

# **Počítačové sítě**

texty pro distanční studium

Doc. Ing. Cyril Klimeš, CSc.

Ostravská univerzita v Ostravě, Pedagogická fakulta  
Katedra informatiky a počítačů

## OBSAH

<b>1</b>	<b>ÚVOD DO POČÍTAČOVÝCH SÍTÍ .....</b>	<b>5</b>
1.1	DŮVODY, HISTORIE A FILOZOFIE VZNIKU POČÍTAČOVÝCH SÍTÍ.....	5
1.1.1	<i>Dávkové zpracování</i> .....	5
1.1.2	<i>Host/terminál</i> .....	6
1.1.3	<i>Terminálové sítě</i> .....	7
1.1.4	<i>Éra izolovaných počítačů</i> .....	9
1.1.5	<i>Model file server/pracovní stanice</i> .....	10
1.1.6	<i>Model klient/server</i> .....	11
1.1.7	<i>Výpočetní modely - shrnutí</i> .....	13
1.2	ČLENĚNÍ POČÍTAČOVÝCH SÍTÍ .....	13
<b>2</b>	<b>FUNKCE SÍTÍ PC-LAN .....</b>	<b>22</b>
2.1	SDÍLENÍ TECHNICKÝCH ZAŘÍZENÍ SÍTĚ .....	22
2.2	SDÍLENÍ SPOLEČNÝCH DAT .....	22
2.3	ELEKTRONICKÁ POŠTA .....	23
2.4	MONITOROVÁNÍ JINÝCH ÚČASTNÍKŮ SÍTĚ .....	23
2.5	KOMUNIKACE V SÍTI .....	23
<b>3</b>	<b>STRUKTURA SÍTÍ PC-LAN .....</b>	<b>25</b>
3.1	SÍŤOVÝ SERVER .....	25
3.2	WORKSTATION (PRACOVNÍ STANICE).....	25
3.3	KOMUNIKACE NA SÍTI .....	25
3.4	KONCEPCE PEER TO PEER .....	26
3.5	ZABEZPEČENÍ SÍTĚ .....	27
<b>4</b>	<b>TOPOLOGIE, PŘÍSTUPOVÉ METODY, ROZDĚLENÍ SÍTÍ PC-LAN.....</b>	<b>28</b>
4.1	TOPOLOGIE SÍTÍ PC-LAN .....	28
4.1.1	<i>SBĚRNICOVÁ TOPOLOGIE</i> .....	28
4.1.2	<i>KRUHOVÁ TOPOLOGIE</i> .....	28
4.1.3	<i>HVĚZDICOVÁ (STROMOVÁ) TOPOLOGIE</i> .....	29
4.2	PŘÍSTUPOVÉ METODY .....	29
4.2.1	<i>CSMA/CD (Carrier sense multiple access/collision detection)</i> ..	29
4.2.2	<i>TOKEN PASSING</i> .....	30
<b>5</b>	<b>SÍŤOVÝ MODEL A SÍŤOVÁ ARCHITEKTURA .....</b>	<b>31</b>
5.1	FYZICKÁ VRSTVA.....	36
5.2	LINKOVÁ (SPOJOVÁ) VRSTVA.....	37
5.3	SÍŤOVÁ VRSTVA.....	37
5.4	TRANSPORTNÍ VRSTVA .....	38
5.5	RELAČNÍ VRSTVA.....	38
5.6	PREZENTAČNÍ VRSTVA.....	39
5.7	APLIKAČNÍ VRSTVA .....	40
<b>6</b>	<b>SÍŤOVÉ PROTOKOLY .....</b>	<b>41</b>
6.1	PROTOKOLY NETBIOS A NETBEUI .....	41
6.2	PROTOKOL IPX/SPX .....	42
6.3	PROTOKOL TCP/IP .....	43
6.4	SROVNÁNÍ TCP/IP A REFERENČNÍHO MODELU ISO/OSI.....	47

<b>KONTROLNÍ OTÁZKA.....</b>	<b>48</b>
<b>7 TECHNICKÉ A PROGRAMOVÉ VYBAVENÍ SÍTÍ PC-LAN .....</b>	<b>49</b>
7.1 TECHNOLOGIE POČÍTAČOVÝCH SÍTÍ .....	49
7.1.1 Technologie sítě Ethernet (IEEE 802.3) .....	50
7.1.2 Technologie 10BASE 5.....	50
7.1.3 Technologie 10BASE2.....	51
7.1.4 Technologie 10BASE-T .....	52
7.1.5 Technologie 100BASE-TX.....	55
7.2 TECHNICKÉ VYBAVENÍ SÍTÍ PC-LAN .....	57
7.2.1 PŘENOSOVÉ MÉDIUM.....	57
7.2.2 SÍŤOVÁ KARTA - ADAPTÉR.....	58
7.3 PROGRAMOVÉ VYBAVENÍ SÍTÍ PC-LAN .....	59
7.3.1 DRIVER ADAPTÉRU.....	59
7.3.2 SÍŤOVÝ OPERAČNÍ SYSTÉM.....	59
7.4 PROPOJENÍ SÍTÍ PC - LAN S OSTATNÍMI SÍTĚMI .....	59
7.4.1 Opakovač (Repeater).....	60
7.4.2 Přepínač (Switch), most (Bridge).....	61
7.4.3 Směrovač (Router).....	61
7.4.4 Brána (Gateway).....	62
7.5 INTERNETWORKING .....	63
<b>8 SÍŤOVÝ OPERAČNÍ SYSTÉM WINDOWS NT.....</b>	<b>72</b>
8.1 REFERENČNÍ MODEL OSI VE WINDOWS NT .....	72
8.2 ZABUDOVANÉ SÍŤOVÉ VLASTNOSTI WINDOWS NT.....	73
<b>9 VLASTNÍ REALIZACE POČÍTAČOVÉ SÍTĚ .....</b>	<b>75</b>
9.1 ROZVAHA PRO VYTVOŘENÍ POČÍTAČOVÉ SÍTĚ .....	75
9.2 PARAMETRY PŘI MĚŘENÍ KABELŮ S UKÁZKOU PROTOKOLU .....	75
<b>10 RODINA PROTOKOLŮ TCP/IP, ARCHITEKTURA TCP/IP ....</b>	<b>79</b>
10.1 VZNIK RODINY PROTOKOLŮ TCP/IP.....	79
10.2 ARCHITEKTURA TCP/IP .....	79
10.3 ROZDĚLENÍ TCP/IP DO VRSTEV .....	84
10.4 DALŠÍ ASPEKTY TCP/IP: STANDARDIZACE, DOHLED NAD FUNGOVÁNÍM INTERNETU .....	86
<b>11 IP ADRESY .....</b>	<b>90</b>
11.1 STRUKTURA IP ADRES, TŘÍDY IP ADRES, DISTRIBUCE ADRES .....	90
11.2 OMEZENÝ ROZSAH ADRESNÍHO PROSTORU .....	92
<b>12 IP PROTOKOL, VLASTNOSTI SÍŤOVÉ VRSTVY .....</b>	<b>95</b>
12.1 IP PROTOKOL.....	95
12.2 PROTOKOL ICMP .....	96
12.3 ROZPOZNÁVÁNÍ ADRES .....	98
12.4 SMĚROVÁNÍ.....	98
<b>13 PROTOKOLY TRANSPORTNÍ VRSTVY .....</b>	<b>103</b>
13.1 FUNKCE TRANSPORTNÍ VRSTVY.....	103
13.2 TYPY SLUŽEB TRANSPORTNÍ VRSTVY .....	104
13.3 PROTOKOLY TRANSPORTNÍ VRSTVY .....	104

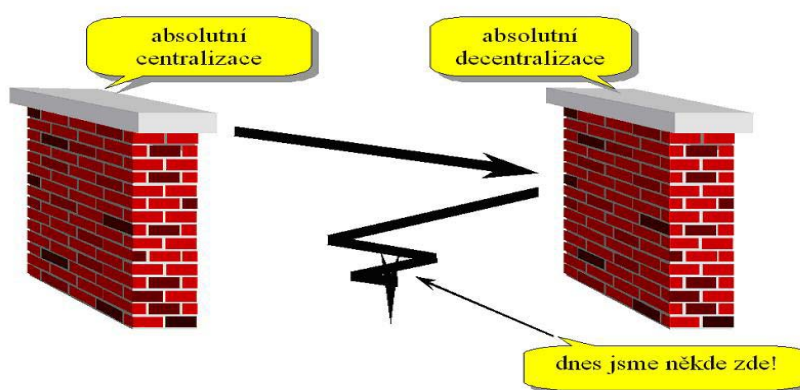
<b>14</b>	<b>SLUŽBY APLIKAČNÍ VRSTVY .....</b>	<b>108</b>
14.1	VLASTNOSTI APLIKACÍ V TCP/IP .....	108
14.2	TELNET .....	108
14.3	PROTOKOLY FTP A NFS .....	109
14.4	ELEKTRONICKÁ POŠTA .....	113
14.5	WORLD WIDE WEB .....	117
14.6	SLUŽBA DNS .....	119
<b>15</b>	<b>MODERNÍ PŘENOSOVÉ TECHNOLOGIE .....</b>	<b>121</b>
15.1	ATM .....	121
15.2	FDDI .....	122
15.3	GIGABITOVÝ ETHERNET .....	123



# 1 Úvod do počítačových sítí

## 1.1 Důvody, historie a filozofie vzniku počítačových sítí

Pochopení vzniku počítačových sítí vychází z objasnění tzv. výpočetního modelu. Výpočetní model je ucelená představa o tom, kde jsou aplikace uchovávány jako programy a kde skutečně běží, zda (a jak) jsou aplikace rozděleny na části, jak tyto části vzájemně spolupracují, kde a jak se uchovávají a zpracovávají data, kde se nachází uživatel, kdy, jak a jakým způsobem komunikuje se svými aplikacemi apod. Výpočetní model se vyvíjel a stále vyvíjí. Některé výpočetní modely nepočítají s existencí sítě (např. dávkové zpracování), jiné výpočetní modely s existencí sítě spíše počítají, ale bezpodmínečně ji nevyžadují (např. klient/server), další modely vyžadují existenci sítě (např. distribuované zpracování, network-centric computing). Proto správné pochopení výpočetních modelů je důležité i pro zvládnutí problematiky sítí. Výpočetní model se vyvíjel od absolutní decentralizace zpracování informací až po absolutní decentralizaci a postupně si ve většině aplikací hledá pozici určitého kompromisu (obr. 1.1). Je třeba na tomto místě zdůraznit, že výpočetní model je vlastností konkrétní aplikace, nikoli nutně počítačové sítě jako celku, proto je možné (a v moderních počítačových sítích běžné, že se souběžně využívá v různých aplikacích různých výpočetních modelů).

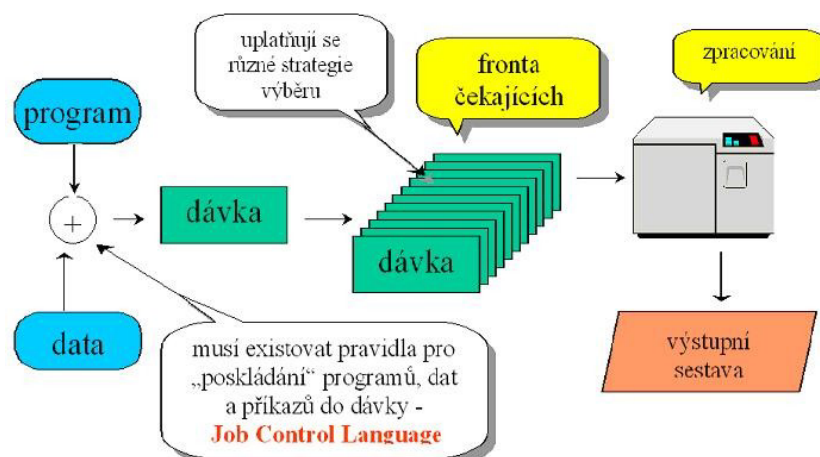


Obr. 1.1 Vývoj výpočetního modelu

### 1.1.1 Dávkové zpracování

Historicky nejstarší výpočetní model je dávkové zpracování (batch processing). Byl vynucen dobou, (ne)dokonalostí technologické základny, malými schopnostmi SW i HW (nebyla systémová podpora multitaskingu), vysokými náklady, potřebou „kolektivního“ využití dostupné výpočetní techniky. Dnes ještě není mrtvý, i když jeho použití se omezuje na speciální případy. Podstata dávkového zpracování je naznačena na obr. 1.2.

## Počítačové síť



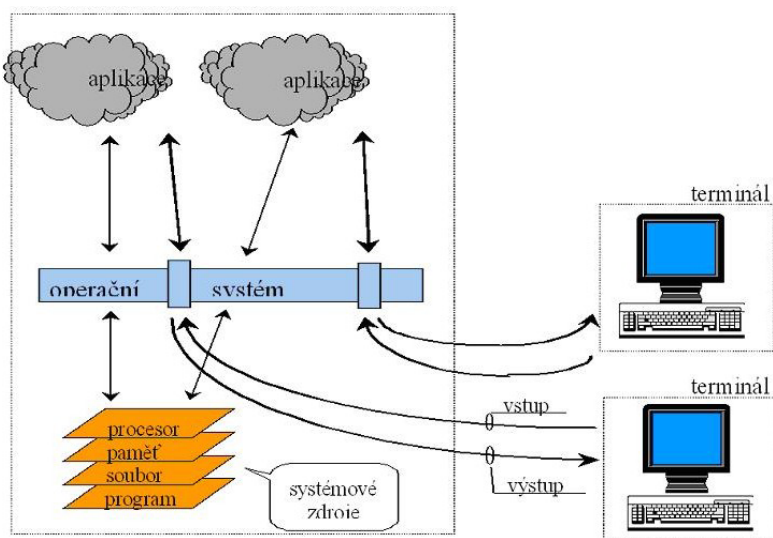
Obr. 1.2 Podstata dávkového zpracování

Dávkového zpracování má řadu nevýhod. Uživatel nemá bezprostřední kontakt se svou úlohou, chybí interaktivita, uživatel nemůže reagovat na průběh výpočtu (volit varianty dalšího průběhu, opravovat chyby, ...), doba obrátky (od odevzdání vstupní dávky do získání výstupní sestavy) bývá relativně dlouhá. Dávkového zpracování má však řadu výhod. Dokáže (relativně) dobře vytižít dostupné zdroje, vychází vsťříc intenzivním výpočtům (hodně „počítavým“ úlohám, s minimem V/V), nutí programátory programovat „hlavou“ a ne „rukama“ (protože při dlouhé obrátce si nemohou dovolit experimentovat). Dávkového zpracování se používalo v prostředí sítě tzv. vzdálené zpracování úloh (Remote Job Execution, Remote Job Entry). Uživatel na jednom uzlu připravil dávku a poslal ji ke zpracování na jiný uzel (!! uživatel sám určoval, kam dávku pošle!!!). V současnosti se využívá modernější alternativa RJE („distribuovaná aplikační platforma“??). Tu lze zjednodušeně charakterizovat tak, že síť si sama určuje, kam pošle dávku ke zpracování.

### 1.1.2 Host/terminál

Dalším výpočetním modelem je host/terminál. Vznikl jako reakce na neinteraktivnost dávkového zpracování. Dokáže uživatelům zajistit přímý kontakt s jejich úlohami a interaktivní způsob práce, dokáže „obsloužit“ více uživatelů současně. Realizace byla umožněna zdokonalením SW a HW, především vznikem SW mechanismů pro sdílení času (time sharing) a uživatelských pracovišť (terminálů). Pod pojmem host rozumíme počítač, který je „hostitelem“ systémových zdrojů, procesoru, paměti, V/V zařízení, programů, dat, systémových utilit apod. Výpočetní model host/terminál je zřejmý na obr.1.3

## Počítačové sítě



Obr.1.3 Model host/terminál

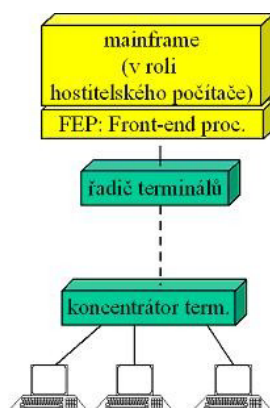
Podstatou modelu host/terminál je, že vše je „na jedné hromadě“, programy (úlohy) běží na hostitelském počítači a data se zpracovávají v místě kde se nachází (nedochází k přenosům velkých objemů dat). Mezi hostitelským počítačem a terminály se přenáší pouze výstupy na obrazovku uživatele a vstupy z uživatelské klávesnice. Terminály mohou být umístěny v různé vzdálenosti tj. buď blízko (místní, lokální terminály) a nebo daleko (vzdálené terminály) či kdekoli v síti. Model host/terminál je způsob fungování, kdy hostitelský počítač je v roli, ve které nějaký konkrétní počítač vystupuje jako střediskový počítač, mainframe. Přitom mainframe může fungovat dávkově (používá dávkové zpracování) a nebo v režimu sdílení času. Jako hostitelský počítač může fungovat např. PC s Unixem (rozhodující je charakter OS, nikoliv HW).

