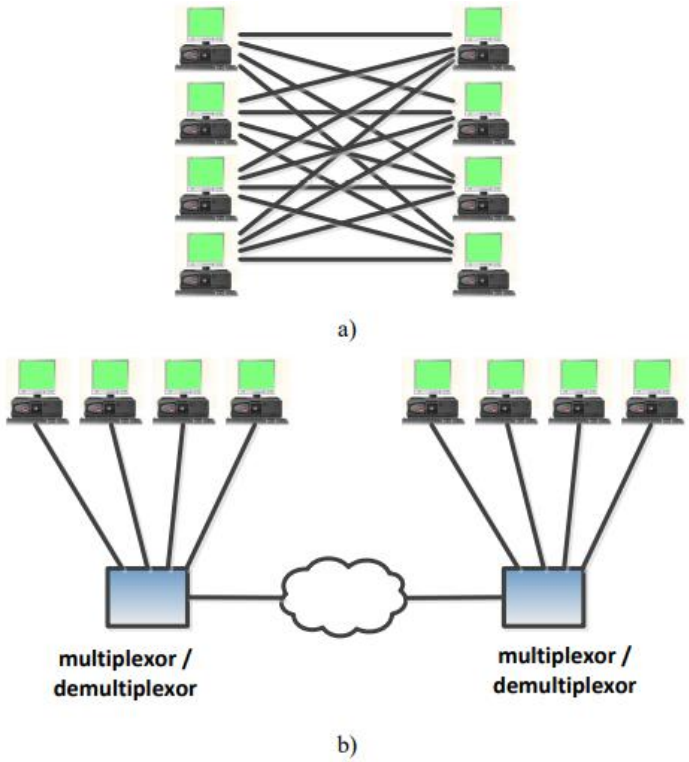
Obr. 3-15: Ukázka propojení sítí v rámci Internetu

- 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. 3-15, 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.

4 Principy komunikačních technik - vícenásobné využití cest, zajištění obousměrné komunikace

4.1 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. 4-1, 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.



Obr. 4-1: 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.

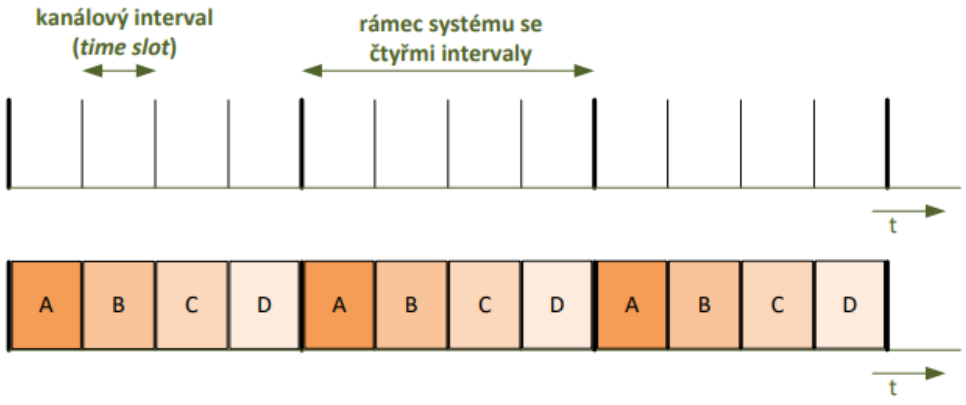
- **Vlnové dělení** (vlnový multiplex), anglicky WDM (Wavelength-Division Multiplex). WDM představuje variantu kmitočtového dělení používanou v optice. Základní charakteristikou je, že se do jednoho optického vlákna multiplexuje více signálů, které jsou odlišeny svoji vlnovou délkou (tj. barvou).
- **Časové 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.
- **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.
- 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

4.1.1 Časové dělení

- 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. 3-3. Synchronní přenosový režim se využívá např. v přístupové části GSM síti, ale též v přenosovém systému PCM (Pulse-Code Modulation). 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.

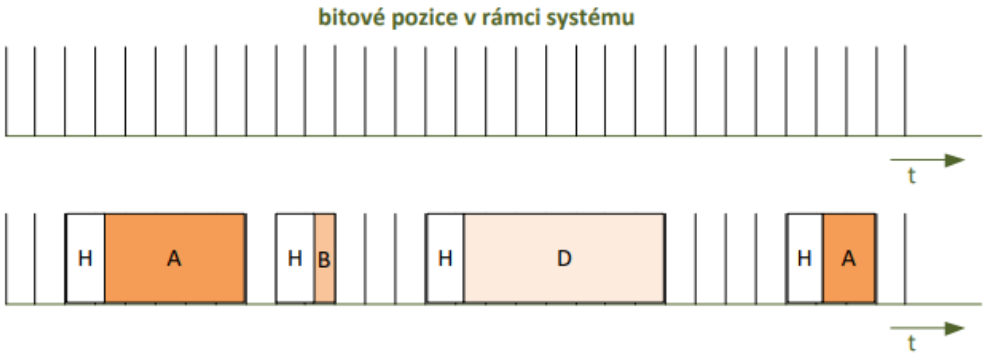


Obr. 4-2: Příklad synchronního přenosového módu – rovnoměrné rozdělení kapacity mezi čtyři stanice

Přenosový režim paketů

- Tento režim odpovídá komutaci paketů z Obr. 3-3 a připouští proměnnou délku zpráv a tedy i nerovnoměrné rozdělení kapacity mezi vysílací stanice. Zprávy jsou odesílány tehdy, existuje-li k tomu

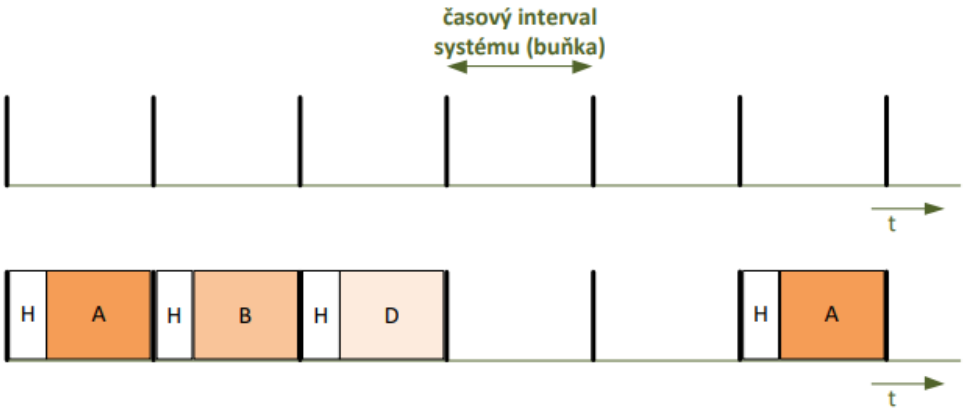
požadavek. Každá zpráva musí obsahovat řídicí záhlaví, jelikož není předem dáno, komu patří. Tento režim je používán zcela běžně v současných datových sítích. Systém je flexibilnější než synchronní režim, avšak bez dalších mechanizmů nezajišťuje vysílacím stanicím žádnou přenosovou kapacitu, jelikož ta může být blokována jinými stanicemi.



Obr. 4-3: Příklad paketového přenosového módu – čtyři stanice (C nevysílá), různá velikost jednotek se záhlavím (značeno H), libovolná bitová pozice

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. 4-4.



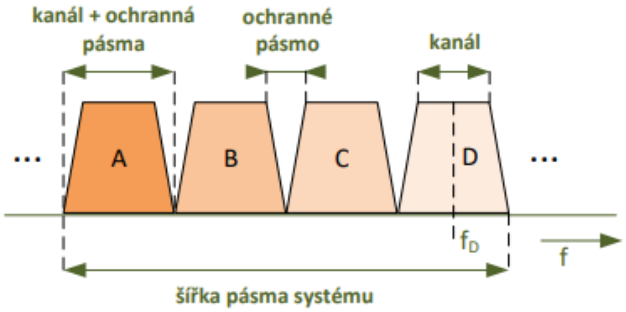
Obr. 4-4: Příklad asynchronního přenosového módu – čtyři stanice (C nevysílá), jednotky umístěné v libovolném intervalu, se záhlavím, a vždy se stejnou velikostí rámce

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. 4-5. 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. 4-5: Princip kmitočtového dělení do kanálů a ochranných pásem

4.2 Metody zajištění obousměrné komunikace

- Existují celkem dva základní typy spojení či provozu z hlediska obousměrnosti. Jsou to simplexní spojení (simplex) a duplexní spojení (duplex).
- **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 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. 4.1.1 a 4.1.2), 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 - přenosová média, analogové a digitální modulace, klíčovací techniky, princip digitalizace řečového signálu.

5.1 Přenosová média

- 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)

- Ve všech výše uvedených případech je vlastní přenos realizován pomocí elektromagnetických vln, avšak v různých kmitočtových pásmech.
- Podle typu média se zpravidla liší i to, jak vypadá přenášený signál.

5.1.1 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 Herzích, 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.
- **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.