Model host/terminál má řadu výhod. Má centralizovaný charakter, správu stačí zajišťovat na jednom místě, snazší sdílení dat, programů. Relativně snadná implementace neklade příliš velké nároky na aplikace a neklade velké nároky ani na přenos dat mezi hostitelským počítačem a terminály protože se přenáší pouze výstupy na obrazovku uživatele a vstupy z uživatelské klávesnice (!!jsou to malé objemy dat, protože se (typicky) pracuje ve znakovém režimu!!). Nevýhody modelu host/terminál jsou v první řadě v iluzi uživatele, že má hostitelský počítač výhradně ke své dispozici, ale ve skutečnosti má k dispozici jen n-tou část jeho výkonnosti! Uživatelský komfort je relativně nízký vzhledem ke znakovému režimu (!!! není to vina výpočetního modelu, ale způsobu jeho využití!!!).

### 1.1.3 Terminálové sítě

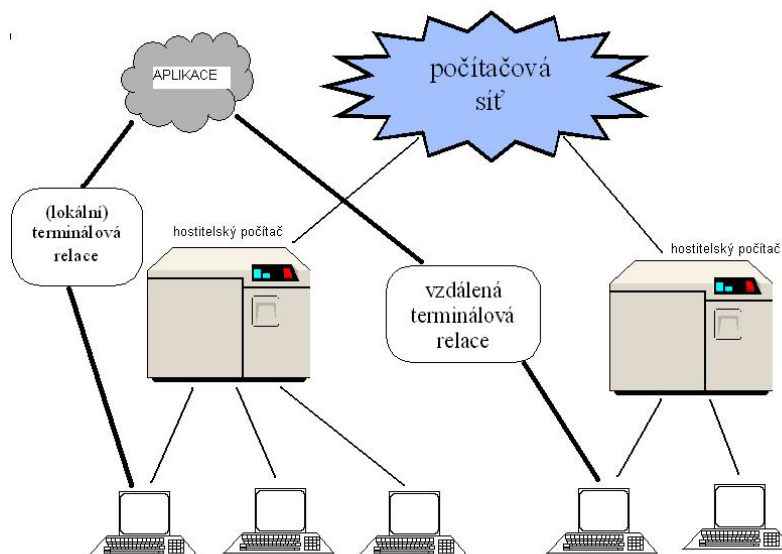
V době největšího rozmachu střediskových počítačů se budovaly celé rozsáhlé terminálové sítě využívající specializované prvky (řadičů, koncentrátorů, FEP, ...). Terminálová síť je pouze (rozsáhlý) terminálový rozvod, nikoli skutečná počítačová síť. Filosofii terminálové sítě převzala síťová architektura SNA firmy IBM

## Počítačové sítě



Obr. 1.4 Princip terminálové sítě

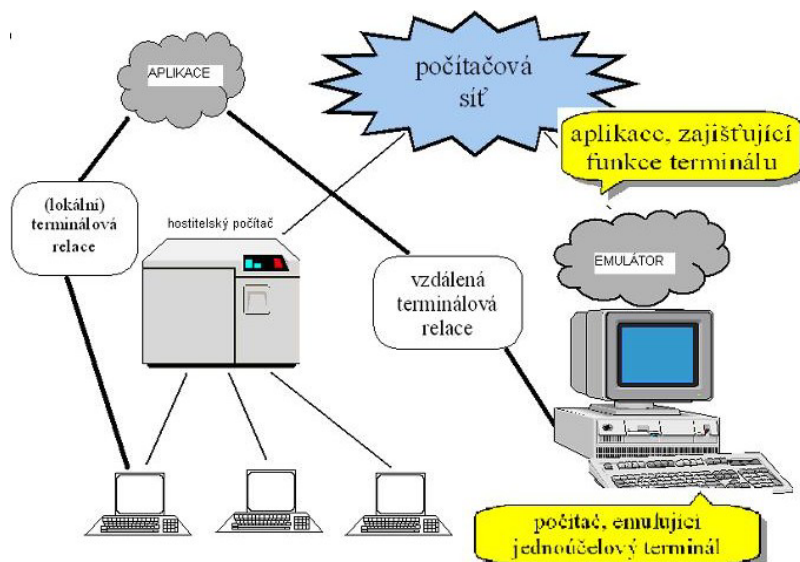
Terminálová relace je vztah vznikající mezi terminálem a aplikací běžící na hostitelském počítači. Rozeznáváme tak lokální a vzdálené terminálové relace. Při lokální (místní) terminálové relaci jsou data přenášena jen po terminálové síti daného hostitelského počítače, naproti tomu při vzdálené terminálové relaci jsou data přenášena po skutečné počítačové síti. Představa vzdálené terminálové relace je naznačena na obr. 1.5.



Obr. 1.5 Vzdálená terminálová relace

Původní podstatou vzdálené terminálové relace byla skutečnost, že terminál jednoho hostitelského počítače se dostával do postavení terminálu jiného hostitelského počítače, tzn. že šlo o vzdálený terminál, kdy cílem bylo využívat zdroje (aplikace, data, ...) vzdáleného počítače. V současnosti se do postavení vzdáleného terminálu dostává jednouzivatelský počítač (pracovní stanice), který se pouze chová jako skutečný terminál a jedná se o tzv. terminálovou emulaci. Terminálová emulace je naznačena na obr. 1.6.

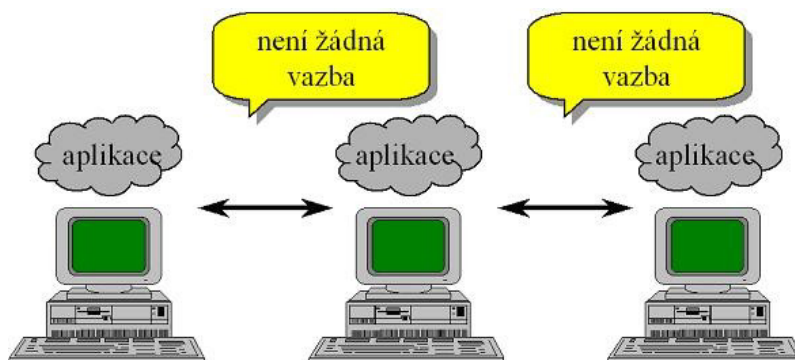
## Počítačové sítě



Obr. 1.6 Terminálová emulace

### 1.1.4 Éra izolovaných počítačů

Postupně se výpočetní technika stávala čím dál tím lacinější, zrodily se minipočítače, ale výpočetní model se nezměnil. Stále bylo nutné (z ekonomických důvodů), aby více uživatelů sdílelo jeden počítač. Zlom nastal až s příchodem osobních počítačů, kdy už bylo ekonomicky únosné přidělit každému uživateli jeho vlastní počítač, k výhradnímu použití. Vzniká nový výpočetní model, model izolovaných osobních počítačů.



Obr. 1.7 Izolované počítače

Od příchodu osobních počítačů si lidé slibovali především vyšší komfort, větší pružnost a flexibilitu, nezávislost na ostatních (žádnou potřebu sdílení). Tyto požadavky se v zásadě podařilo splnit, ale objevily se jiné problémy. Základním problémem izolovaných počítačů je, že dříve se každý problém řešil jednou, na jednom místě, nyní se každý problém řeší n-krát na n-místech. Uživatelé jsou mnohem více odkázáni na sebe, jsou problémy se sdílením dat a programů.



### 1.1.5 Model file server/pracovní stanice

S vyšším využíváním osobních počítačů se musel začít řešit kompromis mezi přísnou centralizací danou modelem host/terminál a izolovanými osobními počítači. V životě většinou vítězí rozumný kompromis a proto i zde se postupně našel kompromis, kdy něco se dá každému do výhradního vlastnictví a něco se naopak bude sdílet. Každý uživatel může mít vlastní výpočetní kapacitu, která je již relativně laciná a tudíž lze vytvořit uživateli příjemné pracovní prostředí tj. vlastní pracovní místo (klávesnici, monitor, myš, apod.) a některé programy a data. Naopak je vhodné sdílet drahé periferie např. laserové tiskárny, plottery, scannery, společná data, firemní databáze, sdílené dokumenty, event. některá „soukromá“ data, např. kvůli zálohování. Předpokladem úspěšného sdílení je, že uživatel nesmí sdílení poznat a nesmí pozorovat významnější rozdíl v rychlostech přístupu ke sdíleným a privátním objektům, přitom je vhodné, když si uživatel vůbec nemusí uvědomovat fakt sdílení. Proto jsou nutné dostatečně rychlé přenosové technologie a mechanismy sdílení musí být implementovány transparentně. S těmito požadavky vznikají první sítě LAN řešící především potřebu sdílení souborů (programů, dat) a periférií (tiskáren, ....). Jsou řešeny tak, aby je „nebylo vidět“ a aby na nich mohly pracovat aplikace, které nejsou uzpůsobeny síťovému prostředí (neuvědomují si existenci sítě). Teprve později se sítě mohou stát „viditelné“ když se objevují aplikace, které přímo počítají s existencí sítě. Další motivací pro vznik sítí (především WAN) je překlenutí vzdáleností pro potřeby komunikace, sdílení výpočetní kapacity a sdílení dat. Vznikají první rozlehlé sítě WAN na kterých kvůli omezeným přenosovým možnostem (pomalým přenosům) nelze dosáhnout transparentního sdílení. Proto případné sdílení je řešeno netransparentně a tím si uživatelé si uvědomují rozdíl mezi „místním“ a „vzdáleným“ připojením. První výpočetní model, který využívá existence sítě, se nazývá file server / pracovní stanice viz obr.1. 8



Obr.1.8 Model file server/pracovní stanice

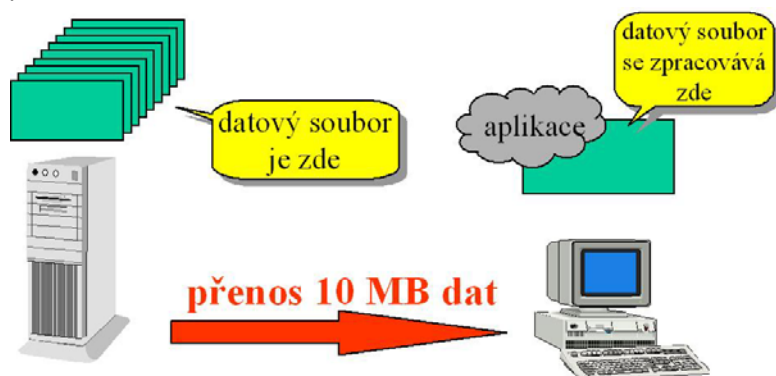
Model file server/pracovní stanice je pro aplikace „neviditelný“ a zajišťuje plně transparentní sdílení. Tím je použitelný pro aplikace, které si neuvědomují existenci sítě a pro aplikace určené původně pro prostředí izolovaných počítačů, které umožňuje sdílení dat i programů a umožňuje centrální správu.

## Počítačové sítě

Model file server / pracovní stanice lze snadno implementovat v případě, že tomu operační systém vychází vstříc tj. když operační systém dokáže rozlišit požadavek na místní a vzdálený soubor. Model file server / pracovní stanice lze implementovat i v případě, že tomu operační systém nevychází vstříc. Pak operační systém se překryje vrstvou, která zajistí přesměrování požadavku.

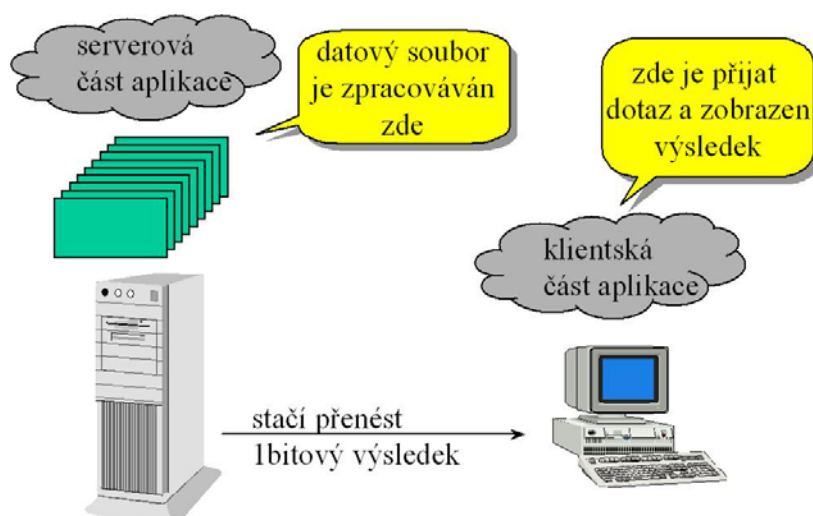
### 1.1.6 Model klient/server

Nevýhodou modelu file server/pracovní stanice je, že v některých situacích je hodně neefektivní, způsobuje zbytečný přenos a může snadno dojít k zahlcení sítě. Důvodem jsou data, která jsou zpracovávána jinde, než jsou umístěna (a proto musí být přenášena) a nebo programy, které musí přenést obrovské množství dat pro svoji funkci. Příkladem je práce s databází, kdy je potřebné prohledat databázový soubor velikosti 10 MB ve kterém se nachází položka XY.



Obr. 1.9 Řešení požadavku pracovní stanice na file server

Řešením je model klient/server.



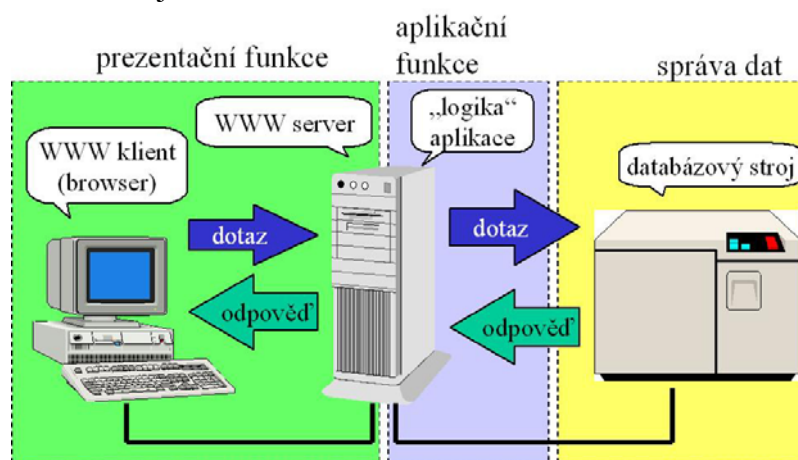
Obr. 1.10 Řešení požadavku v režimu klient/server

Základní myšlenkou je ponechat zpracování dat tam, kde se data nachází a naopak výstupy pro uživatele generovat tam, kde se nachází uživatel. Musí

## Počítačové sítě

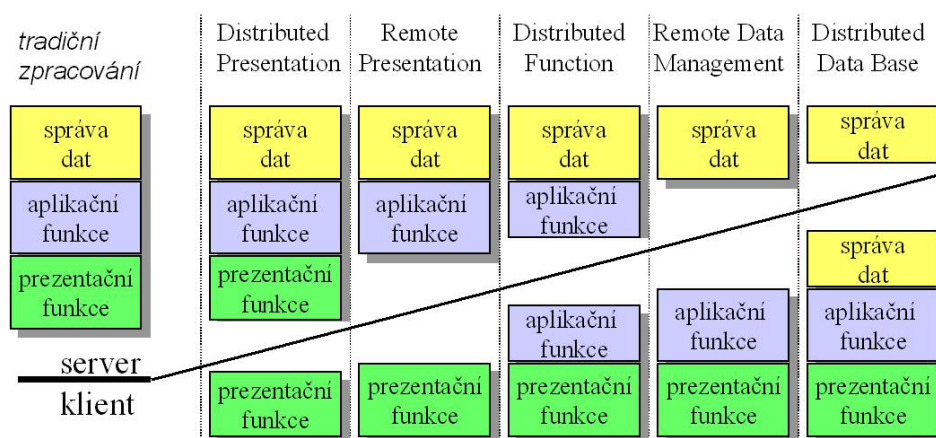
pak dojit k rozdělení původně monolitické aplikace na dvě části tj. na serverovou část, zajišťující zpracování dat a klientskou část, zajišťující uživatelské rozhraní.

Základní vlastnosti modelu klient/server je, že klient a server si posílají data představující dotazy a odpovědi. Pokud se klient a server dobře dohodnou, mohou účinně minimalizovat objem přenášených dat, tím mají výrazně menší přenosové nároky a mohou pracovat i v prostředí rozlehlých sítí. Klient a server mohou stát na různých platformách. Klasické řešení klient/server rozděluje aplikaci na dvě části. Vzniká dvouvrstvá architektura která je v současnosti nahrazována třívrstvou architekturou. Funkce jsou rozděleny do 3 částí - prezentační funkce (uživatelské rozhraní, sběr dotazů, prezentace výsledků), aplikační funkce (vlastní logika aplikace) a správa dat (vlastní databázové operace). Představa 3-úrovňové architektury klient/server, s využitím WWW je naznačena na obr. 1.11.



Obr. 1.11 Tříúrovňová architektura klient/server

Možné varianty rozdělení 3 úrovní modelu do dvou částí jsou uvedeny na obr. 1.12.