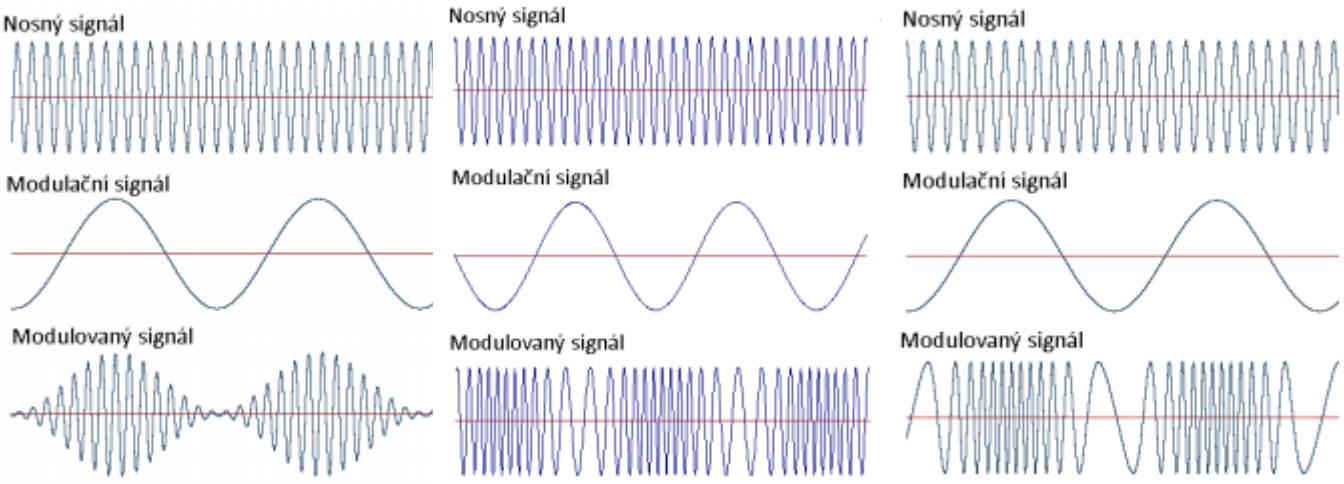
5.2 Analogové modulace

- 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).

- **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).

- **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.



5.3 Digitální modulace

- 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)

5.3.1 Amplitudové klíčování (ASK)

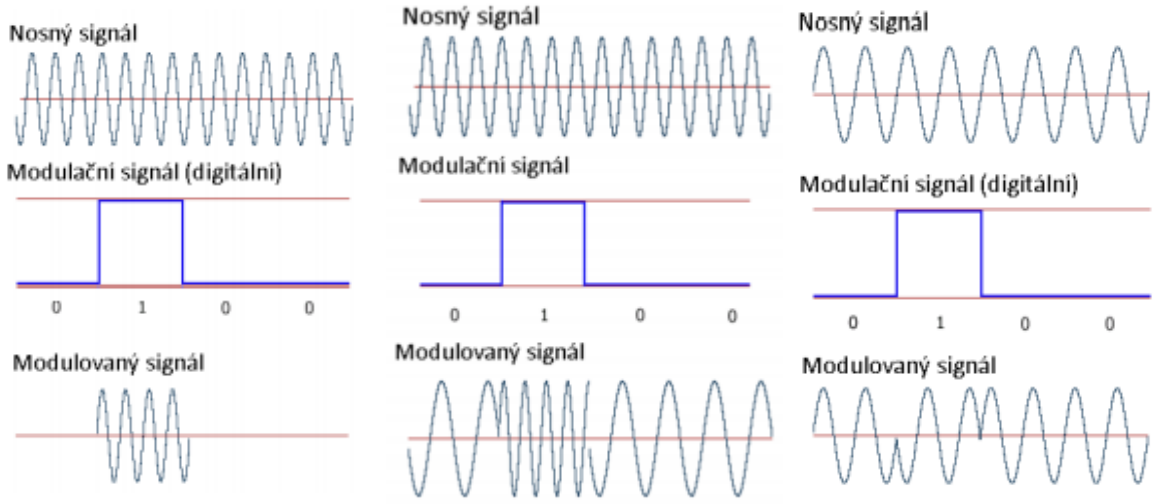
- Tato velice jednoduchá technika spočívá v tom, že modulační signál střídavě spíná a vypíná nosný signál, podle toho, zda je právě modulována hodnota „1“ nebo hodnota „0“. Princip je ilustrován na Obr. 5-10. V této nejjednodušší podobě se ASK příliš nepoužívá, protože nemá moc výhodné vlastnosti, s 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.

5.3.2 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.

5.3.3 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.



5.3.4 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

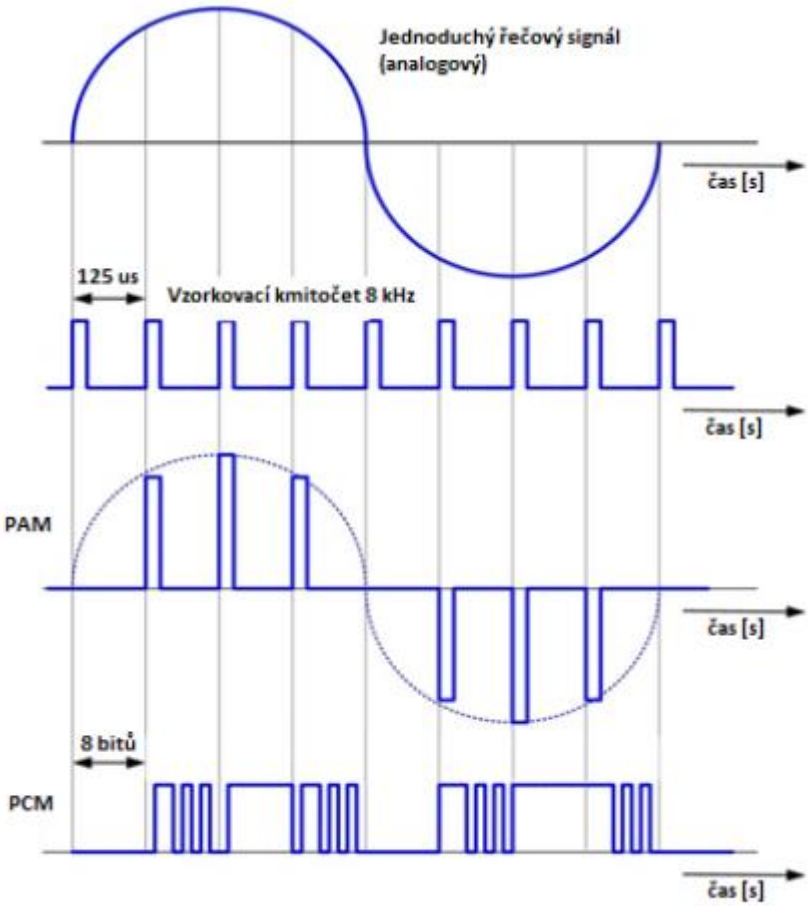
se snižuje odolnost přenosu vůči chybám, rušení apod. V názvech vícestavových modulací se objevuje počet stavu, se kterým technika pracuje, např. 8-PSK (nejčastěji jsou vícestavové modulace využívány právě u fázových modulací).

5.3.5 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.

5.4 Digitalizace řečového signálu

- 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 diskrétní hodnoty 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.



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

6 Spojová vrstva přenosových systémů - podvrstvy, rámce spojové vrstvy, adresace, metody zajištění spolehlivého přenosu

6.1 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).

6.1.1 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).

6.1.2 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).

6.2 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
        - **adresy** – zdrojová a cílová, identifikace komunikujících uzlů.
    - 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. 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.

6.2.1 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:

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

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

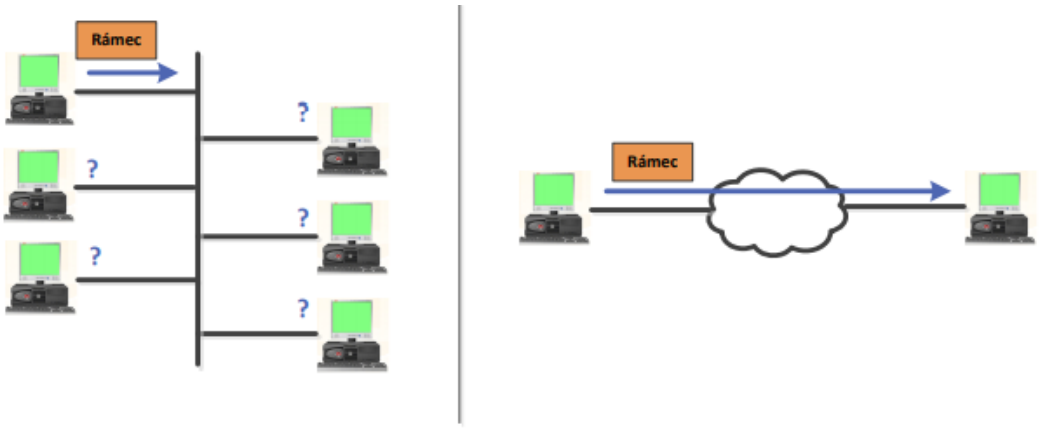
- S oběma rámci se lze dnes běžně potkat v rámci jedné sítě, přestože běžnější je spíše rámec Ethernet II. Rozdíl mezi rámci je především u pole, které je v obrázcích zaznačeno zelenou barvou – tj. „délka“ u IEEE 802.3 a „typ“ u Ethernet II. Pole „délka“ podle očekávání informuje o tom, jak dlouhá je 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 (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.