Obr. 1.12 Varianty rozdělení tříúrovňové architektury

Dalším vývojem výpočetního modelu lze očekávat model agent/manažer, jako speciální, pro aplikace z oblasti správy sítí, dále network-centric computing jako výpočetní model zavedený v souvislosti s jazykem Java a technologií



## Počítačové sítě

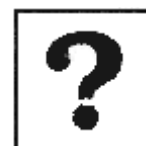
ActiveX , plně distribuovaný model ve kterém jsou části aplikace „roztroušeny“ po síti konečně komponentní model ve kterém aplikace nejsou monolitní, ale skládají se z částí roztroušených v síti.

### 1.1.7 Výpočetní modely - shrnutí

Výpočetní modely se neustále vyvíjejí spolu s vyvíjejícím se výpočetním prostředím. Vzhledem ke zvyšujícímu se počtu rutinně používaných aplikací v podnicích a institucích se poněkud zvyšuje průměrná doba jejich celkové obměny. Díky tomu dochází k dlouhodobějšímu přetrvání starších a koncepčně překonaných výpočetních modelů, jako např. terminálové zpracování. Lze proto očekávat, že vedle nových aplikací využívajících moderní výpočetní modely se budeme ještě dlouhou dobu setkávat i se staršími výpočetními modely, a to stále častěji i spolu v jednom výpočetním prostředí.

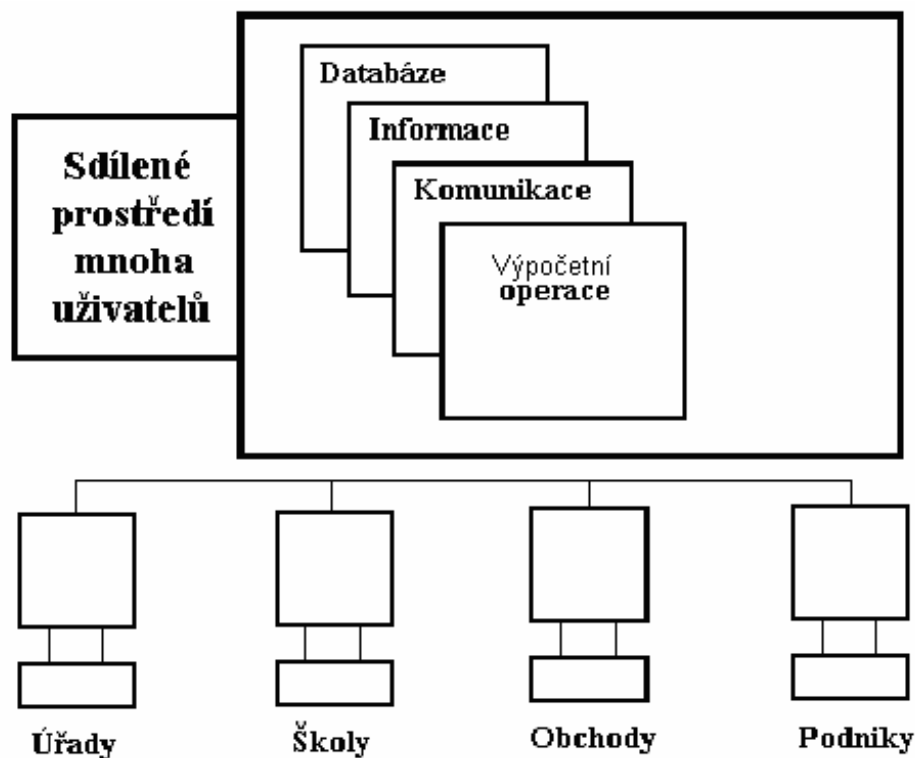
#### Kontrolní otázky:

1. Co je výpočetní model?
2. Čím je důležitý pro problematiku počítačových sítí?
3. Je možné používat více výpočetních modelů zároveň?
4. Co považujete za nejvýznamnější nedostatky pro plně distribuovaný výpočetní model (izolované “osobní” počítače)?



## 1.2 Členění počítačových sítí

Jak jsme si uvedli v předchozí kapitole, potřeba řešení lokálních úloh bez použití ústředního počítače vedla ke vzniku tzv. *počítačových sítí*, které umožňují uživatelům pracovat v síti i mimo ní. Rozvoj počítačových sítí umocnilo též prudké nasazení počítačů řady PC v komerční sféře, co se projevilo prudkým rozvojem sítí LAN v této oblasti (úřady, školy, obchody, podniky). Síťové prostředí přibližuje obr. 1.13.



Obr. 1.13 Sdílené síťové prostředí pro více uživatelů

Počítačové sítě LAN patří do oblasti datových sítí, které rozdělujeme podle územního rozložení do třech skupin :

1. **WAN** (wide area network) - veřejné datové sítě
2. **MAN** (metropolitan area network) - městské datové sítě
3. **LAN** (local area network) - lokální datové sítě

**WAN** - veřejné datové sítě jsou svým rozsahem neomezené, přičemž zabírají území států i kontinentů.

**MAN** - městské sítě zabírají území města tedy řádově km. Vznikají spojením více různých vzdálených sítí LAN.

**LAN** - lokální sítě pokrývají území nepřesahující 1 - 2 km. Tedy svým nasazením pokrývají rozsah pracovišť, budov, závodů.

Abychom mohli jednoznačně začlenit počítačovou síť je nutné znát členění podle nejrůznějších kritérií. Příkladem kritéria je dosah sítě, architektura sítě, role (postavení) uzlů sítě účel, kterému síť slouží, vlastnické vztahy k síti, použité přenosové techniky, použité přenosové technologie, použitá přenosová média, mobilita uživatelů apod. Všechna tato kritéria nemusí být exaktně definována, ani výsledné kategorie („škatulky“) nemusí být přesně vymezeny, hranice mezi nimi nemusí být ostré a konkrétní klasifikace může mít i subjektivní složku neboť kritéria nejsou vzájemně disjunktní. Výsledné „škatulky“, představující dělení podle různých kritérií, se mohou vzájemně prolínat, jedna a tatáž síť může patřit do různých „škatulek“ současně (při uvážení různých kritérií). Pokusme se nyní objasnit některá kritéria.

Základním kritériem je dosah sítě – viz tab.1.1.

## Počítačové sítě

VZDÁLENOST	UMÍSTĚNÍ	DRUH SÍTĚ
10 m	místnost	LAN
100 m	budova	lokální síť
1 km	prostor úřadu, továrny	městské síť
10 km	město	MAN
100 km	stát	městské síť
1000 km	kontinent	WAN
10 000 km	planeta	rozlehlé síť Internet

Z této tabulky vyplývá základní srovnání LAN a WAN – viz. Tab. 1.2

	LAN	WAN
kvůli čemu se zřizují	spíše pro potřeby sdílení	spíše pro potřeby komunikace
přenosová rychlost	vyšší (např. 10 až 100 Mbps)	nižší (např. 64 kbps)
topologie sítě	systematická (pravidelná)	nesystematická (nepravidelná)
vlastnictví přenosové infrastruktury	vlastní provozovatel	provozovatel si pronajímá
charakter uzlů	„menší“, převažují pracovní stanice	„větší“, převažují servery
dostupnost uzlů	„občas“ (podle potřeb uživatelů)	trvale
přenosové zpoždění	malé	velké
spolehlivost přenosových cest	vyšší	nižší

Hranice mezi LAN a WAN není ostrá a rozdíly se stále stírají. Síť LAN se zvětšují a síť WAN se zrychlují. Trend se vyznačuje neustálým zmenšováním rozdílu mezi oběma druhy sítí. V cílovém stavu bude uživateli jedno, zda pracuje v síti LAN či WAN, všude bude mít stejné možnosti a bude používat stejný styl práce a tím si nebude muset uvědomovat rozdíl mezi LAN a WAN.

Dnes existují přenosové technologie, které jsou buď vhodné jen pro LAN (např. Novell IPX/SPX), nebo vhodné jen pro WAN (např. X.25), ale také vhodné pro LAN i WAN (TCP/IP, ATM). Je tomu proto, že splňují vždy určité předpoklady např. krátké přenosové zpoždění (IPX/SPX), nebo nespolehlivost přenosových cest (X.25). Do budoucna, při splývání LAN a WAN, přežijí jen technologie schopné velkého přizpůsobení (TCP/IP, ATM). *Poznámka* - existují sítě MAN (Metropolitan Area Networks), které jsou pokusem „zabydlet“ předěl mezi sítěmi LAN a WAN. Někdy se označují jako síť s dosahem v rámci celého města, nebo síť sloužící potřebám města. Používají technologie je DQDB (Distributed Queue Dual Bus). V rámci IEEE byla založena skupina pro standardy sítí MAN (DQDB=IEEE 802.6). Dalšími kritérii specifikace sítí je dle celkové architektury, kde jde hlavně o celkovou koncepci síťového modelu (počet vrstev, roli vrstev), o protokoly jednotlivých vrstev (hlavně vyšších), o přístup k otázkám spolehlivosti, charakteru služeb, garanci kvality, apod. Vše tvoří tzv. síťovou architekturu

## Počítačové sítě

kde příkladem jsou sítě na bázi TCP/IP, sítě ISO/OSI, sítě SNA, sítě IPX/SPX (Novell) apod.

Jiné kritérium dělení sítí je dle postavení (role) uzlů a jde o to, zda uzel sítě pouze nabízí své vlastní zdroje k využití ostatním uzlům, formou sdílení (chová se jako server), nebo pouze využívá zdroje ostatních uzlů, prostřednictvím sdílení (chová se jako klienti) a nebo nabízí vlastní zdroje a současně využívají zdroje jiných uzlů (chová se současně jako klient i server). Pokud převažuje současně využívání i nabízení, jde o síť typu peer-to-peer, kde postavení uzlů je zde symetrické, uzly komunikují jako „rovný s rovným“. Pokud existuje ostrá hranice mezi nabízením a využíváním jde o síť serverového typu a postavení uzlů je asymetrické, některé uzly se chovají jako klienti, jiné jako servery. Srovnání je zřejmé z přiložené tab. 1.3.

	Síť serverového typu	Síť peer-to-peer
<b>postavení uzlů sítě</b>	asymetrické	symetrické
<b>umístění sdílených zdrojů</b>	na jednom místě (na centrálním serveru)	na více místech (u vlastníků)
<b>optimalizováno na</b>	rychlost a výkon	jednoduchost
<b>předpokládá se správce sítě</b>	ano	ne
<b>cena odvozena od počtu</b>	uživatelů	uzlů
<b>cena je inkrementální</b>	(typicky) ne	(typicky) ano

Uvedené dělení se týká hlavně lokálních sítí, přičemž se rozdíly zmenšují a obě kategorie postupně splývají.

Jiným kritériem dělení je účel sítě ke kterému slouží. Z tohoto pohledu existuje intranet a extranet.

**Intranet** je síť, sloužící potřebám fungování vlastní organizace (podniku, firmy, instituce, ...) nikoli prezentaci „navenek“ či zpřístupnění vlastních informací jiným subjektům, obchodování a dalším „externím“ aktivitám. Technicky se jedná o využití Internetových technologií (TCP/IP) „uvnitř“ podnikových sítí, využití Internetových služeb (hlavně WWW) pro interní informační systémy a sdílení informací. Pro uživatele to znamená, že mohou používat jednotný styl práce směrem „dovnitř“ i „navenek“ a mohou pracovat s jednotným uživatelským rozhraním.

**Extranet** je takové využití sítě, které sleduje „vnější“ cíle tj. prezentaci firmy, podniku, instituce atd. směrem navenek. Jedná se hlavně o obchodování (marketing a reklama, dojednávání a uzavírání obchodů, placení a dodávání zboží) a další aktivity zahrnující součinnost externích subjektů. Po technické

## Počítačové sítě

stránce je Extranet (typicky) založen na technologiích Internetu, využívá přenosových infrastruktur Internetu a jejich služeb.

Další kritérium dělení je vztah k vlastnictví sítě. Zde je třeba uvažovat o tom, kdo je vlastníkem sítě jako celku, kdo je faktickým provozovatelem sítě, kdo je uživatelem sítě, komu smí být služby sítě poskytovány a jaké služby jsou poskytovány. Z těchto hledisek existují sítě privátní, poloprivátní, veřejné, virtuální privátní a sítě VAN.

U **privátní počítačové sítě** je vlastníkem, provozovatelem i uživatelem tentýž subjekt i když některé části (např. přenosové trasy) mohou být pronajaty od jiných subjektů a ten, kdo síť vybudoval a uvedl do provozu, může být jiný subjekt.

Pro **veřejnou (datovou) síť** je vlastníkem i provozovatelem sítě určitý (stejný) subjekt,

který sám není uživatelem své sítě a vlastní uživatelé mohou být jiné subjekty. Služby sítě jsou poskytovány na komerčním principu, mohou být nabízeny zájemcům bez omezení (skutečně „veřejně“) a nabízené služby mají nejčastěji charakter pouhého přenosu dat.

U **(polo)privátní sítě** je vlastníkem i provozovatelem sítě určitý (stejný) subjekt, který sám (typicky) není uživatelem své sítě. Uživatelé mohou být jiné subjekty a služby sítě jsou jim poskytovány na komerčním principu a služby sítě jsou nabízeny jen určitému omezenému okruhu uživatelů, například jen vlastním zákazníkům. Důvodem pro poskytování služeb jen omezenému okruhu zájemců jsou obchodní strategie a záměr provozovatele a nemožnost získání licence na veřejné poskytování.

**Virtuální privátní síť** je samostatnou podsítí jiné sítě, typicky veřejné datové sítě. Technicky a provozně jde o součást „mateřské“ (veřejné) sítě a z pohledu uživatele jde o samostatnou síť (uživatel si může myslet, že síť je jen jeho a je mu plně k dispozici). Smyslem takového řešení je, že uživatel chce mít vlastní síť, ale nevyplatí se mu ji budovat a provozovat, neboť na to nemá lidi, znalosti, zázemí a je to pro něj takto výhodnější.

Dalším kritériem dělení sítí je dle použité přenosové techniky a to buď přenos s přepojováním okruhů, nebo s přepojováním paketů. Další členění je dle toho, jak velké „kusy“ dat jsou přenášeny najednou a to přepojování zpráv (hodně velké), přepojování paketů a rámců (velké) a přepojování buněk (malé). Konečně na jaké úrovni se realizuje přepojování. Buď na úrovni síťové vrstvy (paketů) a nebo přepojování na úrovni linkové vrstvy (rámců a buněk).

Základní způsoby přenosu lze porovnat se známými poštovními službami. Při variantě „telefon“ vzniká mezi příjemcem a odesílatelem (fyzicky) přímá, souvislá cesta a komunikace probíhá v reálném čase. Představa je taková, že od odesílatele vede až k příjemci jednodílná „trasa“, přenášená data se nikde nehromadí a data nemusí být příjemci explicitně adresována. Příjemce je jednoznačně určen a je to ten, kdo je na druhém konci „trasy“. Jedná se o tzv. **přepojování okruhů (circuit switching)**. Při variantě „listovní pošta“ mezi příjemcem a odesílatelem nevzniká žádná souvislá vyhrazená cesta a na cestě od příjemce k odesílateli existují přestupní body, které si zásilku postupně předávají, a jsou schopny ji nakonec dopravit až k příjemci. Přenášená data cestují podle principu „store & forward“ kdy

jednotlivé přestupní uzly nejprve přijmou celý přenášený blok dat, a teprve pak jej předají dál (není to v reálném čase). Přenášená data musí být explicitně adresována a musí nějak identifikovat svého příjemce. Jedná se o tzv.

**přepojování paketů (packet switching)**. Metoda přepojování okruhů pochází ze „světa spojů“ (funguje tak telefonní síť) a je výhodná pro „rovnoměrné“ přenosy např. pro multimediální formáty (živý zvuk a obraz). Používá se např. v sítích ISDN. Metoda přepojování paketů pochází ze „světa počítačů“ a je výhodná pro „nárazové“ přenosy, např. přenosy souborů a nevhodná pro zvuk a obraz. Takto fungují prakticky všechny sítě LAN i WAN.

Ve světě počítačů se používá přepojování zpráv, paketů, rámců, buněk. Proto si nyní provedme detailnější srovnání. Při **přepojování zpráv (message switching)** se přenáší hodně velké bloky dat najednou, přičemž velikost bloku není apriorně omezena což je problematické např. při přenosu bufferů. Proto se dnes již nepoužívá. Při **přepojování paketů (packet switching)** mohou být přenášené bloky různě velké a maximální velikost paketu je omezena. Musí být předem známo, jak veliký buffer musí stačit. **Přepojování rámců (frame relay)** je „odlehčené“ přepojování paketů (na úrovni linkové vrstvy), velikost rámce je proměnná, ale omezená.

Maximálně odlehčené přepojování (na linkové vrstvě) je **přepojování buněk (cell relay)**. Buňky jsou velmi malé a mají pevnou velikost.

Příklady:



X.25: „klasická“ implementace přepojování paketů, dnes zbytečně robustní a příliš těžkopádná

Frame Relay: „odlehčená“ X.25, nemá zabezpečovací mechanismy

Ethernet switching: zdokonalené přepojování rámců v prostředí Ethernetu

ATM: „klasická“ implementace přepojování buněk, pracuje s buňkami velikosti 48 + 5 byte

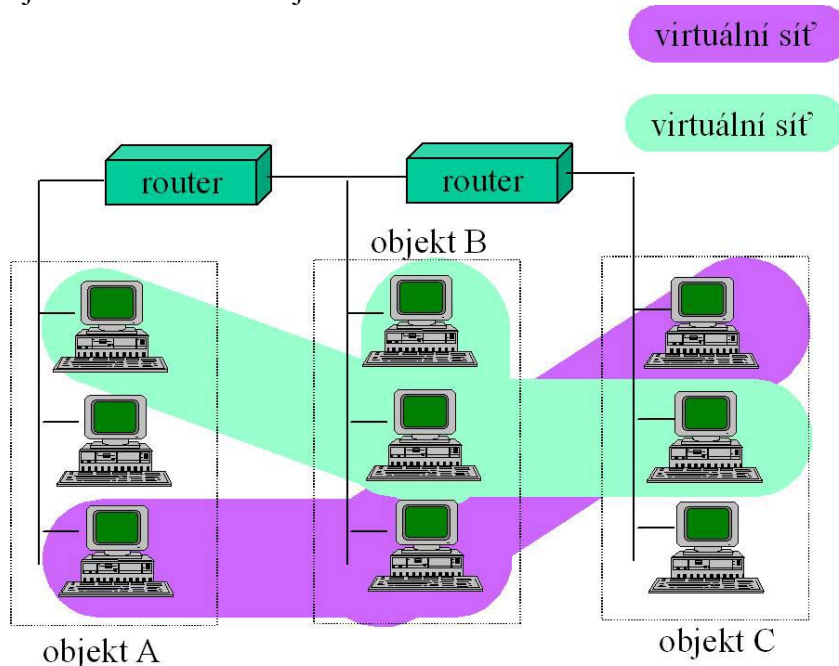
Poznámka - čím větší jsou bloky dat, přenášené najednou, tím větší je rozdíl mezi přepojováním paketů a přepojováním okruhů - např. z hlediska podpory přenosu zvuku a obrazu. Zmenšováním velikosti přenášeného bloku se rozdíly zmenšují, při extrémně malých blocích (buňkách, resp. při přepojování buněk) se rozdíly téměř ztrácejí. ATM vyhovuje potřebám „světa spojů“ i „světa počítačů“.

Dalším důležitým pojmem je routing a switching. **Routing** je přepojování na úrovni síťové vrstvy, kdy se bere do úvahy topologie celé sítě. Přitom se vyžaduje náročnější rozhodování o dalším směru přenosu dat, obecně je složitější a pomalejší a většinou se řeší softwarově, nelze snadno „zadrátovat“. **Switching** je přepojování na úrovni linkové vrstvy a bere v úvahu jen nejbližší okolí uzlu. Rozhodování o dalším směru přenosu je jednoduché a tudíž je obecně jednodušší a rychlejší a lze „zadrátovat“ (řešit přímo v HW).

Na tomto místě je důležité vysvětlení pojmu síť VLAN (virtuální síť LAN - Virtual LAN). Doposud se počítače zařazovaly do samostatných sítí, mezi kterými dochází k „routování“ což je dáno fyzickým umístěním jednotlivých uzlů. Nyní při zavádění VLAN fyzické umístění nehraje roli, je to záležitost

## Počítačové sítě

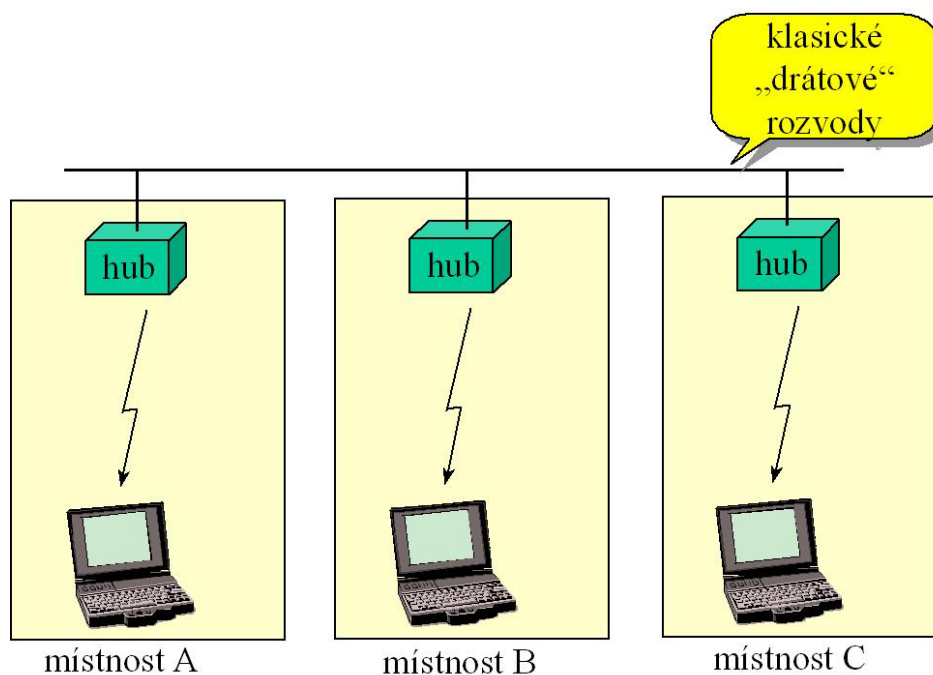
logická a o zařazení do určité sítě rozhoduje správce, pomocí konfiguračních nástrojů. Představa VLAN je zobrazena na obr. 1.14.



Obr. 1.14 Virtuální síť LAN

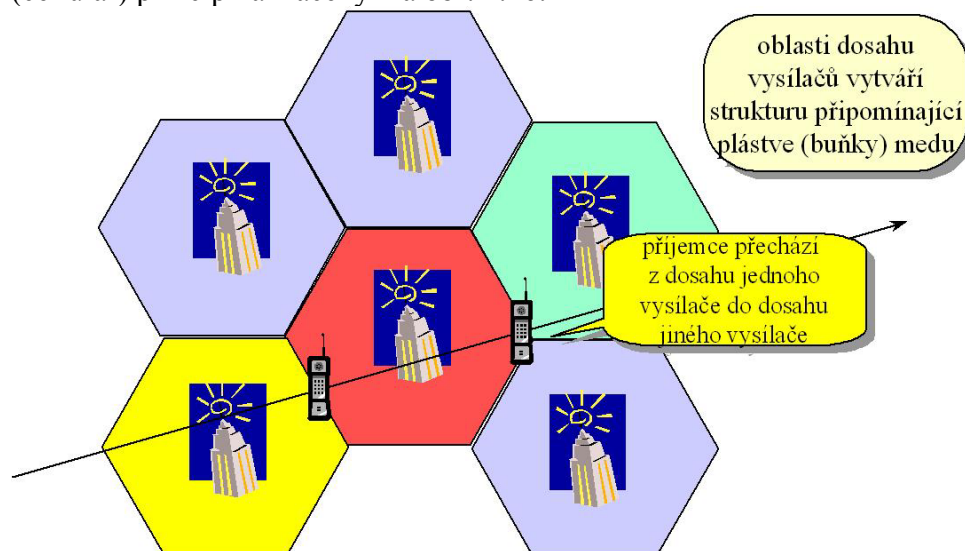
Dalším kritériem dělení sítí je dle použité přenosové technologie. Toto dělení se týká hlavně nejnižších vrstev tj. fyzické, linkové a někdy síťové. Zde je dělení na Ethernet, Token Ring pro sítě LAN a pro WAN na X.25, Frame relay, PPP, SLIP, HDLC a univerzálně ATM.

Jiné kritérium dělení je dle použité přenosové cesty a míry mobility uživatelů. Použité přenosové cesty podmiňují možnost pohybu uživatelů. Drátové přenosové cesty vylučují mobilitu kdežto bezdrátové ji připouští. Za bezdrátovou považujeme takovou síť, která používá bezdrátové přenosové cesty, umožňuje plnou mobilitu uživatelů. Bezdrátové přenosové cesty mohou nahrazovat některé „drátové“ části přenosových okruhů v sítích WAN přičemž to nemá vliv na mobilitu uživatelů. Bezdrátově může být řešena „účastnická zásuvka“ (připojení uživatele počítače na nejbližší rozvaděč, tzv. rádiový hub) - „cordless“ (bezešňůrově) zde se většinou nepočítá s mobilitou uživatele (nebo např. jen v rámci místnosti). Důvodem použití je nejčastěji nemožnost pokládky klasických drátových rozvodů (např. v památkově chráněných objektech). Představa bezdrátového rozvodu je na obr.1.15.



Obr. 1.15 Bezdrátové rozvody

Dalším pojmem je roaming. Tato služba umožňuje uživateli přecházet z dosahu jednoho rádiového hub-u do dosahu jiného, a je jím „přebírán“. Některé bezdrátové sítě LAN (spíše „cordless“) umožňují roaming mezi svými rádiovými hub-y. Tím se řeší omezená mobilita uživatele, například v rámci budovy či jiného objektu. Pro obecnou mobilitu se používá tzv. buňkový (cellular) princip naznačený na obr. 1.16.



Obr. 1.16 Cellulární síť

Posledním kritériem členění je dle míry strategie centralizace. Síť rozdělujeme na systémy:

- *distribuovaného zpracování* - společná data neexistují vůbec
- *hierarchického zpracování* - data jsou rozdělena lokálně - v uzlech, centrálně - hlavní počítač, v jednom místě



## Počítačové sítě

- *centralizovaného zpracování* - všechna data jsou trvale udržována na jediném místě.

### Úkoly k zamyšlení:

1. Pokuste se síť, se kterou pracujete v zaměstnání, ve škole či jinde, zařadit do skupin podle všech výše uvedených kritérií. (např. síť lokální, na bázi protokolové sady SPX/IPX, serverového typu, privátní síť, ...)
2. Může síť patřit do více skupin zároveň?



### Korespondenční úkoly:

1. Uveďte příklad použití více výpočetních modelů v jedné síti, nejlépe z praxe!
2. Jaký výpočetní model (jaké výpočetní modely) dnes používáte ve své praxi (ilustrovat prosím podrobnějším popisem reálné situace)?
3. Několika (nejvýše 10) větami či heslovitě popište počítačovou síť, s kterou v praxi (ve svém zaměstnání, škole apod.) pracujete. Pokud je takových více, vyberte tu, se kterou pracujete nejčastěji (tu byste měli nejlépe znát). Pokud máte možnost, zkuste v této věci konzultovat popis s Vaším správcem sítě.



## 2 Funkce sítí PC-LAN

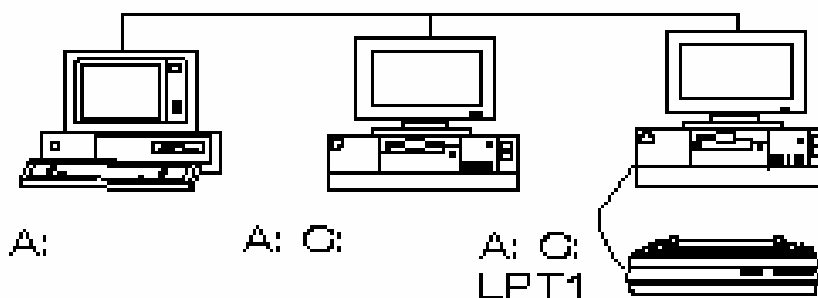
Síť PC-LAN je vlastně skupina počítačů, které jsou navzájem propojené tak, aby byla možná jejich vzájemná komunikace. Uživatelům mohou poskytovat následující služby :

1. Sdílení technických zařízení sítě
2. Sdílení společných dat sítě
3. Elektronickou poštu mezi uživateli sítě
4. Monitorování jiných účastníků sítě
5. Komunikaci v síti

Sítě poskytují svým uživatelům nejčastěji první dva druhy služeb. Kromě těchto služeb bývá nejvíce využívanou službou *elektronická pošta*, která umožňuje zasílat soubory jednotlivým uživatelům sítě. Její realizace je u jednotlivých sítí odlišná.

### 2.1 Sdílení technických zařízení sítě

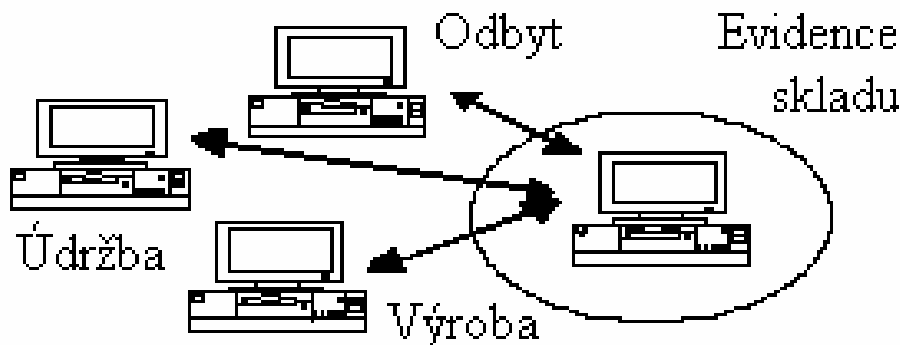
Síť LAN umožňuje používat všem účastníkům sítě společná technická zařízení. Obvykle se jedná o velkokapacitní disky, laserové tiskárny, a další speciální hardware. Tuto službu si vynutily ekonomické důvody, aby byl zajištěn kompromis mezi vysokou cenou některých zařízení a potřebou jejich používání více účastníky. Na obr. 2.1 je příklad sítě se třemi počítači. Jeden má k dispozici jenom jednotku pružného disku „A:“, druhý má i pevný disk „C:“ a třetí má i laserovou tiskárnu. Síť poskytuje potom každému uživateli sítě laserovou tiskárnu, pevný disk, nebo jednotky pružných disků.



obr. 2.1 Sdílení síťových zdrojů

### 2.2 Sdílení společných dat

Tento důvod bývá nejčastější příčinou budování počítačových sítí LAN. Všichni uživatelé sítě mohou v síti využívat a zpracovávat společná data. Tato služba se používá, jakmile potřebuje větší počet pracovníků přístup ke stejným datům. Takovýto případ je obr. 2.2, na kterém je síť počítačů libovolného podniku. S daty, která jsou na počítači skladu, kde je evidence skladu, mohou pracovat i pracovníci odbytu, údržby a výroby.



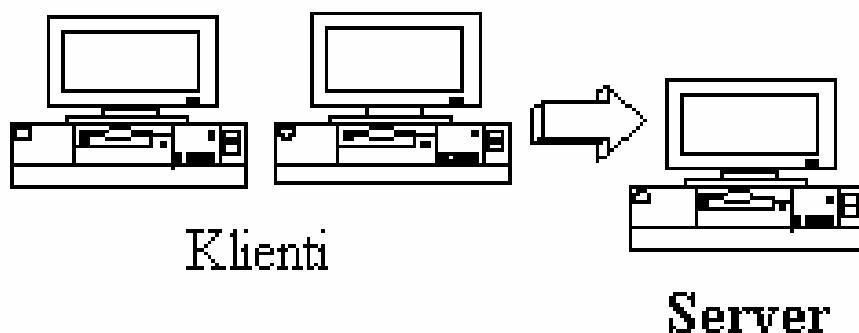
Obr.2.2: Síťové řešení v imaginární firmě

### 2.3 Elektronická pošta

Používá se velmi často, přičemž umožňuje zasílání a výměnu textových zpráv, souborů a programů mezi uživateli sítě. U různých operačních systémů bývá realizovaná různým způsobem.

### 2.4 Monitorování jiných účastníků sítě

Výhodou služby je hlavně kontrola práce v síti. Monitorování umožňuje zobrazovat na obrazovce počítače, obsahy obrazovek jiných počítačů. Možnost využití této služby prezentuje obr. 2.3, kde mohou klientské počítače sledovat obsah obrazovky serveru.

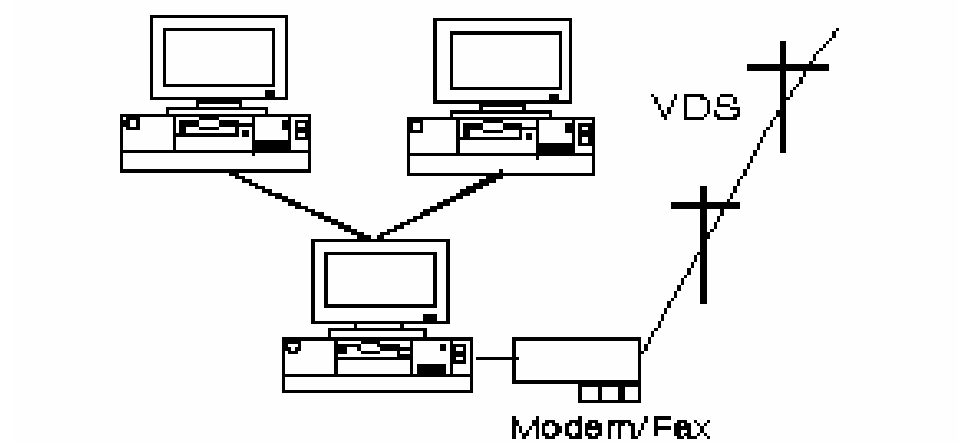


Obr.2.3: Monitorování síťových zdrojů

### 2.5 Komunikaci v síti

Řízení jiných počítačů, či posílání dat umožňuje uživateli sítě pracovat prostřednictvím svého počítače na vzdálených počítačích sítě. Příklad takovéto aplikace ukazuje obr. 2.4, na kterém může libovolný počítač sítě pracovat s modemem, který je v skutečnosti připojený na jiný počítač sítě.