Další rámce:

- Rámec protokolů Bisync a PPP
- Rámec protokolu HDLC
- Rámec technologie ATM
- Rámec technologie Frame Relay

6.3 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, 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. 6-10a.



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. 6-10b). Nicméně i v těchto případech se můžeme setkat s rámci, které adresování obsahují (viz kap. 6.4), např. z důvodu větší univerzálnosti daného protokolu.

6.3.1 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.

6.4 Techniky detekce chyb

6.4.1 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 bitů13. 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
- 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.

6.4.2 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.
- 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.

6.4.3 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 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. 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.

6.5 Spolehlivý přenos

6.5.1 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.
- 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:

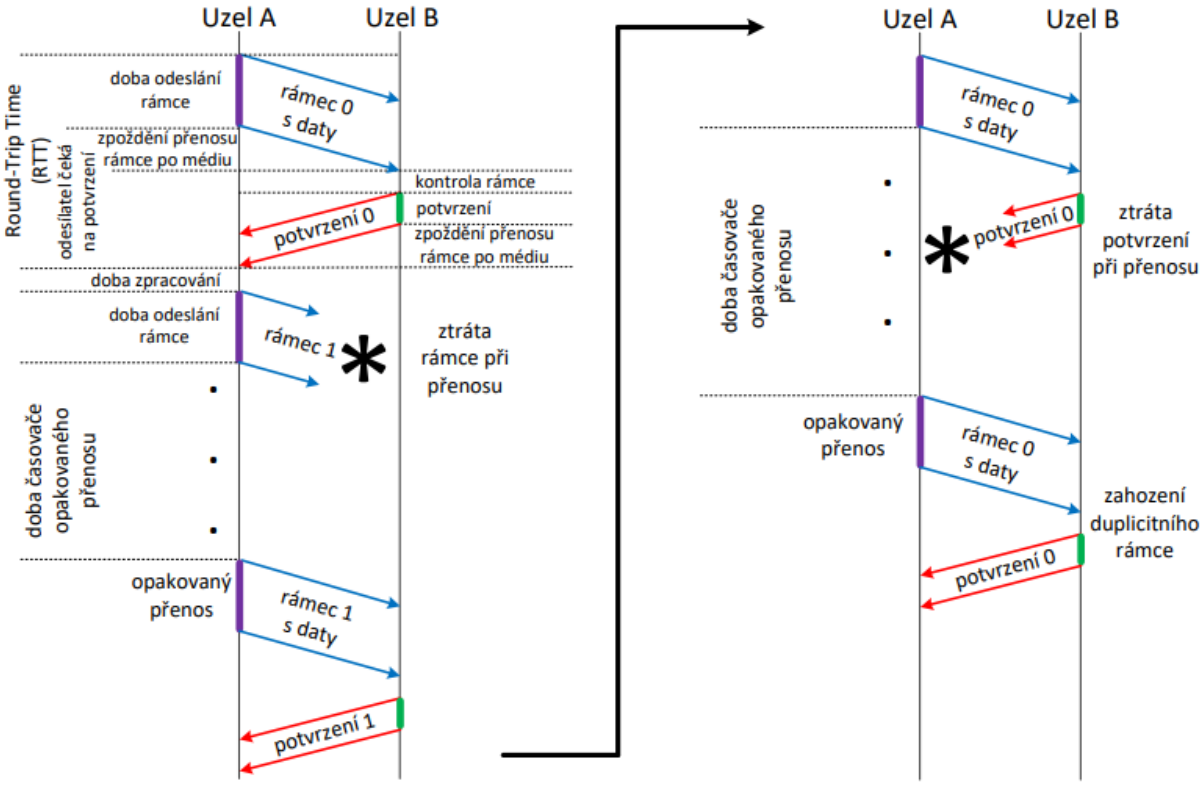
- **Velikost vyrovnávací paměti příjemce není neomezená** (počet rámců, které je možné vyslat pro určitého příjemce v rámci jednotky času, je omezen),
- **Delší rámec = větší pravděpodobnost výskytu chyb**, riziko opakovaného přenosu rámce roste,
- **Kratší rámec = rychlejší detekce chyb**, což také 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 (ztraceny nebo poškozeny), jsou ze strany vysílače přeneseny opakovaně. K opakování dochází až po vypršení určitého časového intervalu (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, Go-Back-N ARQ a Selective Repeat ARQ.

6.5.2 Stop-and-wait ARQ

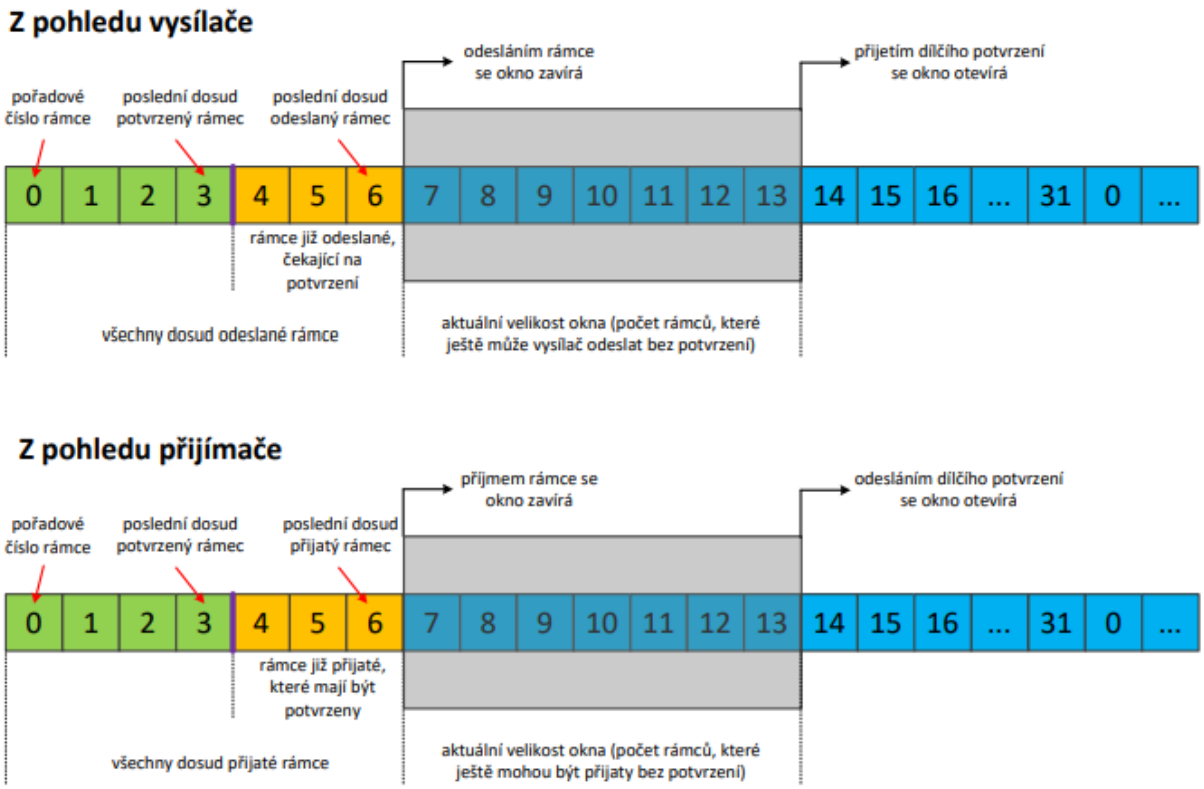
- Tato metoda je tím nejjednodušším, co si lze v případě řízení toku dat představit. Vysílač a přijímač pracují sekvenčně. Vysílač odešle rámec a následně čeká, dokud nemá od příjemce potvrzení o přijetí (acknowledgment). Pokud při přenosu nedošlo k žádné chybě, je toto potvrzení bráno zároveň jako signál, 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 typicky pomocí CRC. Vysílač musí také pracovat s určitým časovačem (timeout), pokud dojde ke ztrátě celého rámce (ať už původního nebo potvrzujícího).
- 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 stopand-wait je tedy nutné vyhradit jeden bit pro tento účel.
- Z obrázku 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.



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

6.5.3 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). 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.
- V případě využití této techniky musí být rámce taktéž nějak číslovány a je zřejmé, že již nevystačíme pouze s jedním bitem. Pole musí být vícebitové. 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 překročí 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.



Obr. 6-12: Postup komunikace při využití techniky klouzavého okna z pohledu vysílače i přijímače

6.5.4 Metoda Go-back-N ARQ

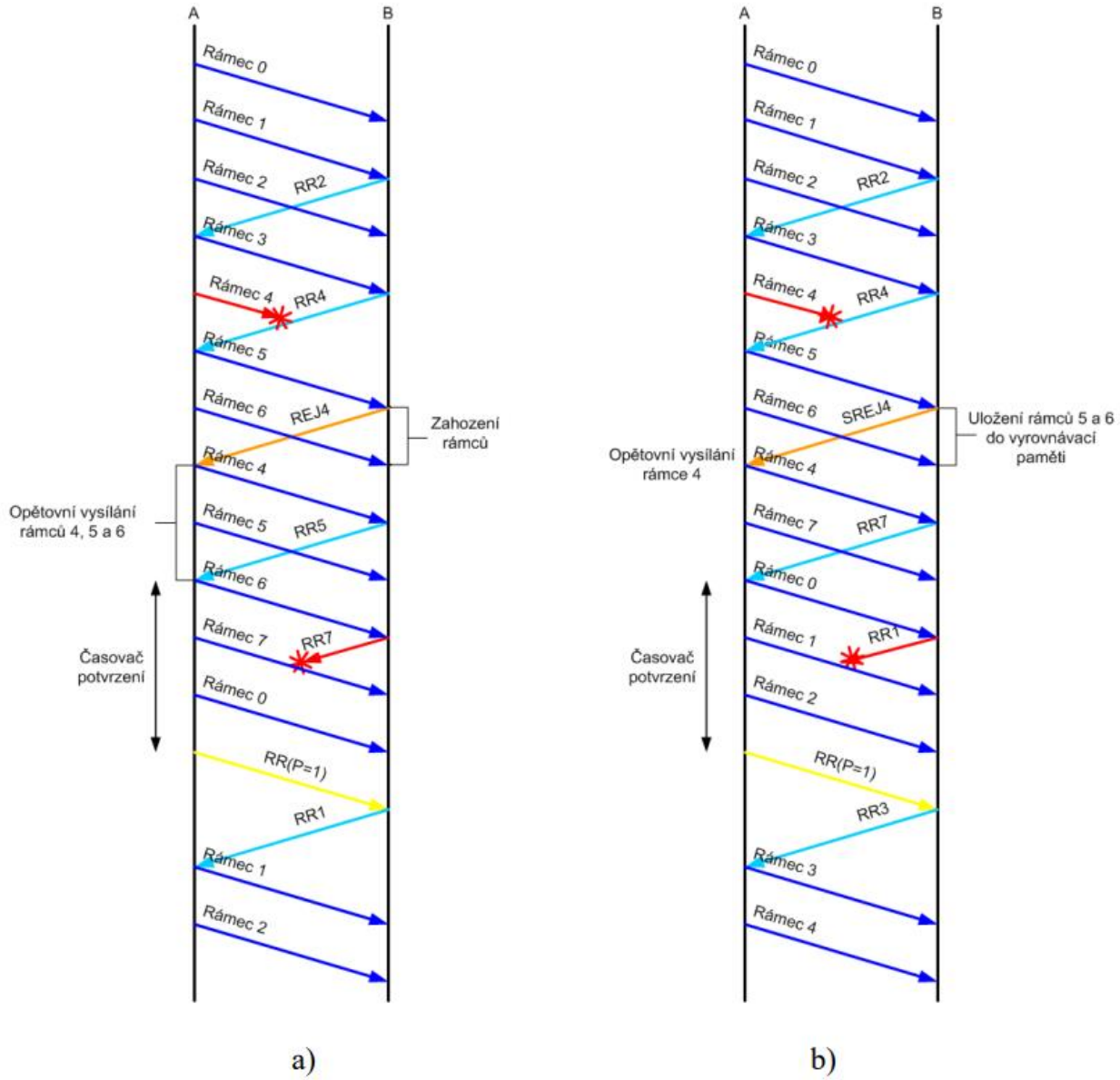
- Tato metoda využívá jako základní mechanizmus techniku klouzavého okna tak, jak byla popsána v předcházející kapitole. Jak vyplývá z názvu, metoda dále pracuje s návratem do určitého stavu. Jak bude vysvětleno dále, jedná se o návrat do stavu, kdy došlo při přenosu k chybě. V praxi je možné se s tou 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 přenosovém systému s malým množstvím chyb 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, v tomto případě je to však rámec, který již byl přijat, avšak s chybou, a proto je třeba, aby byl přenesen znovu.

6.5.5 Metoda Selective Repeat ARQ

- Tato metoda řeší v principu stejné situace jako v předcházející kapitole popsaná Goback-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, přesto je v praxi pro svou jednoduchost běžnější metoda Go-back-N ARQ.
- Zpráva, kterou příjemce zasílá poté, co zjistí, že rámec je porušen, se obvykle nazývá Selective Reject (SREJ) 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.



Obr. 6-13: Ukázka fungování mechanizmu a) Go-back-N ARQ, b) Selective Repeat ARQ (různé situace)

6.5.6 Technika klouzavého okna a řízení toku

- 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. 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.

7 Síťová vrstva přenosových systémů - spínání paketů, služby síťové vrstvy, IPv4 adresy, techniky směrování, IPv4 datagram

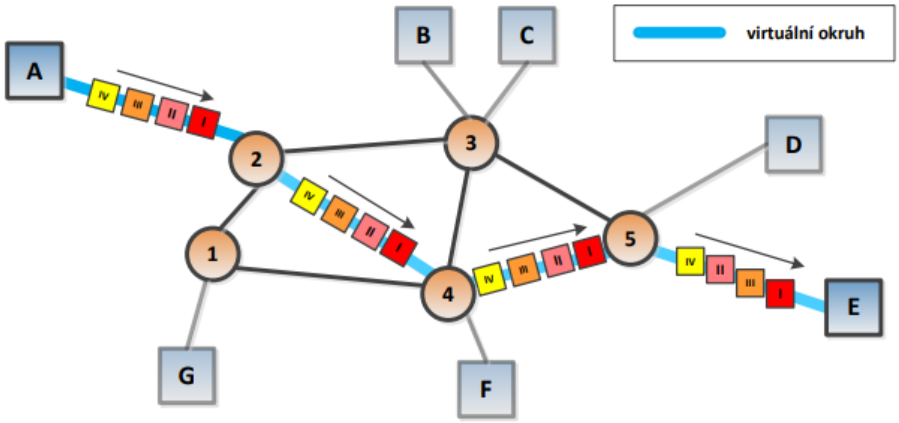
7.1 Přepojování paketů

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. 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. 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.

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.

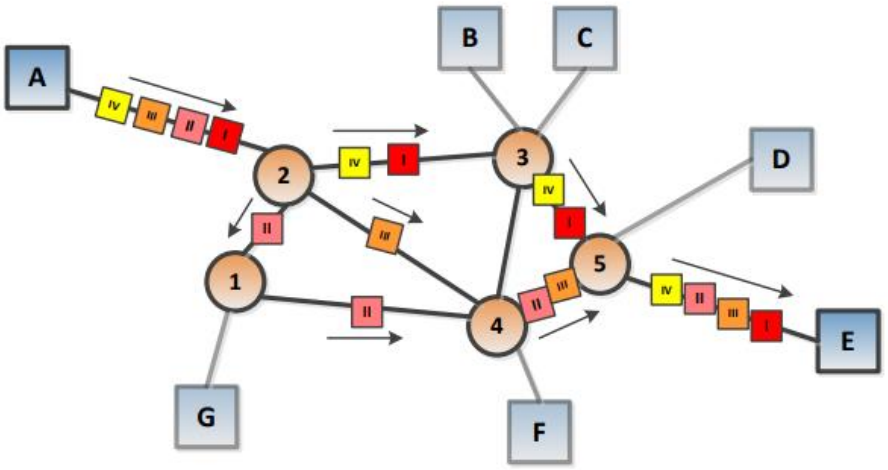


Obr. 7-2: Ukázka přepojování paketů s vytvářením virtuálního okruhu (služba se spojením)

  - 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.



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

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.
- Velikost paketu má na zpoždění komunikace podstatný vliv.

7.2 Služby síťové vrstvy

7.2.1 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.
    - **vyhledávání logické adresy dalšího uzlu (skoku) směrem k cíli** – datagram obsahuje adresu zdroje a cíle, nicméně 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
    - **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
    - **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.
- 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.

7.2.2 Základní služby síťové vrstvy poskytované na každém směrovači

- Síťová vrstva na každém mezilehlém uzlu musí spolupracovat s dvěma spojovými vrstvami. 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).

7.2.3 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)

7.2.4 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. 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.
    - **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.
    - **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)

7.2.5 Služby síťové vrstvy poskytované 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.

7.2.6 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 Adresy síťové vrstvy u IPv4 protokolu

- V sadě TCP/IP má každé zařízení v libovolné síti svoji vlastní logickou adresu, která bývá nejčastěji označována jako IP adresa. 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. 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.

7.3.1 Zápis IP adres

- Počítače pracují zejména s binární reprezentací dat. 32-bitové 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.

- 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.

7.3.2 Maska sítě

- 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)

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

7.3.3 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.
- 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. 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

7.3.4 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-tých 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

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		

7.4 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).

STRATEGIE SMĚROVÁNÍ NEDYNAMICKÉHO CHARAKTERU

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.

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.

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).

STRATEGIE SMĚROVÁNÍ 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.

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.

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

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.

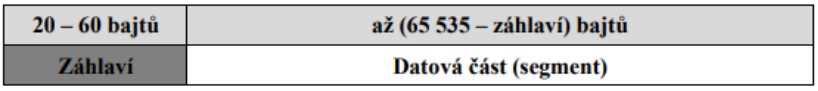
7.4.1 Fungování směrování v sítích TCP/IP

- **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. 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.

- 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.