## Počítačové sítě



Obr.2.4: Vzdálený přístup k datům pomocí modemu



### Korespondenční úkol:

Jaké funkce počítačové sítě, které byly popsány v kapitole 2, nejčastěji používáte? Existují také takové funkce, které nevyžíváte vůbec? Pokud ano, proč?

### 3 Struktura sítí PC-LAN

Počítače zapojené do sítě PC-LAN mohou mít funkci *SERVERu*, nebo *pracovní stanice* (WORKSTATION). Technickým vybavením se obvykle liší málo (spíše jen ve výkonnosti), vždy se však liší svým programovým vybavením.

#### 3.1 SÍŤOVÝ SERVER

Je počítač, který poskytuje ostatním počítačům síť svoje technické zařízení, anebo svoje data. Slouží tedy potřebám sítě, z čehož plyne i jeho název *server - sluha*. Servery mohou být dvojího druhu :

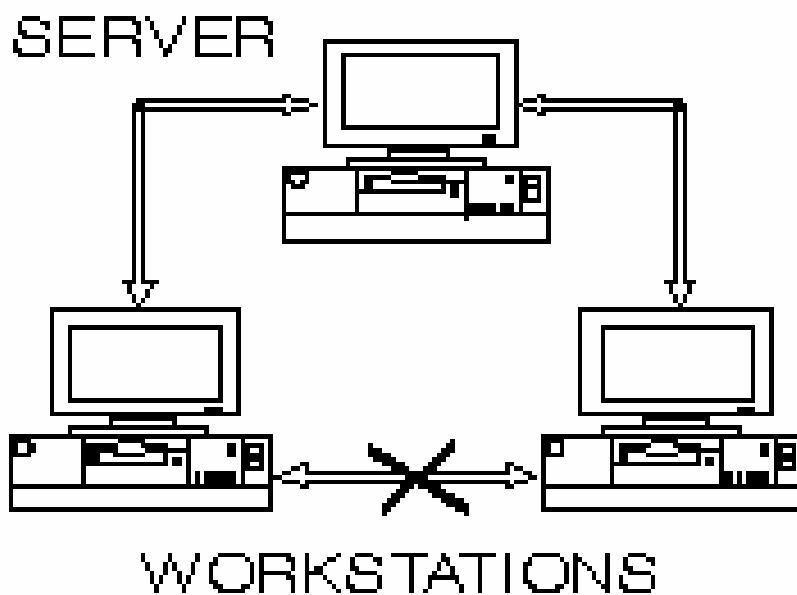
- *vyhrazené* (dedicated) - takovéto servery slouží jen pro potřeby sítě, není je možné používat na jiné účely
- *nevyhrazené* (non dedicated) - vedle práce v síti, je možné pracovat i na jiných aplikacích

#### 3.2 WORKSTATION (Pracovní stanice)

Je počítač, který využívá služby sítě. Má k dispozici data a technická zařízení poskytovaná servery. Obzvláště výhodnými jsou tzv. *bezdiskové stanice* (diskless stations), které nemají pevný a pružný disk (Net-PC). Pracují proto výhradně na síti, přičemž operační systém získávají ze serveru tzv. *REMOTE BOOT*.

#### 3.3 Komunikace na síti

Komunikace počítačů na síti probíhá prostřednictvím serverů. V případě elektronické pošty, když jeden uživatel posílá druhému poštu, je pošta od odesílatele předaná na server, který ji uloží do "příhrádky" adresáta. Tuto komunikaci zachycuje obr. 3.1.

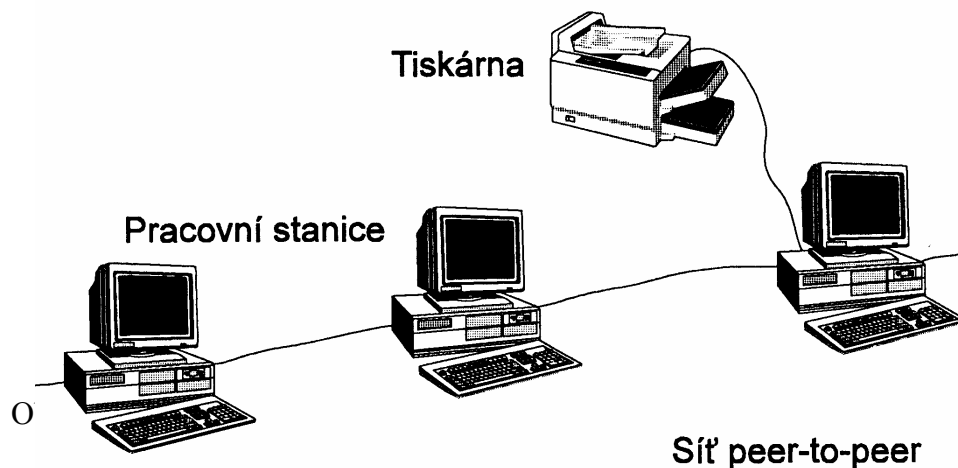


Obr. 3.1: Přístupy pracovních stanic k serveru

Počet stanic na síti není výrazně omezený. U serverů je však situace jiná. Některé sítě používají *centralizovaný server*, který slouží pro celou síť. Protikladem této koncepce je koncepce *PEER TO PEER* (rovný s rovným).

### 3.4 KONCEPCE PEER TO PEER

Je protikladem centralizované koncepce sítě. Umožňuje aby každý počítač na síti byl serverem i pracovní stanicí. Výhodou této koncepce je skutečnost, že si všechny počítače na síti mohou poskytovat navzájem svoje data a technická zařízení. Naopak nevýhodou v praxi je, že se většinou ztrácí data a nikdo nemůže najít data, které hledá, čímž vzniká chaos.



### 3.5 Zabezpečení sítě

Úlohou zabezpečení sítě je zabránit nepovolaným osobám dostat se do sítě a získat tak přístup k cenným datům. Obvykle se používá třístupňové zabezpečení sítě.

**1.stupeň** - Každý uživatel má v síti přiřazené svoje jméno a heslo. Při přihlašování na síť musí uvést správné jméno a heslo aby se dostal do sítě. Obvykle má i omezený čas přístupu na síť.

**2.stupeň** - Jakmile se uživatel přihlásí na síť má k dispozici jen adresáře, které mu určí správce. Taktéž má určené činnosti, které může v těchto adresářích vykonávat. Tím jsou chráněná data, která mu nepatří.

**3.stupeň** - Při práci v síti se většinou tvoří kontrolní záznam, v kterém je zachycena činnost v síti. Tímto postupem je možné mapovat činnost jednotlivých uživatelů na síti.

#### Úkol k zamyšlení:

Může být jedna počítačová síť být zároveň sítí serverového typu a sítí peer-to-peer?

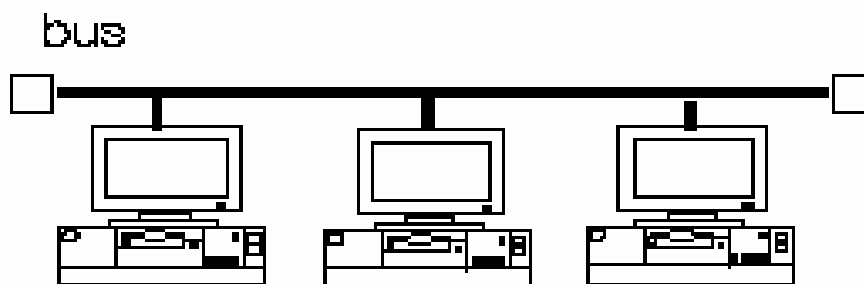
## 4 TOPOLOGIE, PŘÍSTUPOVÉ METODY, ROZDĚLENÍ SÍTÍ PC-LAN

### 4.1 Topologie sítí PC-LAN

Topologie sítě je způsob vzájemného propojení účastníků sítě. U sítí PC-LAN se setkáváme nejčastěji s následujícími druhy topologií :

#### 4.1.1 SBĚRNICOVÁ TOPOLOGIE

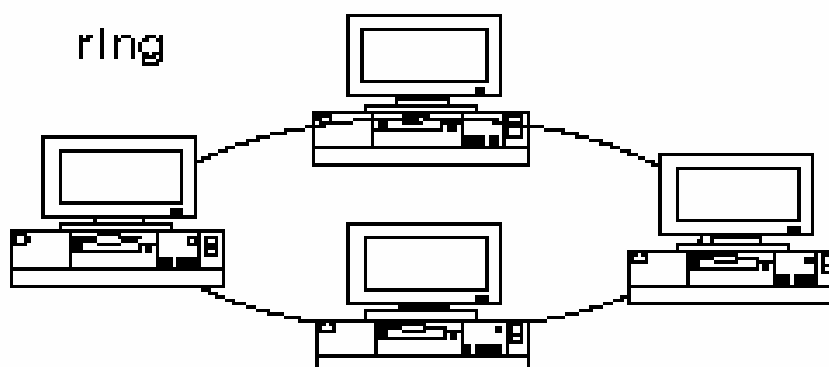
Všichni účastníci sítě jsou připojeni paralelně na společnou sběrnici. Výhodou této metody je hlavně lehká instalace a připojení nových účastníků do sítě. Nevýhodou je, že porucha libovolného počítače způsobí výpadek celé sítě. Tato topologie je zobrazena na obr. 4.1.



Obr.4.1: Sběrníková topologie

#### 4.1.2 KRUHOVÁ TOPOLOGIE

Jednotlivé počítače sítě jsou spojené přenosovým médiem do kruhu, takže signál prochází postupně přes všechny počítače sítě. Nevýhodou však je podstatně horší instalace sítě a skutečnost, že porucha libovolného počítače může způsobit neprůchodnost sítě. Kruhovou topologii zobrazuje obr. 4.2.



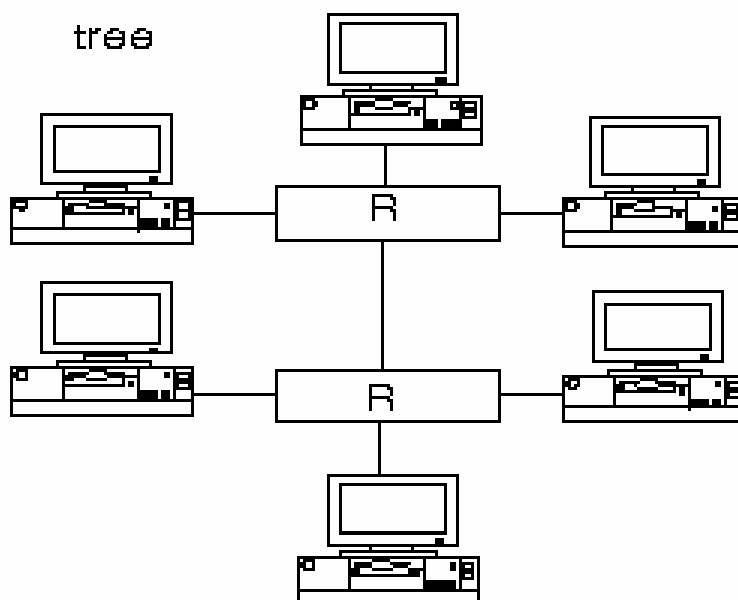
Obr.4.2: Kruhová topologie



### 4.1.3 HVĚZDICOVÁ (STROMOVÁ) TOPOLOGIE

Jednotlivé počítače jsou propojené navzájem pomocí rozvětvovačů (hubů). Výhodou této koncepce je lehké rozšiřování sítě a velký dosah sítě.

Příklad této koncepce zobrazuje obr. 4.3.



Obr.4.3: Hvězdicová topologie

## 4.2 Přístupové metody

Při práci v síti mohou přijímat zprávy všichni uživatelé sítě najednou. U vysílání zpráv je však potřebné zajistit, aby na přenosovém médiu byla současně zpráva jen od jednoho účastníka sítě. V opačném případě by totiž docházelo ke kolizím, tedy vzájemnému rušení a zkracování vysílaných zpráv. Proto je úlohou přístupové metody zajistit současně vysílání dat jen jednomu uživateli sítě. V současnosti se používají v sítích PC-LAN dvě základní přístupové metody *CSMA/CD* a *TOKEN PASSING*.

### 4.2.1 CSMA/CD (Carrier sense multiple access/collision detection)

Při této metodě jednotlivé počítače sítě zjišťují stav na přenosovém médiu. Jestliže libovolný počítač sítě vysílá, ostatní jsou blokovány až do doby dokud se neuvolní sběrnice. Po uvolnění může začít vysílat další počítač. Jestliže však čeká na uvolnění sběrnice více počítačů najednou může dojít při současném vysílání ke kolizi dat. Při velkém provozu na síti roste riziko kolizí a výrazně klesá i výkon sítě. Zlepšení přineslo až tzv. *dodatečné sledování sběrnice*, kdy každý vysílající počítač současně sleduje stav sběrnice. Jakmile se liší vysílaný signál od signálu na sběrnici, vyhodnotí se tento stav jako kolize a přestane se vysílat.

### 4.2.2 TOKEN PASSING

Libovolný počítač sítě může začít s vysíláním dat až tehdy, když získá tzv. *vysílací právo* - *TOKEN*. Jakmile ho získá může začít vysílat data. Tyto data jdou postupně přes jednotlivé počítače sítě, až se dostanou k adresátovi, který je přečte, označí a pošle dále. Po jejich přijetí odesílatelem, dojde k předání vysílacího práva ve formě speciálního paketu dalšímu počítači sítě. Výhodou této metody je, že každý počítač má zaručené získání vysílacího práva. Tyto sítě jsou proto vhodné i na řízení technologických procesů, kde se vyžaduje práce v reálném čase. Metoda se osvědčila hlavně u sítí s kruhovou topologií.



#### **Kontrolní otázky:**

Jakou základní odlišnost mají metody CSMA/CD a Token Passing?

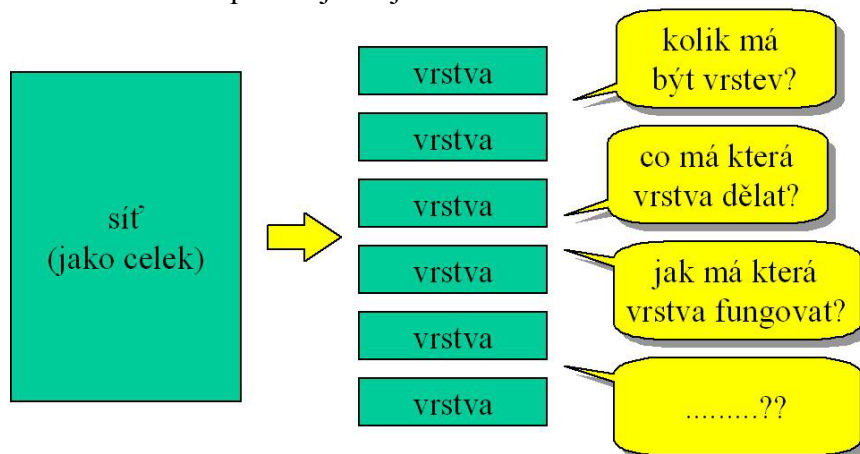


#### **Korespondenční úkoly:**

1. Jakou fyzickou topologii má Vámi nejčastěji používaná počítačová síť? Pokuste se nakreslit její schéma.
2. Víte, se kterými metodami řízení přístupu k médium pracují Vámi používané sítě? Pokud ne, pokuste se to zjistit např. u správce sítě!