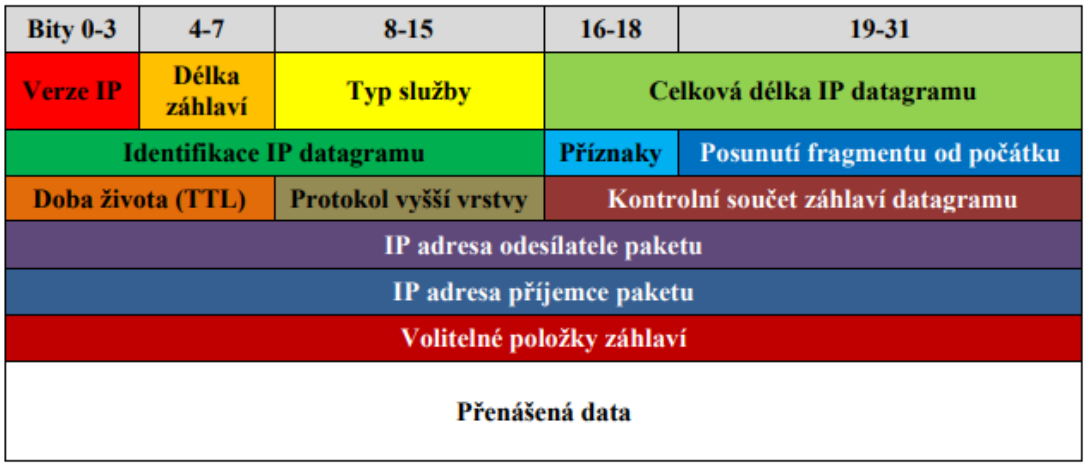
7.5 IPv4 datagramy

- 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.



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

- **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. Mi**nimální 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
- **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,

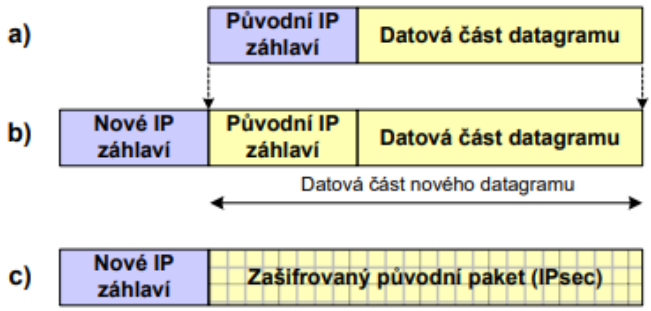
která nejlépe odpovídá požadavkům datagramu. Využití pole v praxi je sporadické, můžeme se setkat s tím, že se položka používá k podobným účelům – nese značku pro mechanizmy zajišťující služby s definovanou kvalitou služby (QoS).

- **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. Vždy přiděleno odesílatelem a hodnota se nemění.
- **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. MFbit (more fragments) říká, že datagram byl fragmentován a že bude následovat d
- **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.
- **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.

8 Síťová vrstva přenosových systémů - tunelování paketů, ARP, NAT, ICMPv4, IPv6.

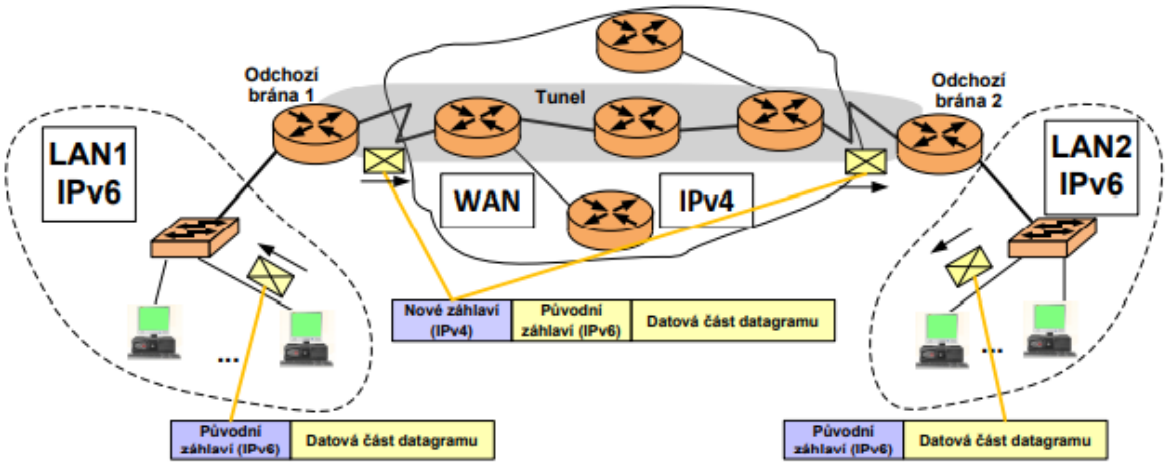
8.1 Tunelování paketů

- 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. 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.



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. Tento typ tunelování představuje jeden z přechodových mechanizmů na protokol IPv6.



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

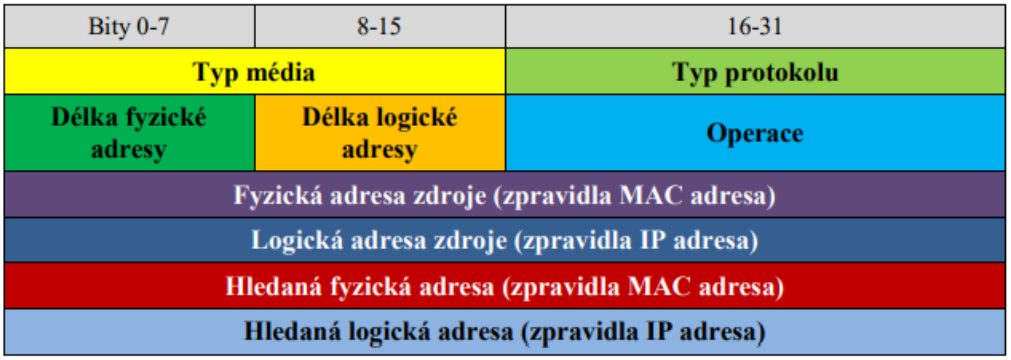
8.2 Address Resolution Protocol (ARP)

- 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).

8.2.1 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**



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

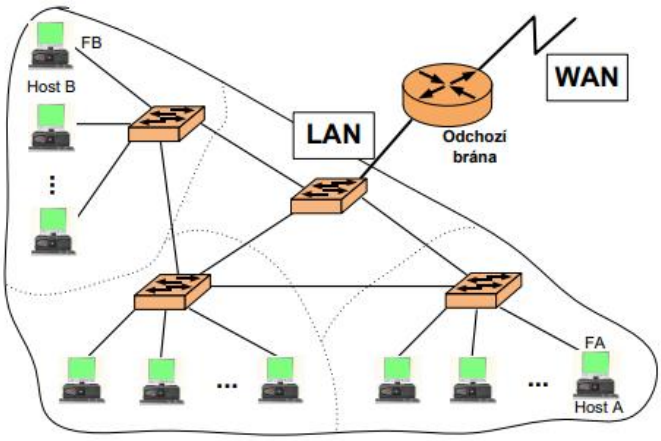
- **Typ média** (Hardware type) – 16 bitů délky, hodnota indikující typ použitého média, resp. spojové technologie, např. pro Ethernet je hodnota 0x0001, ATM má 0x0010.
- **Typ protokolu** (Protocol type) – 16 bitů, hodnota indikuje typ vyššího protokolu, v rámci něhož se logická adresa používá, pro IP je hodnota 0x8000.
- **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.

- 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. 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.

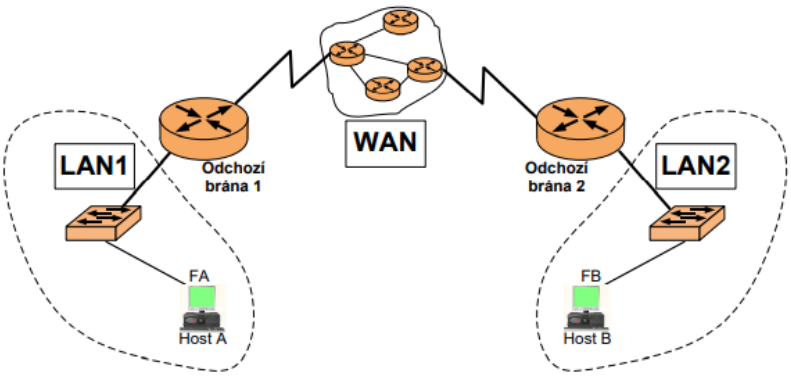
počítači B), musí být schopna zajistit převod IP adresy (IB) na fyzickou adresu (FB). Tento údaj je nezbytný, aby mohl být vytvořen rámec.