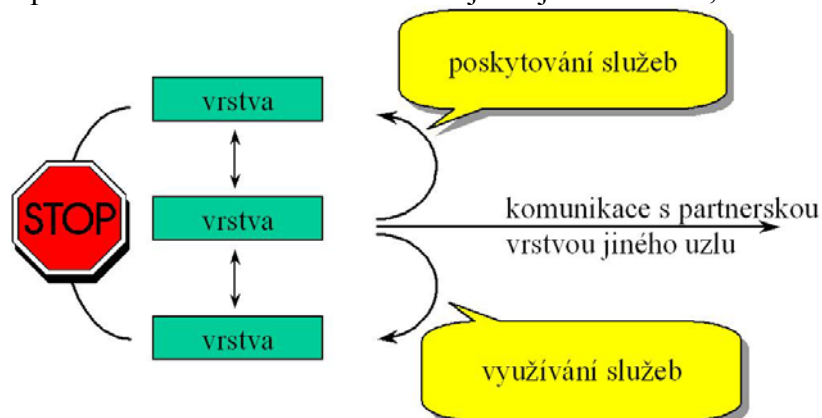
## 5 Síťový model a síťová architektura

Implementovat funkční síť je hodně složité a náročné, jedná se o obdobnou situaci jako při řešení velkých SW celků. Jde o jeden velký problém, který se vyplatí dekomponovat tj. rozdělit na menší části, které je možné řešit samostatně. Dekompozice se provede po hierarchicky uspořádaných vrstvách, což dobře odpovídá povaze řešeného problému. Přináší to i další výhody jako je možnost alternativních řešení na úrovni nižších vrstev, větší modulárnost. Představa dekompozice je zřejmá dle obr. 5.1



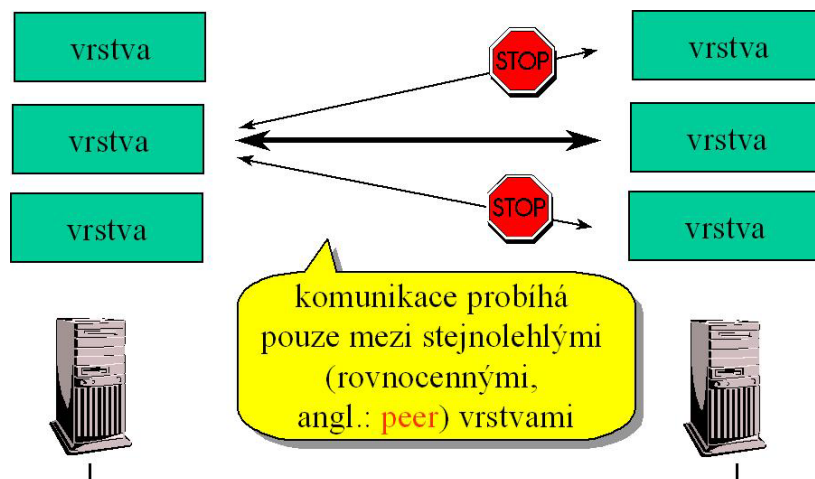
Obr. 5.1 Dekompozice sítě na vrstvy

Způsob komunikace mezi vrstvami je zřejmá z obr.5.2, 5.3 a 5.4.

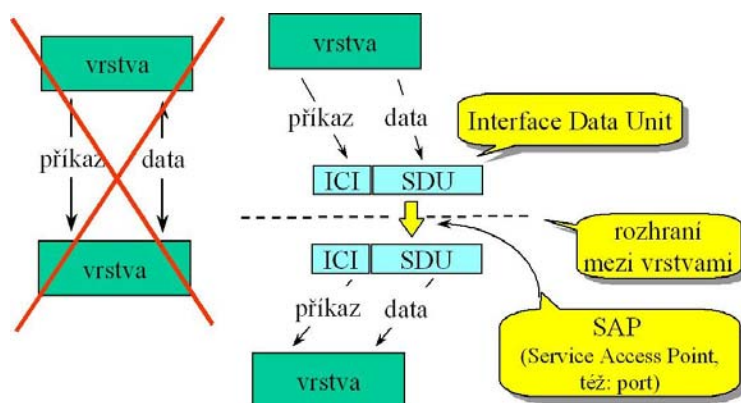


Obr. 5.2 Komunikace mezi sousedními vrstevmi

## Počítačové sítě

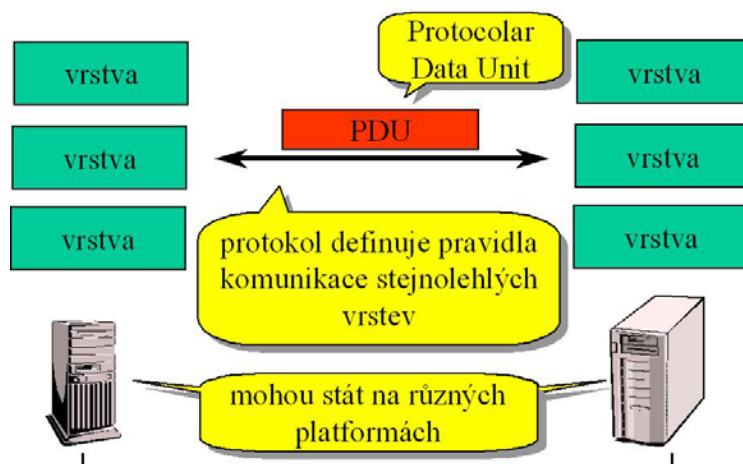


Obr. 5.3 Komunikace mezi stejnohlými vrstvami

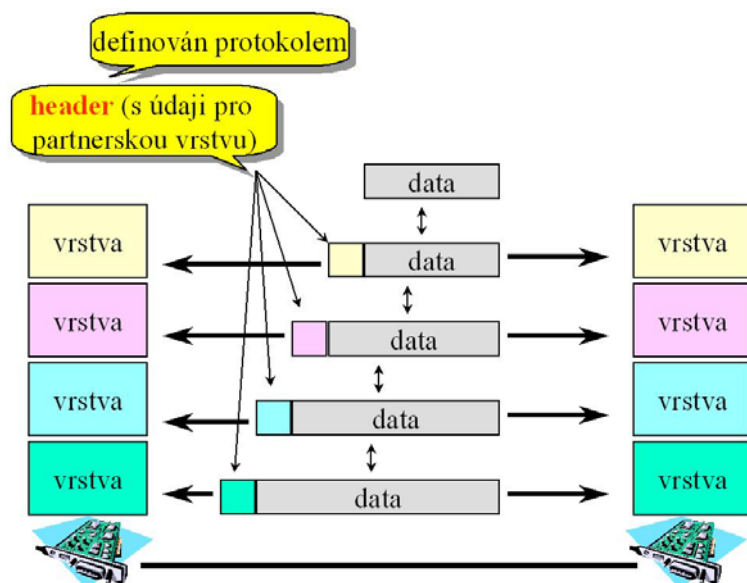


Obr. 5.4 Komunikace mezi sousedními vrstvami

V této souvislosti je nutné vysvětlit pojem protokol – viz. Obr 5.5 a 5.6. Protokol definuje pravidla komunikace na stejnohlých vrstvách, přičemž komunikující subjekty mohou pracovat na různých platformách.



Obr. 5.5 Definice protokolu



Obr. 5.6 Protokoly na jednotlivých vrstvách

Z uvedených obrázků vyplývá, že vrstvy nejsou „jednotlivé“, v každé vrstvě může existovat a fungovat několik relativně samostatných entit. Pod pojmem entita může chápat např. proces, démon, úlohu apod. Entity ve stejné vrstvě mohou plnit rozdílné funkce (nekonkurovat si), nebo plnit obdobné funkce (ale jiným způsobem, tj. konkurovat si). Protokol definuje pravidla komunikace mezi entitami stejnohlých vrstev. Každý protokol vždy „patří“ do určité konkrétní vrstvy a určuje způsob, jakým je realizována určitá služba. Pro každou vrstvu může existovat několik alternativních protokolů přičemž současné použití různých protokolů (v rámci téže vrstvy) se nemusí vylučovat.

Síťový model je ucelená představa o tom, jak mají být sítě řešeny. Zahrnuje představu o počtu vrstev představu o tom, co má mít která vrstva na starosti. Nezahrnuje konkrétní představu o tom, jak má která vrstva své úkoly plnit. Síťová architektura obsahuje navíc také konkrétní představu o způsobu fungování jednotlivých vrstev tj. obsahuje i konkrétní protokoly. Příkladem síťového modelu je referenční model ISO/OSI a příkladem síťové architektury je TCP/IP.

V několika uplynulých letech byla vyvinuta řada síťových norem. Některé rozhodující organizace v tomto oboru vytvořily protokoly nebo pravidla, které zajišťují kompatibilitu pro síťový hardware a software různých výrobců.

Dosud jsme si všimli hlavních komponentů sítě LAN. Kdyby byly počítače, aplikační programy, síťový software a kabeláž vyrobeny stejným dodavatelem, nebyl by žádný problém zajistit, aby vše hladce spolupracovalo. Dnešní realita je však taková, že obvykle síťový software jednoho výrobce LAN nebude pracovat v síti jeho konkurenta, takže aplikační programy a dokonce i kabeláž musí být vybrány pro určitou síť LAN. Aby bylo dosaženo alespoň nějaké úrovně sjednocení mezi různými dodavateli sítí, vytvořila organizace ISO normy nazvané OSI (Open Systems Interconnection). Různé počítače vzájemně propojené do sítě musí vědět, v jaké formě budou přijímat informace. Kde je začátek určitého slova, kde je jeho konec a kde začne slovo následující?

Má počítač možnost zjistit, zda byla zpráva při přenosu zkreslena? Model OSI odpovídá na tyto a další otázky pomocí řady norem, které by měly v budoucnu umožnit veřejnosti, aby si kupovala síťové prvky různých dodavatelů s určitou jistotou, že budou spolupracovat.

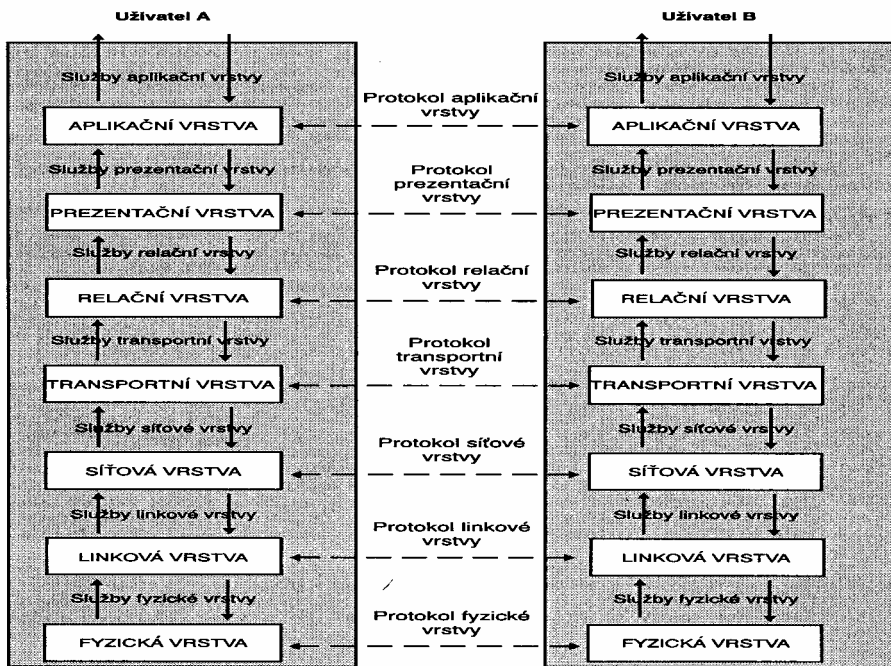
Referenční model ISO/OSI byl pokusem vytvořit univerzální síťovou architekturu, skončil jako síťový model bez protokolů, ty se dodělávaly postupně. Pochází „ze světa spojů“ od organizace ISO (International Standards Organization, správně: International Organization for Standardization) - členy ISO jsou národní normalizační instituce. Byl „oficiálním řešením“ dnes je prakticky odepsaný, prohrál v souboji s TCP/IP.

Referenční model ISO/OSI reagoval na vznik proprietárních a uzavřených sítí IBM SNA. Prvotním záměrem bylo definovat, jak mají vypadat otevřené systémy. Odsud je název Open Systems Architecture. Chování „uvnitř“, nejen „mezi sebou“ ukázalo se jako příliš náročné, došlo k redukci ambic. Proto nastalo druhé přiblížení, které se týkalo jen vzájemného propojení otevřených systémů a nastala změna názvu na Open Systems Interconnection Architecture. Opět se ukázalo jako příliš náročné a proto nastalo třetí přiblížení, které neobsahovalo konkrétní protokoly, ale jen představu o počtu vrstev a o tom, co má která vrstva dělat. Poslední iterace názvu je tedy Open Systems Interconnection. Byly „odstraněny“ protokoly, zbyl jen síťový model, tedy obecný „rámec“, do kterého jsou konkrétní řešení „zasazována“ a konkrétní protokoly pro RM ISO/OSI jsou vyvíjeny samostatně a teprve dodatečně jsou zařazovány do rámce ISO/OSI. Filosofie RM ISO/OSI vznikala „od zeleného stolu“ a pak byla „nadiktován“ uživatelům. Vznikala maximalistickým způsobem a autoři se snažili zahrnout „vše, co by někdy někomu mohlo hodit“. Výsledek byl dosti odtahitý od reálné praxe, celá řešení se často ukázala jako nerealizovatelná, a hledala se implementovatelná podmnožina avšak, vzájemně kompatibilní. Mnohé výchozí předpoklady se ukázaly jako chybné. Autoři ISO/OSI se dosti dlouho přeli o počtu vrstev, nakonec zvítězil návrh na 7 vrstev a dnes se to zdá být zbytečně mnoho. Kritéria pro volbu vrstev byly stanoveny následovně:

- činnosti na stejném stupni abstrakce mají patřit do stejné vrstvy
- odlišné funkce by měly patřit do odlišných vrstev
- aby bylo možné převzít již existující standardy
- aby datové toky mezi vrstvami byly co nejmenší
- aby vrstvy byly rovnoměrně vytíženy.

Standard OSI (Open Systems Interconnection) představuje model se sedmi vrstvami, který zajišťuje účinnou komunikaci v rámci sítě LAN a také mezi různými sítěmi. Model OSI se skládá ze sedmi vrstev specifikací, které popisují manipulaci s daty při jednotlivých fázích přenosu. Každá vrstva poskytuje určité služby vrstvě, jež je bezprostředně nad ní. Sedm vrstev ISO/OSI je na obr. 5.7.

## Počítačové sítě



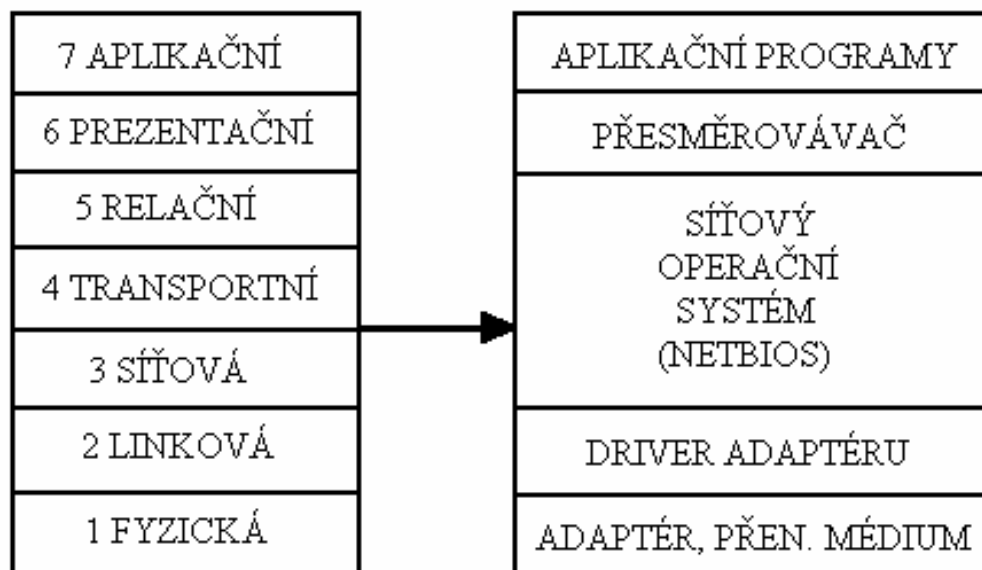
Obr. 5.7 Vrstvy ISO/OSI

Orientace vrstev je zřejmá na obr. 5.8



Obr. 5.8 Orientace vrstev ISO/OSI

Každá vrstva vykonává vlastní služby přičemž zajišťuje návaznost na sousední vrstvy viz. obr. 5.9



Obr. 5.9 Vrstvy modelu ISO/OSI

Model OSI přiřazuje složitým procedurám, nezbytným pro komunikaci v síti, sedm různých vrstev. Model je navržen tak, aby bylo možno snadno dosáhnout výchozí dohody na nižších vrstvách a konečně na všech sedmi vrstvách.

## 5.1 Fyzická vrstva

*Standardy fyzické vrstvy se týkají technických norem zajišťujících kompatibilitu sítí. Zahrnují použité napěťové úrovně, časování přenosu dat a mechanismy vzájemného potvrzování.*

První vrstva (fyzická) představuje řadu pravidel týkajících se technického vybavení užitého pro přenos dat. Zabývá se takovými věcmi, jako jsou napěťové úrovně, časování přenosu dat a pravidla pro komunikaci, která slouží k vytvoření spojení. Fyzická vrstva určuje, zda mají být bity odesílány v poloduplexním režimu (to je podobné způsobu, jakých se data předávají pomocí radiostanice) nebo v plně duplexním režimu, což vyžaduje současné vysílání i příjem dat. Další technická specifikace, kterou pokrývá fyzická vrstva, zahrnuje konektory a rozhraní na přenosová média. Na této úrovni se model OSI zabývá elektrickými veličinami a bity (nulami a jedničkami). Bity nemají na této úrovni žádný skutečný význam. Přiřazení významu provádí až další vrstva OSI. Fyzická vrstva se zabývá výhradně přenosem bitů (bez ohledu na jejich význam), otázkami typu kódování, modulace, časování, ynschronizace, el. parametry signálů, konektory, řídící signály rozhraní. Nabízí služby typu přijmi bit a odešli bit a musí zajistit, že v případě vyslání jedničkového bitu jej druhá strana přijme jako jedničkový a ne jako nulový. Nijak neinterpretuje to, co přenáší a jednotlivých bitům nepřisuzuje žádný specifický význam. Na úrovni fyzické vrstvy rozlišujeme paralelní a sériový přenos, synchronní, asynchronní a arytmičkový přenos, přenos v základním a přeloženém pásmu. Standardem fyzické vrstvy je RS-232-C, V.24, X.21.



### 5.2 Linková (spojová) vrstva

*Spojová vrstva se zabývá shlukováním dat do rámců (frame), v nichž jsou přenášena.*

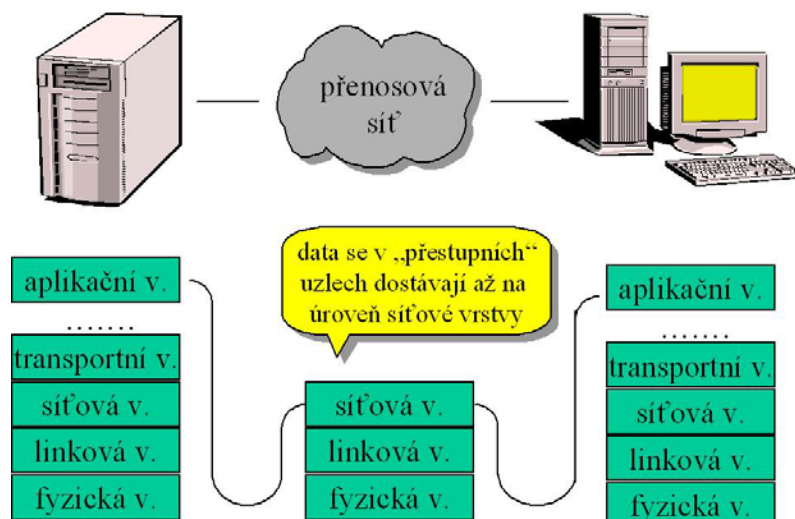
Model OSI je navržen tak, že každá vrstva poskytuje vyšší vrstvě určitý klíčový prvek. Fyzická vrstva předává spojové vrstvě bity. A nyní přichází okamžik, kdy je těmto bitům nutno dát nějaký význam. V tomto bodě se už nezabýváme bity, nýbrž datovými rámci (pakety), obsahujícími jak data, tak řídicí informace. Spojová vrstva přidává na začátek a na konec zprávy značky (flag). Normy této vrstvy provádějí dvě důležité funkce. Zaručují, že data nejsou zaměněna za značky a prověřují výskyt chyb uvnitř datového rámce. Toto prověřování chyb může probíhat tak, že se na stranu přijímače odešle informace o určitém datovém rámci, a pak se čeká na potvrzení, že vše bylo správně přijato. Linková vrstva (spojová vrstva) přenáší celé bloky dat tzv. rámce (frames) a zajišťuje přenos pouze v dosahu přímého spojení tj. bez „přestupních stanic“. Může fungovat spolehlivě či nespolehlivě, spojovaně či nespojovaně a může využívat různé přenosové technologie tj. linkové i bezdrátové. Úkolem je synchronizace na úrovni rámců (správné rozpoznání začátku a konce rámce, i všech jeho částí), zajištění spolehlivosti (detekce chyb a náprava), řízení toku (aby vysílající nezahltl příjemce) a přístup ke sdílenému médiu (řeší konflikty při vícenásobném přístupu ke sdílenému médiu). Linková vrstva řídí přístup uživatele na síť a tvoří obálku paketů. Úkolem je změnit prosté komunikační vybavení na spoj, který se vůči síťové vrstvě chová jako bezchybný.

### 5.3 Síťová vrstva

*Síťová vrstva se zabývá přepojováním paketů. Vytváří virtuální spojení pro datovou komunikaci mezi počítači nebo terminály.*

Třetí vrstva modelu OSI, síťová vrstva, se zabývá přepojováním paketů. Vytváří virtuální spojení (trasu mezi dvěma počítači nebo terminály) pro datovou komunikaci. U odesílatele zformuje síťová vrstva zprávu z transportní vrstvy do datových paketů, které pak mohou nižší dvě vrstvy přenášet. Na straně přijímače zrestauruje síťová vrstva původní zprávu. Abychom pochopili použití datových paketů, bude nezbytné podívat se na průmyslový standard, zahrnující spodní tři vrstvy modelu OSI. Je to standard X.25. Síťová vrstva přenáší bloky dat označované jako pakety (packets) a zajišťuje doručení paketů až ke konečnému adresátovi. V prostředí kde není přímé spojení hledá vhodnou cestu až k cíli tj. zajišťuje tzv. směrování (routing). Musí si uvědomovat skutečnou topologii celé sítě (obecně) přičemž používá různé algoritmy směrování jako adaptivní či neadaptivní, izolované či distribuované. Je poslední vrstvou, kterou musí mít přenosová infrastruktura – viz obr.5.10.

## Počítačové síť



Obr. 5.10 Přenosová infrastruktura

Hlavním úkolem síťové vrstvy je určení směrování dat (paketů) v síti ze zdroje do cíle a musí řešit problematiku zahlcení sítě ke které může dojít při velkém množství paketů. Bývá v ní zabudována také účtovací funkce ACCOUNT ke zjištění kolik který uživatel vyslal bitů.

### 5.4 Transportní vrstva

*Prvotní úlohou transportní vrstvy je rozpoznání chyb a zotavení po chybě, zabývá se však také multiplexem zpráv a regulací toku informací.*

Transportní vrstva modelu OSI má mnoho funkcí včetně několika úrovní rozpoznávání chyb a zotavení po chybě. Na nejvyšší úrovni může transportní vrstva rozpoznávat (či dokonce opravovat) chyby, odhalovat pakety, které byly odeslány v nesprávném pořadí, a přerovnávat je do pořadí správného. Tato vrstva také multiplexuje několik zpráv do jednoho spoje a vytváří hlavičky určující kterému spoji která zpráva patří. Transportní vrstva také reguluje tok informace tím, že řídí pohyb zpráv. Transportní vrstva zajišťuje komunikaci mezi koncovými účastníky (end-to-end komunikaci) přičemž může měnit nespolehlivý charakter přenosu na spolehlivý, méně spolehlivý přenos na více spolehlivý, nespojovaný přenos na spojovaný. Přitom vychází ze skutečnosti, že nelze „hýbat“ s vlastnostmi a funkcemi nižších vrstev. Vzhledem k tomu, že vyšší vrstvy mohou chtít něco jiného, než co nabízí nižší vrstvy, je úkolem transportní vrstvy zajistit potřebné přizpůsobení. Transportní vrstva rozkládá data na menší části tzv. pakety řeší problematiku komunikaci koncových uživatelů (end-to-end) přičemž využívá ke komunikaci záhlaví a řídicí zprávu (transportní záhlaví).

### 5.5 Relační vrstva

*Relační vrstva se věnuje řízení sítě. Ovládá rozpoznávání hesel, procedury při hlášení a odhlášení a rovněž dohlížení nad sítí a vytváření přehledových zpráv.*

Dosud jsme viděli, že model OSI se zabývá jenom bity a datovými zprávami a nerozlišuje jednotlivé uživatele v síti. Relační vrstvu si můžeme představit jako vrstvu, jejíž úlohou je řízení sítě. Je schopna přerušit určitou relaci a řídí řádné ukončení relací. Uživatel komunikuje přímo s touto vrstvou. Relační vrstva může prověřovat heslo zadané uživatelem a umožnit uživateli, aby přepnul z poloduplexního přenosu na plně duplexní. Dokáže určit, kdo hovoří, jak často a jak dlouho. Řídí přenosy dat a zodpovídá rovněž za zotavení systému po jeho výpadku. Konečně relační vrstva může sledovat využití systému a účtovat uživatelům spotřebovaný čas. Relační vrstva zajišťuje vedení relací. Relace může zajišťovat synchronizaci, šifrování a podporu transakcí. Relační vrstva kontroluje přístup uživatele a jeho programů na síť, dovoluje spojení uživatelů na různých typech zařízení, kteří mají mezi sebou zavedeny tzv. relace (session) a dovoluje, aby se uživatel mohl přihlásit ve vzdáleném víceuživatelském systému a přenesl soubor mezi dvěma počítači. Řeší řízení dialogu mezi jednotlivými počítači

### 5.6 Prezentační vrstva

*Bezpečnost sítě, přenosy souborů a formátovací funkce jsou úkoly prezentační vrstvy.*

Prezentační vrstva modelu OSI se věnuje bezpečnosti sítě, přenosu souborů a formátovacím funkcím. Na bitové úrovni je prezentační vrstva schopna kódovat data v řadě různých formátů včetně ASCII a EBCDIC. Americký standardní kód pro výměnu informací (ASCII) je kód pro přenos dat používající 7 bitů pro kódování znaku a osmý bit jako paritní. Je to kód používaný nejobecněji. Mnoho větších počítačů IBM používá rozšířený dvojkově kódovaný desítkový kód (EBCDIC - Extended Binary Coded Decimal Intechange Code). Prezentační vrstva musí být schopna použít pro přenos dat libovolný z těchto standardů. Prezentační vrstva má na starosti potřebné konverze z různých kódování znaků (ASCII, EBCDIC,...), formátu čísel, formátu struktur, polí a ukazatelů (pointerů). Nižší vrstvy se snaží doručit každý bit přesně tak, jak byl odeslán a přitom stejná posloupnost bitů může mít pro příjemce jiný význam než pro odesílatele. Prezentační vrstva určuje tvar dat v jakém jsou dostupné uživateli a přesměrovává požadavky uživatele na síť, provádí kódování jednotlivých informací v paketech - kryptografie z hlediska utajení před nepovolanými uživateli (ASCII, EBDIC) a způsob komprese a zhuštění informací pro zmenšení počtu přenášených bitů.

Pro dosažení komunikace musí prezentační vrstva v obou vzájemně komunikujících počítačích obsahovat tytéž protokoly neboli pravidla pro manipulaci s daty. Tato vrstva provádí rovněž konverzi protokolů mezi různými počítači používajícími různé formáty. Většina funkcí textových procesorů, které spojujeme s formátováním textu (stránkování, počet linek na obrazovce, pohyby kurzoru po obrazovce) je rovněž náplní prezentační vrstvy.

Práce s terminály, které mají nekompatibilní kódy, je rovněž zajišťována touto vrstvou. Terminálový protokol řeší odlišnosti tak, že umožní každému datovému terminálu aby fungoval jako stejný virtuální terminál. Výsledkem tohoto postupu je, že existuje řada překladových tabulek fungujících mezi

místním a vzdáleným terminálem. Lokální terminál vysílá datovou strukturu, která definuje okamžitý obsah jeho obrazovky v takových termínech, jako je zobrazený počet znaků na řádce. (Tento počet se může podstatně lišit. Mnoho terminálů zobrazuje 132 znaků na řádek, ale existují i jiné formáty.) Tato datová struktura přichází do řídicího prvku vzdáleného terminálu a ten převede tento počet na takový kód, kterému terminál rozumí a může jej použít. Další kódy se používají pro tučné písmo, podtržení, grafiku atd.

### 5.7 Aplikační vrstva

Síťové programy nacházející se v aplikační vrstvě zahrnují elektronickou poštu, řízení databází, software pro file servery a print servery. Aplikační vrstva zpracovává hlášení, vzdálená přihlašování a zodpovídá za statistiku řízení sítě. Na této úrovni najdete programy pro řízení databází, elektronickou poštu, programy pro file servery a print servery a příkazy operačního systému. Ve většině případů jsou funkce vykonávané touto vrstvou závislé na uživateli. Vzhledem k tomu, že různé uživatelské programy mají různé požadavky, je obtížné zevšeobecňovat protokoly, které se zde nalézají. Některá odvětví, např. bankovníctví, vyvinula na této úrovni řadu vlastních standardů. Aplikační vrstva původně měla obsahovat aplikace. Problémem je ale velké množství aplikací a ty by musely být všechny standardizovány, což by nemělo ani smysl. Později bylo definováno pouze „jádro“ aplikací, které má smysl standardizovat, například přenosové mechanismy elektronické pošty a ostatní části aplikací (typicky uživatelská rozhraní) byly vysunuty nad aplikační vrstvu. Aplikační vrstva představuje vlastně úroveň aplikačních programů např. elektronická pošta, síťové databázové systémy, vytváří všeobecně platné protokoly pro přenos údajů mezi nekompatibilními terminály, definuje pohyby kurzoru po obrazovce s možností vytvoření virtuálního terminálu.



#### Úkol k zamyšlení

Proč se ve většině síťových modelů používá rozdělení do vrstev?

## 6 Síťové protokoly

Bloky dat předávané mezi koncovými účastníky obvykle označujeme jako *pakety*. V jejich formátech najdeme síťové adresy obou koncových účastníků a informace potřebné pro potvrzování a případně i řízení toku. Pakety mohou být předávány jako zcela nezávislé datagramy, nebo jako součást souvislejší komunikace po virtuálním kanále. Mezi nejdůležitější a nejpoužívanější síťové protokoly v lokálních sítích patří : *NetBEUI*, *IPX/SPX*, *TCP/IP*.

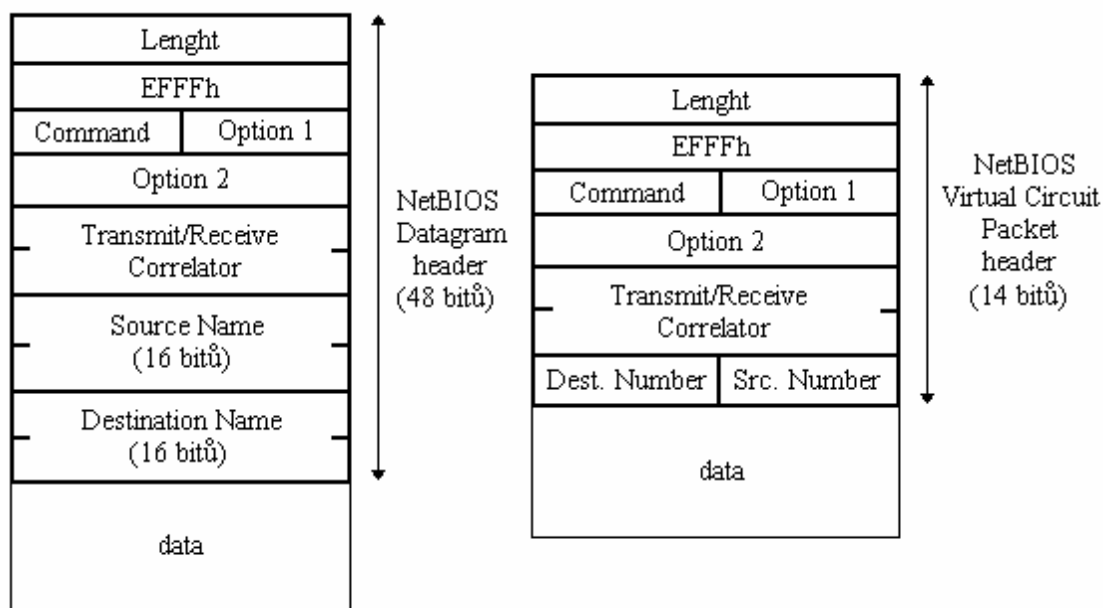
### 6.1 Protokoly NetBIOS a NetBEUI

Nejstarším síťovým protokolem určeným specificky pro prostředí lokální sítě (kde existuje možnost, aby rámec odeslaný jednou ze stanic sítě byl přijat všemi ostatními stanicemi) je *NetBIOS* navržený firmou IBM. Aplikace se identifikuje jménem a protokol pro správu NetBIOSu se stará o jedinečnost tohoto jména v síti. NetBIOS byl přímo vázán na ovladač komunikačního řadiče, stejným způsobem je implementován v LAN Manageru, kde je rozšířen, doplněn uživatelsky lepším rozhraním a pojmenován *NetBEUI* (NetBIOS Extended User Interface).

Aplikace musí pro vyžádání funkce NetBIOSu připravit požadavkový blok *NCB* (Network Control Block), ve kterém zadává parametry volání - jména, číslo logického kanálu, adresu a délku předávaných dat, časové limity pro vyslání a příjem. Požadavek předá aplikace NetBIOSu voláním systému (přerušení 5C<sub>H</sub>). Po předání požadavku může aplikace pokračovat ve své činnosti. Ukončení požadavku může aplikace aktivně testovat nebo lze ukončením požadavku aktivovat dokončovací rutinu.

Komunikující aplikace (nebo její komunikační kanály) jsou identifikovány jmény, která mají délku 16 znaků. Jméno může být buď individuální (a jedinečné v určité síti) nebo skupinové.

Základní služba, kterou NetBIOS podporuje, je datagramová služba (u protokolu NetBEUI jí odpovídá služba *MailSlot*) dovolující předání zprávy o délce do 512 Bytů jednomu adresátovi nebo libovolné stanici na síti, která takovou zprávu očekává (Broadcast). Virtuální kanály (u NetBIOSu relace, u NetBEUI služba Named Pipes) dovolují přenášet zprávy o délce 131071 znaků, které jsou při přenosu děleny do paketů. Na obr. 6.1 je struktura paketů protokolu NetBIOS.