- 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
- 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)

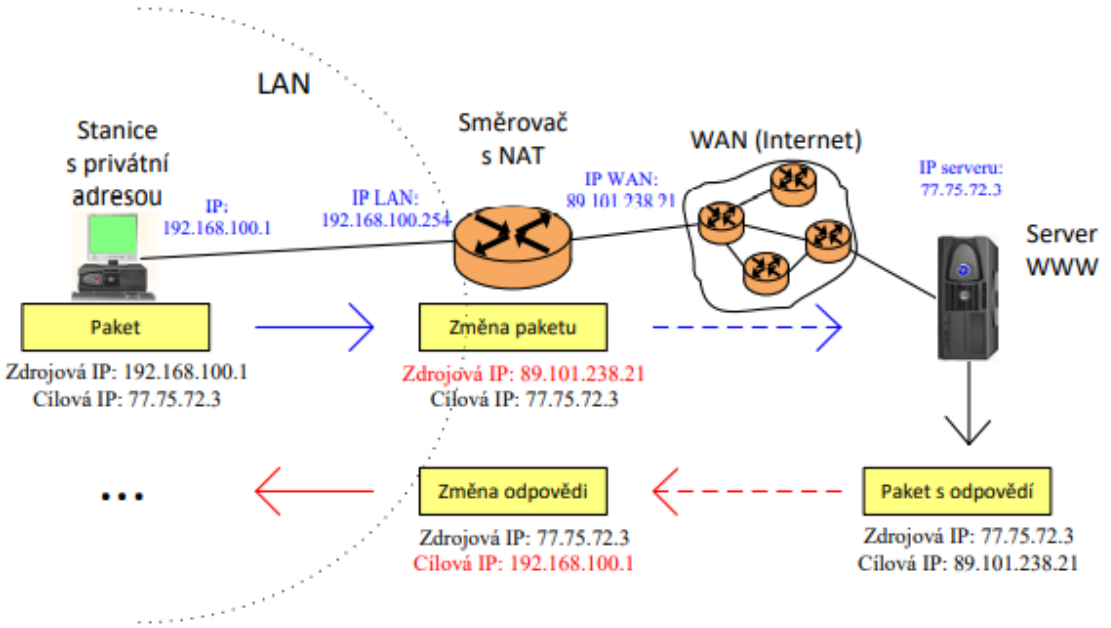


Obr. 7-27: Ilustrace situace kdy zdrojová a hledaná stanice nejsou na stejné síti

- 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 důvodů 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.

8.3 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. 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. 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**.
- 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. Komunikace může být iniciována vždy pouze počítačem z vnitřní sítě.



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

8.3.1 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.

8.3.2 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 NATu 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.

8.4 Internet Control Message Protocol verze 4 (ICMPv4)

8.4.1 Základní popis protokolu

- IP protokol představuje základní protokol síťové vrstvy, který je široce využíván pro přenos paketů. Z popisu protokolu IPv4 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 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

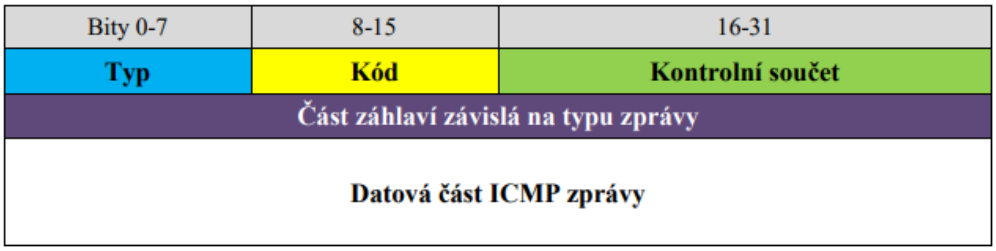
Obr. 7-31: Zapouzdření ICMP paketu přenášeného v síti Ethernet

- 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).

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. 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.



Obr. 7-32: Obecný formát ICMPv4 zprávy

8.4.2 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).

8.4.3 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).

8.5 Internet Protocol verze 6 (IPv6)

8.5.1 Základní vlastnosti IPv6

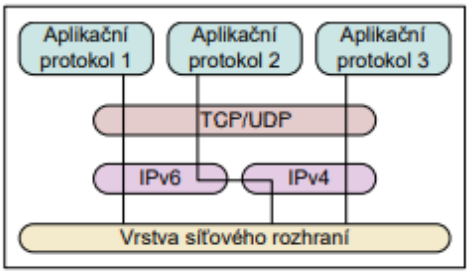
- Kromě rozšíření adresního prostoru došlo k diskuzi nad dalšími vlastnostmi IPv6 (které je s IPv4 nekompatibilní), jako např.:
  - zjednodušení formátu záhlaví – méně povinných položek
  - snaha o zredukování velikosti směrovacích tabulek globální úrovně ve směrovačích,
  - malé snížení hodnoty zpoždění 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),
  - rozšíření adresního prostoru, z 32 bitů na 128 bitů, tedy z 2^{32} adres na 2^{128} adres

8.5.2 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.

8.5.3 Zavádění IPv6

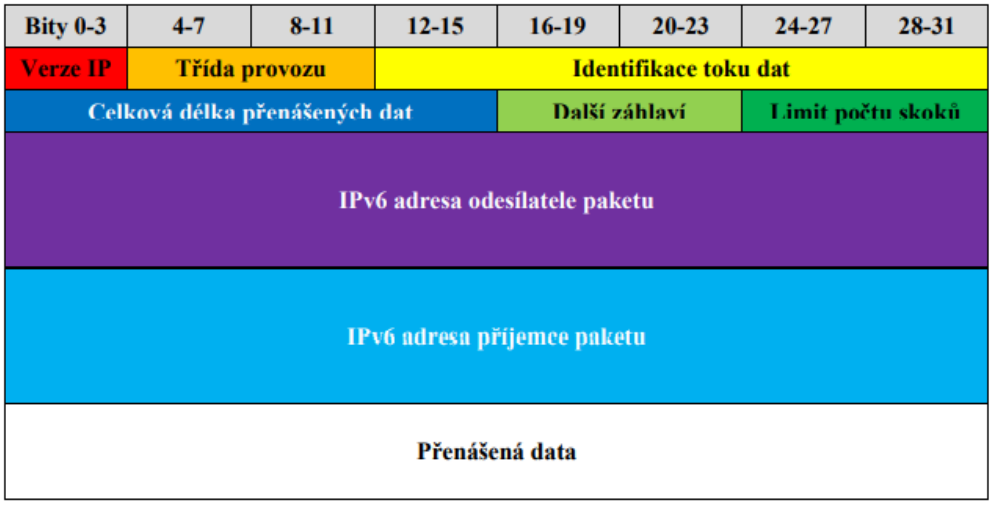
- Základní překážkou v rychlém zavádění IPv6 je především jeho nekompatibilita s IPv4. Bylo proto navrženo několik mechanizmů umožňujících hladký přechod od IPv4 k protokolu IPv6. Souvisí zejména s následujícími technikami:
  - **Souběh Internetových protokolů IPv6 a IPv4** (dual stack) – software a hardware podporuje plně oboje. To samozřejmě vede k zvýšení nákladů na vývoj zařízení, ladění a tím pádem i koncovou cenu. 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. Technika umožňuje komunikaci přes sítě s odlišnou verzí protokolu IP
  - **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)



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

8.5.4 IPv6 datagramy (pakety)

- Struktura základního záhlaví IPv6 není stejná jako ta u IPv4.
    - **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 (zpravidla 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.
- 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

8.5.5 Způsoby adresování

- 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**

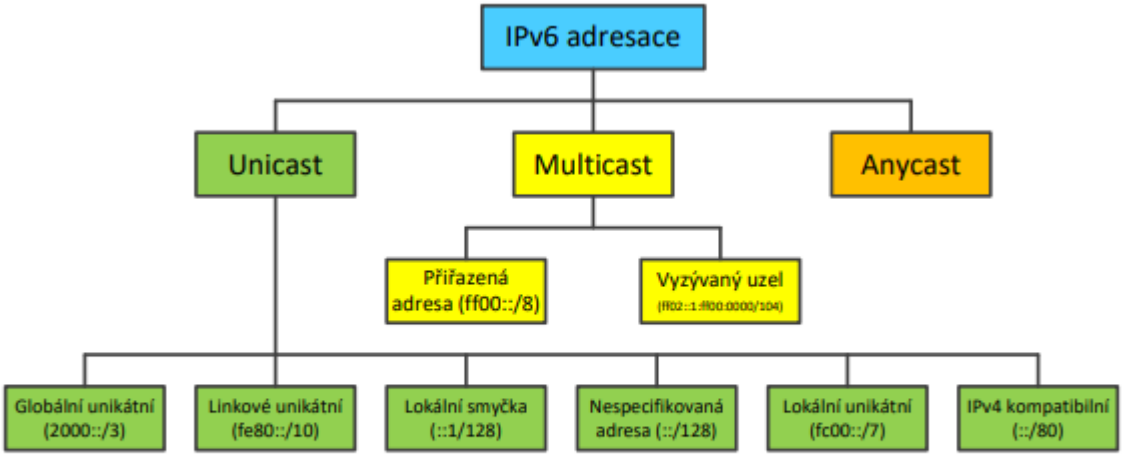
(broadcast) adresy, které nejsou v rámci IPv6 definovány samostatně. V rámci adresního prostoru jsou definovány i některé speciální skupiny.

- **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.

8.5.6 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
- Jestliže adresa obsahuje mnoho nul (souvislý blok), je umožněno užívat zkrácený zápis: 8000::0ABC:DEF1:0345:789A

8.5.7 Typy adres



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

9 Transportní vrstva přenosových systémů - služby transportní vrstvy, UDP protokol, TCP protokol

9.1 Služby transportní vrstvy

9.1.1 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 (endto-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. 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.

9.1.2 Adresování na transportní vrstvě

- Z hlediska organizace komunikace existuje více možných variant, nicméně nejčastější je využití systému klient-server. 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 i TCP) zavádí 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. 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

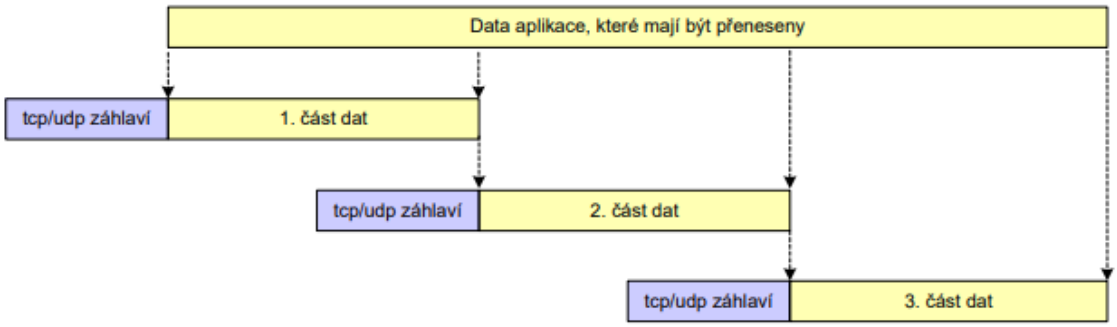
Tab. 9: Základní dělení portů

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

- 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

9.1.3 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.



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

- 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.

9.1.4 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.

9.1.5 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.

- 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 transportní vrstvy. U dvou nejvýznamnějších protokolů transportní vrstvy bude pojednáno o tom, jak k řešení této důležité problematiky přistupují.

9.2 User Datagram Protocol (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.

9.2.1 Datagram protokolu UDP

- Záhlaví UDP je maximálně jednoduché, jak je patrné z Obr. 8-6, sestává pouze ze čtyř 16-ti bitových 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. Pokud je odesílatel server, je číslo portu zpravidla dáno dle typu služby
- **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.

9.2.2 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.

9.2.3 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 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.
- Typickým 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.
- Dalším 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.

9.3 Transmission Control Protocol (TCP)

9.3.1 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
    - **Spolehlivý přenos dat** – TCP používá potvrzovací mechanizmy, které umožňují ověřit, že došlo k úspěšnému přenosu.

9.3.2 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.

9.3.3 Segment protokolu TCP

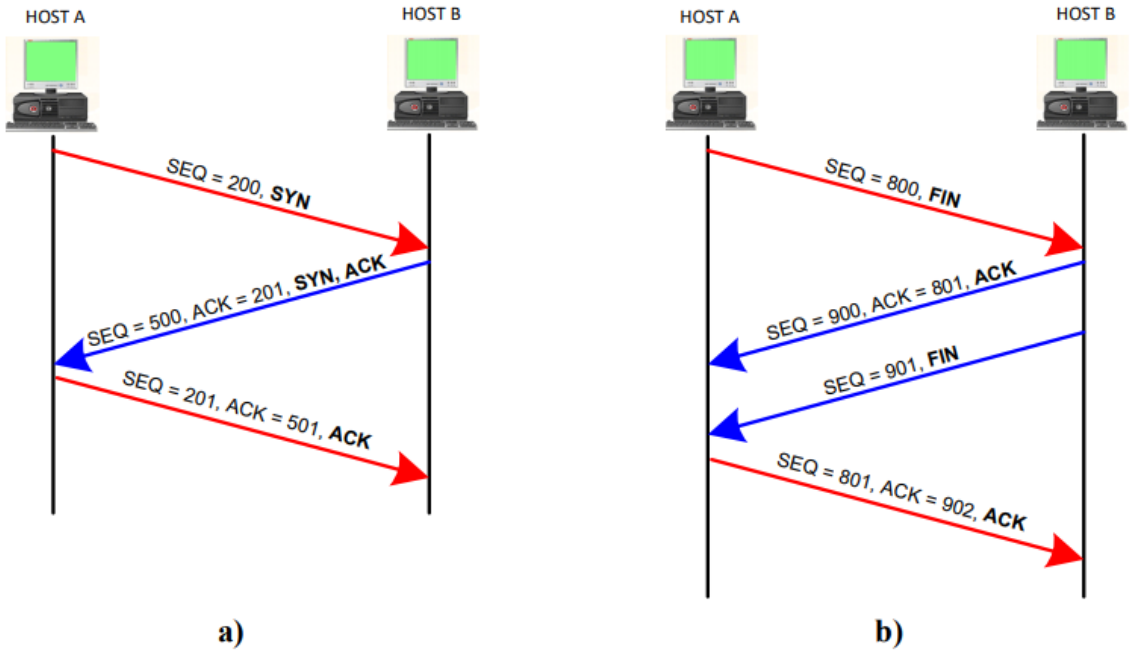
- Vzhledem k tomu, jaké funkce má protokol TCP zajišťovat, je záhlaví jednotky tohoto protokolu (segmentu) výrazně obsáhlejší než u protokolu UDP
  - **Zdrojový port** (source port) – hodnota indikuje port na straně odesílatele segmentu, obdobně jako u UDP.
  - **Cílový port** (destination port) – hodnota 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
  - **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.
  - **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

<table>
<tr><th colspan="8">Bity 0-15</th><th colspan="8">16-31</th></tr>
<tr><td colspan="8">Zdrojový port</td><td colspan="8">Cílový port</td></tr>
<tr><td colspan="16">Pořadové číslo odesílaného bajtu</td></tr>
<tr><td colspan="16">Pořadové číslo potvrzovaného bajtu</td></tr>
<tr><td colspan="2">Délka záhlaví</td><td colspan="2">Rezerva</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="6">Délka okna</td></tr>
<tr><td colspan="8">Kontrolní součet</td><td colspan="8">Ukazatel naléhavých dat</td></tr>
<tr><td colspan="16">Volitelné položky záhlaví</td></tr>
<tr><td colspan="16">Data aplikace</td></tr>
</table>

Obr. 8-7: Struktura TCP záhlaví

9.3.4 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



Obr. 8-8: Průběh spojení TCP a) navázání, b) ukončení

- Tomuto způsobu navázání obousměrného spojení se říká **three-way handshak**e (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.

9.3.5 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.

9.3.6 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 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.

10 Aplikační vrstva přenosových systémů - DHCP protokol, DNS systém, přenos souborů, webové protokoly, elektronická pošta

10.1 Dynamic Host Configuration Protocol (DHCP)

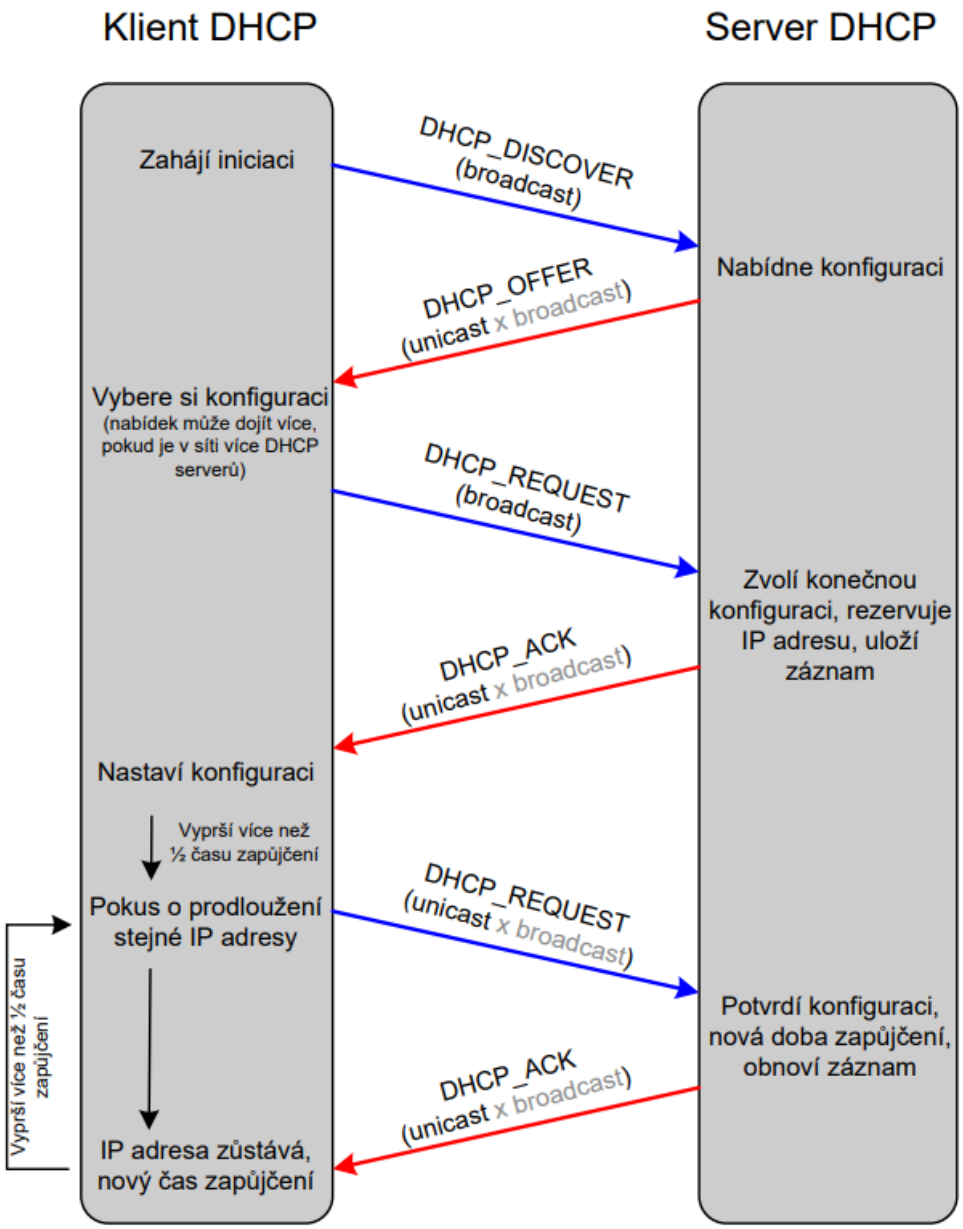
10.1.1 Základní vlastnosti DHCP

- 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. Základní 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.

10.1.2 Princip činnosti DHCP

- 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.
- 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.

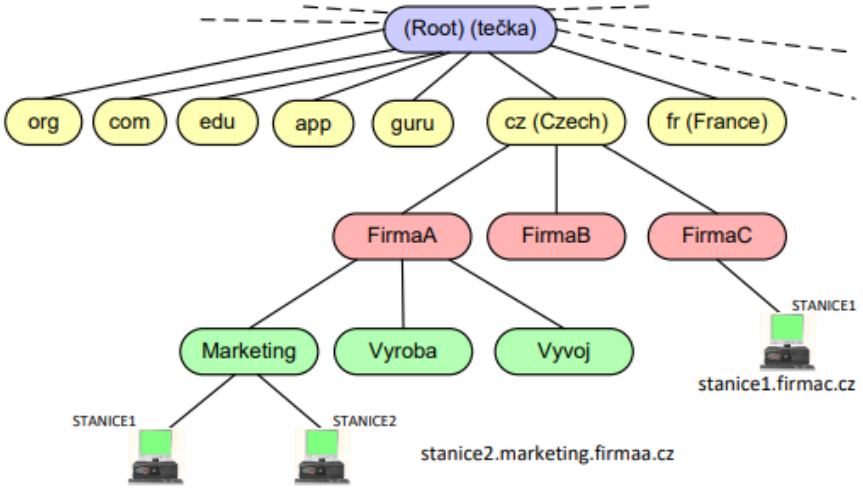
10.2 Domain Name System (DNS)

10.2.1 Motivace existence jmenného systému

- 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.

10.2.2 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. 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



Obr. 11-3: Hierarchie DNS systému

10.2.3 Domény a doménová jména

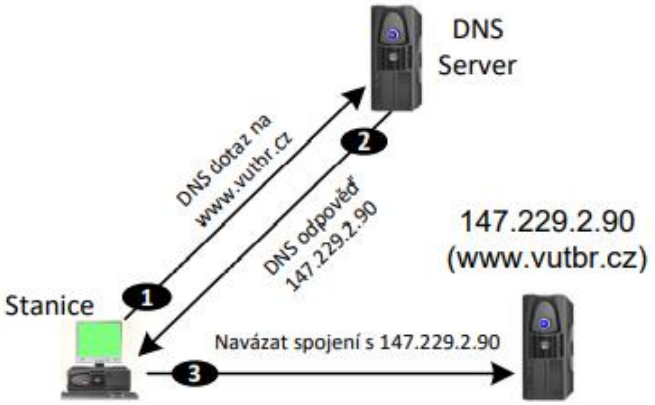
- Počítače jsou organizovány v **hierarchii domén**. Doménou je skupina počítačů, které jsou v nějakém vztahu vůči sobě (buď tvoří nějakou organizační jednotku, nebo jsou geograficky blízko sebe). Např.

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 povoleny **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.

10.2.4 Základní princip komunikace v systému DNS

- 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

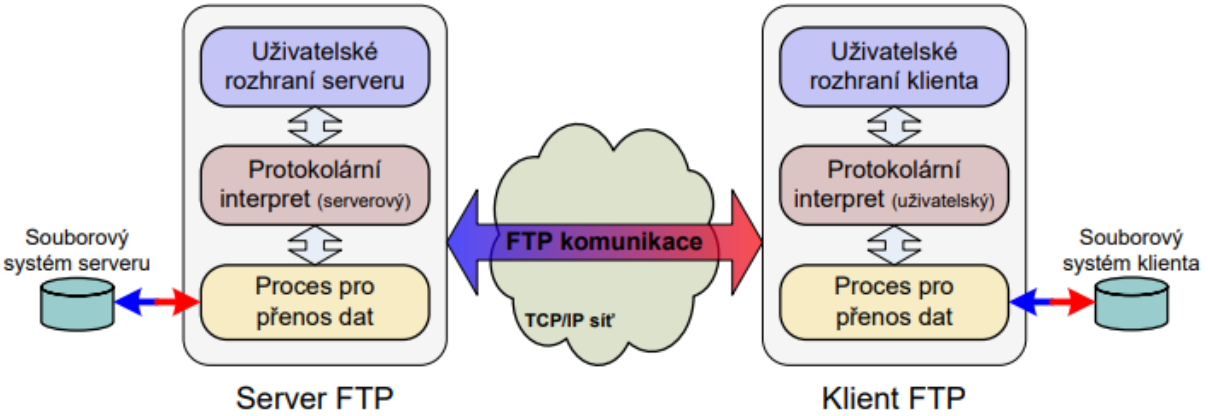


Obr. 11-4: Komunikace s využitím DNS serveru (zjednodušeno)

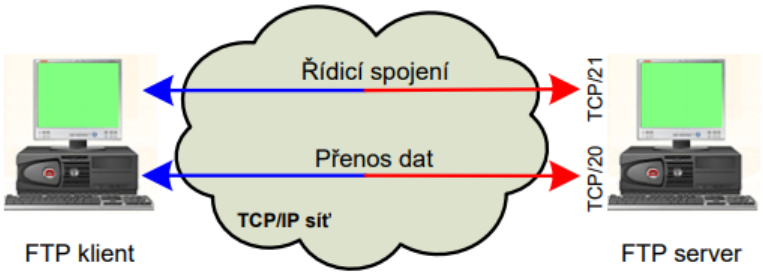
10.3 Přenos souborů a protokol FTP

10.3.1 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.



- 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). 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
- **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.

10.3.2 Základy komunikace klienta se serverem

- Klient posílá po řídicím spojení FTP příkazy (FTP commands), k interpretaci příkazů se využívá NVT protokol
- 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

10.3.3 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.

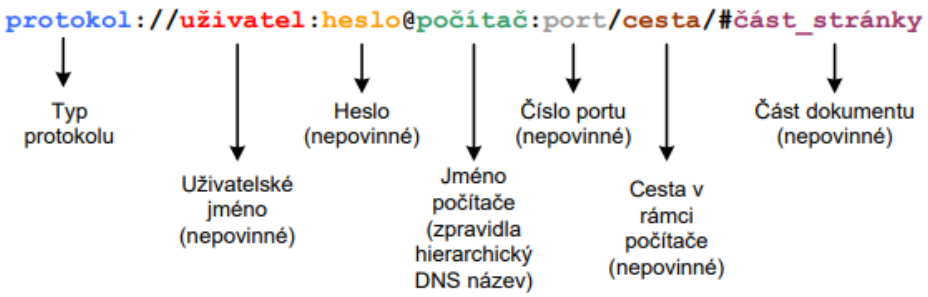
10.4 WWW a protokol http

10.4.1 Technologie kolem WWW

- Zpočátku si World-Wide Web 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

10.4.2 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. Některé položky spolu souvisí, např. heslo není nikdy přítomno bez uživatelského jména. Adresa počítače vzniká hierarchicky podle umístění počítače v doméně. 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)

10.4.3 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 **TCP/80.** V současné době se používá verze HTTP 1.1. Spolu s elektronickou poštou je 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)

- Standard HTTP obsahuje definici **číselných výsledkových kódů**, které tvoří odezvu na podnět klienta od serveru, podobně jako v případě FTP protokolu.

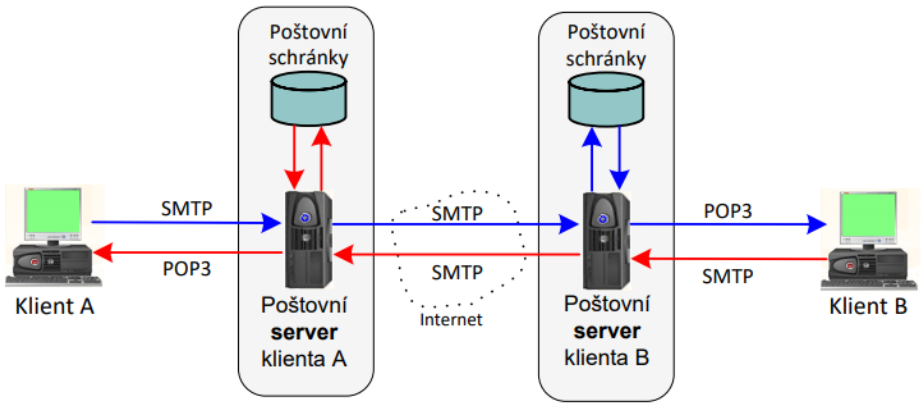
10.4.4 Činnost protokolu http

- 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** 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. To je velmi nepraktické, jelikož navazování a ukončování spojení pak výrazně zpomaluje komunikaci. Od verze 1.1 je spojení perzistentní, což znamená, že během jednoho spojení je možné přenést více objektů, spojení je pak zpravidla po určité době nečinnosti ukončeno (ze strany serveru)

10.5 Elektronická pošta a protokol SMTP

10.5.1 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.



- Klient A 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 @ 41) 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.

10.5.2 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.
    - **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.

10.5.3 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. 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.