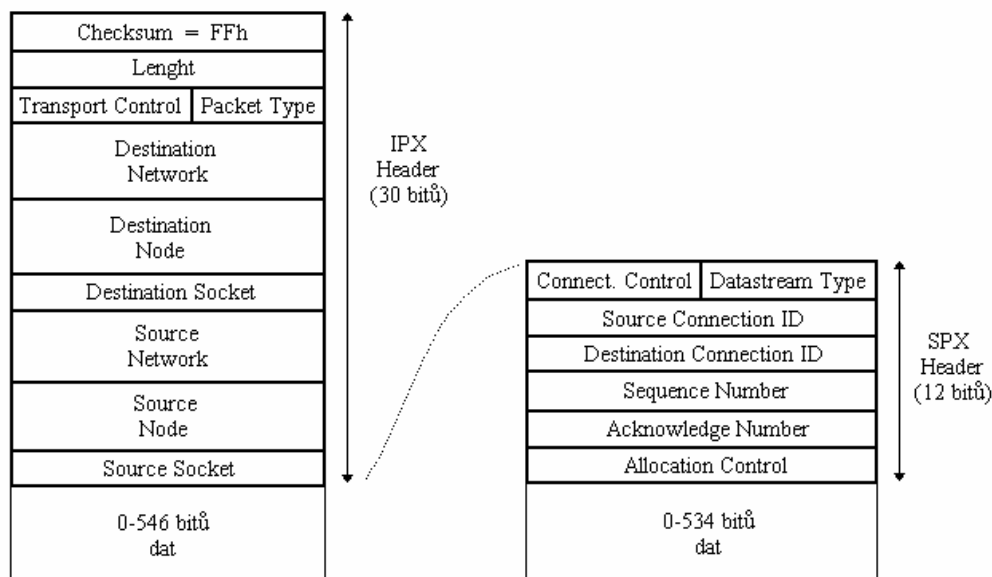


Obr. 6.1 Pakety protokolu NetBIOS

## 6.2 Protokol IPX/SPX

S protokoly *IPX/SPX* (Internet Packet eXchange/Sequential Packet eXchange) komunikuje síťový operační systém Novell Netware. Protokoly vycházejí ze systému *XNS* (Xerox Network System), který byl alternativou firmy Xerox k protokolům TCP/IP. Protokol *IPX* zajišťuje přenos paketů bez potvrzování mezi aplikacemi připojenými na zvolená připojovací místa (Socket). Protokol *SPX* je nadstavbou protokolu *IPX*, zajišťuje potvrzování přenesených paketů a umožňuje práci více aplikačních procesů na jednom portu.

Komunikační funkce *IPX/SPX* vyžadují, aby aplikace uložila potřebné parametry do požadavkového bloku *ECB* (Event Control Block), obsluha požadavků může být asynchronní k dalšímu běhu aplikace. Aplikace může na ukončení požadované funkce aktivně čekat nebo může být přerušena dokončovací rutinou. Obrázek 6.2 ukazuje strukturu paketů protokolů *IPX* a *SPX*.



Obr. 6.2 Pakety protokolů IPX a SPX

Výhodou protokolů IPX/SPX je adresace, která vychází z adresace stanic v lokální síti. Adresa je v IPX definována jako dvojice (32bitová adresa sítě, 48bitová adresa stanice), to zjednodušuje práci směrovačů ale i stanic v síti. Podstatnou nevýhodou IPX/SPX je skutečnost, že adresu sítě definuje správce konkrétní sítě. Chybějící kooperace v přidělování adres v principu znemožňuje vzájemné propojení sítí protokoly IPX/SPX mezi sebou.

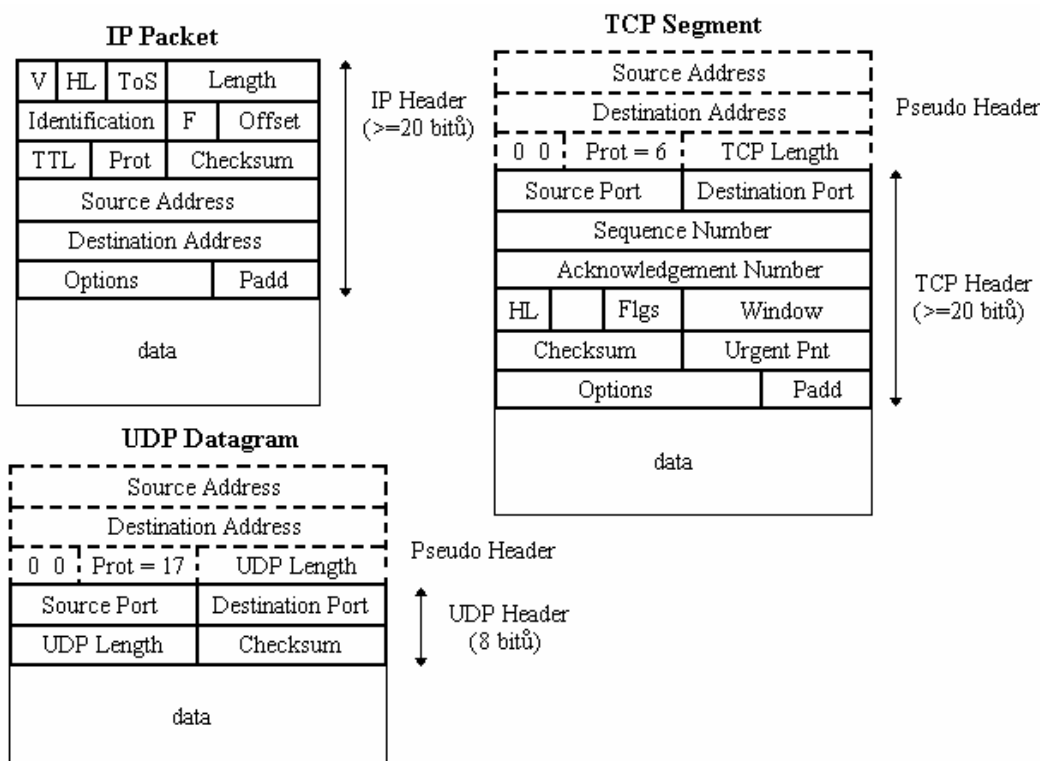
### 6.3 Protokol TCP/IP

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

Aplikační rozhraní protokolů TCP, IP a UDP jsou poměrně přesně definována v systémech UNIX jako *BSD sokety* (BSD Sockets) nebo jako rozhraní *TLI* (Transport Layer Interface). Rozhraní v systémech Windows je obdobou BSD soketů doplněné o podporu asynchronního provádění funkcí.

Funkce rozhraní zahrnují vytváření (Socket) a rušení (Close) datových struktur řídících komunikaci na daném přípojném místě (portu) nebo po virtuálním kanále, jejich vazbu na logický kanál, vazbu na adresaci informaci (Bind) a limit počtu neobsložených požadavků na vstupu (Listen). Součástí rozhraní

TCP jsou funkce pro pasivní a aktivní otevření kanálu (Accept a Connect) a pro jeho uzavření (Close). Přenos paketů a zpráv zajišťující volání funkcí Write a Read, spolu s několika formami funkcí Send a Receive. Formát IP paketů, UDP datagramů a TCP segmentů je na obrázku 6.3.



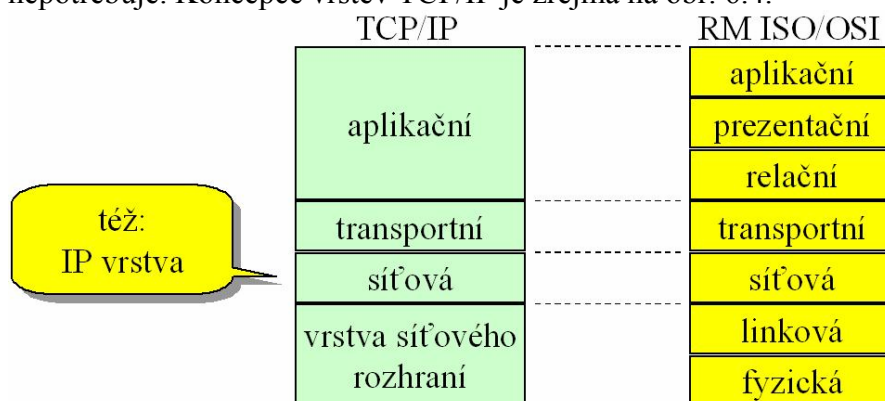
Obr. 6.3 Formáty paketů protokolů IP, TCP a UDP

TCP/IP je ve skutečnosti síťová architektura, která obsahuje ucelenou představu o počtu a úloze vrstev a obsahuje i konkrétní protokoly. Historie vzniku TCP/IP souvisí s Internetem (ARPANETem) postaveným na „prozatímní“ paketové technologii NCP (Network Control Protocol). Cílem bylo ověřit životaschopnost paketové technologie. Protože protokol NCP nebyl vhodný pro rutinní používání byl TCP/IP vyvíjen jako „definitivní“ řešení pro vznikající Internet. Filosofie, uplatněná při vzniku TCP/IP vycházela ještě z původních požadavků na ARPANET (který byl navržen pro vojáky), že nesmí mít žádnou centrální část (tu by nepřítel zničil jako první). Charakterem musí být převážně decentralizovaný, musí to být velmi robustní (tak aby to aspoň nějak fungovalo, když nepřítel část sítě odstřelí) a komunikace bude mít spíše nespojovaný charakter.

Filosofie TCP/IP je řešena tak, aby šlo snadno připojovat dříve samostatné sítě a různé síťové technologie. Požadavkem bylo, aby přenosová část hlavně přenášela data a ne se starala o další věci a spolehlivost si zajistily až koncové uzly a nikoli přenosová část sítě. Proto přenosová část (síťová vrstva) funguje pouze nespolehlivě a mechanismy zajišťující spolehlivost jsou implementovány až v transportní vrstvě (ale jako volitelná možnost - tj. není povinnost je využívat). Otázkou je, proč je výhodnější, aby si spolehlivost zajišťovaly až koncové uzly? Některé aplikace nemusí spolehlivost potřebovat, a dají přednost rychlému a pravidelnému přenosu. Spolehlivost je vždy relativní (nikoli 100%), někomu by nemusela postačovat míra „zabudované“



polehlivosti a musel by si ji zajišťovat sám a znovu a to by bylo neefektivní, protože režie by se sčítala. Protože k zajištění spolehlivosti je třeba výpočetní kapacita, a ta je lacinější v koncových uzlech než „uvnitř“ sítě autoři TCP/IP se nespolehlivých služeb nebáli. Filosofie, uplatněná při vzniku TCP/IP, tj. maximální robustnost vedla k nespojovanému charakteru tj. aby při výpadku nebylo nutné složitě rušit stávající a navazovat nová spojení. Komunikace by měla mít spíše bezstavový charakter (aby se nemusela uchovávat žádná stavová informace o dosavadním průběhu komunikace) tzn. aby nebylo nutné se složitě zotavovat z případného výpadku. Další filosofií, uplatněnou při vzniku TCP/IP je sdílení mechanismů. TCP/IP to řeší tak, aby režii nenesli ti, kdo je nechtějí používat tj. ne-sdíleně, tj. zabudovávají se přímo a pouze do těch aplikací, které je skutečně potřebují. Režii nenese ten, kdo mechanismy nepotřebuje. Koncepce vrstev TCP/IP je zřejmá na obr. 6.4.



Obr. 6.4 Koncepce TCP/IP

**Aplikační vrstva TCP/IP** má koncepci obdobnou aplikační vrstvě ISO/OSI. Jsou zde základní části aplikací, ostatní (UI) jsou nad aplikační vrstvou. Služby relačního a prezentačního charakteru jsou přímou součástí aplikací. Původní služby aplikační vrstvy jsou elektronická pošta, přenos souborů a vzdálené přihlašování. Později vznikají další, jako sdílení souborů, správa sítě, zpřístupnění informací, WWW.

**Transportní vrstva TCP/IP** řeší komunikaci koncových účastníků (end-to-end communication). Sama využívá nespojovaný a nespolehlivý přenos na úrovni síťové vrstvy. Alternativně nabízí spojovaný a spolehlivý přenos a aplikace si mohou vybrat dle vlastního uvážení. Protokol TCP (Transmission Control Protocol) zajišťuje spolehlivý a spojovaný přenos, „tváří se“ jako proud (stream), který přenáší jednotlivé byty. Protokol UDP (User Datagram Protocol) zajišťuje nespojovaný a nespolehlivý přenos a je jen „lehkou nadstavbou“ nad síťovou vrstvou, nemění povahu přenosových služeb síťové vrstvy.

**Síťová vrstva TCP/IP** zajišťuje pouze nespojovaný a nespolehlivý přenos tj. „holé minimum“, ale snaží se být co nejrychlejší. Zajišťuje jednotné přenosové služby nad všemi možnými přenosovými technologiemi nižších vrstev, vytváří „jednotnou pokličku“.

**Protokol IP** (Internet Protocol) je hlavní (a jediný) přenosový protokol, snaží se zakrývat specifika přenosových technologií nižších vrstev, a fungovat nad nimi optimálně. Jsou varianty jako SLIP (Seriál Line IP), PPP (Point-to-point protokol).

**Vrstva síťového rozhraní** (Network Interface Layer) zahrnuje „vše pod síťovou vrstvou“. TCP/IP tuto vrstvu samo nijak nenaplnuje tj. nespecifikuje svoje vlastní přenosové technologie na nejnižších vrstvách. Předpokládá se, že zde se použije to, co vznikne někde jinde (mimo rámec TCP/IP), například Ethernet, Token Ring, ATM. TCP/IP se zabývá pouze tím, jak tyto technologie co nejlépe využít, jak nad nimi provozovat IP.

Snaha překrýt odlišné přenosové technologie jednotnou „pokličkou“ naráží na problémy s adresami. Např. Ethernet používá 48-bitové adresy, ARCnet 8-bitové atd. (tj. linkové adresy jsou hodně odlišné). Protokol IP používá 32-bitové adresy (tzv. IP adresy). Protokol IP sám data fyzicky nepřenáší (ale využívá těch přenosových technologií, které jsou „pod ním“). Služby „pod“ protokolem IP mohou být dosti různorodé, zejména z hlediska:

- velikosti přenášených bloků (rámců)
- míry spolehlivosti
- přenosového zpoždění
- spojovaného či nespojovaného charakteru
- charakterem přenosu (lze dělat broadcasting?).

Protokol IP se zaměřuje na „holé minimum“. Standardy TCP/IP jsou skutečně otevřené, i když nikdo pořádně neví, co to přesně znamená. Nejsou „v rukou“ jediné firmy, vznikají (jsou přijímány) na základě všeobecného konsensu, specifikace těchto protokolů jsou veřejným vlastnictvím za jejich využití se neplatí žádné licenční poplatky, texty specifikací mají povahu volně šiřitelných dokumentů (dokumentů RFC). Standardy vznikají v rámci „sdružení“ IETF (Internet Engineering Task Force) což je dosti volné společenství odborníků, zainteresovaných na vývoji TCP/IP. Nad IETF stojí „dozorčí rada“, IESG (Internet Engineering Steering Group) která řídí práci jednotlivých skupin v rámci IETF a nad IESG stojí IAB (Internet Architecture Board), která vydává vlastní standardy. V současnosti vlastní technická řešení vznikají spíše u firem a firmy předkládají své řešení k IETF. Když je řešení uznáno za potřebné a vhodné, může být přijato jako standard Internetu a tím se standardizované řešení stává „veřejným vlastnictvím“. Každý standard má formu dokumentu RFC (Request for Comment). V současnosti existuje řádově tisíce dokumentů RFC avšak ne všechny dokumenty RFC jsou standardy!!!! Většina má povahu informačních materiálů, návodů, doporučení. Dokumenty RFC jsou volně šiřitelné a nikdy se nemění, jejich obsah zůstává vždy stejný, lze je snadno archivovat a šířit nikdy nevznikají neaktuální verze a když je potřeba nějaké řešení změnit, vydá se nový dokument RFC a ten ve své hlavičce prohlásí předchozí dokument RFC za „přežitý“ (obsolete). K jedné a téže problematice se v různých okamžicích mohou vztahovat různé dokumenty RFC. Každý dokument STD se vždy vztahuje k určité konkrétní problematice. Obsahem dokumentu STD je ten dokument RFC, který v daném okamžiku řeší příslušnou problematiku. Koncepce protokolů TCP/IP vznikla zhruba v letech 1977-8. Internet přešel na TCP/IP k 1.1.1983 a od té doby se koncepce nijak principiálně nezměnila. Další vývoj má charakter zdokonalování a obohacování, přibývají zejména nové služby jako NFS Gopher, WAIS, WWW a další. Stávající služby se obohacují o další možnosti např. el. pošta o podporu netextových formátů (standard MIME). V posledním období je TCP/IP silně kritizován a to z několika pohledů:

- internet, není bezpečný, neboli protokoly TCP/IP nezajišťují takovou míru bezpečnosti, jakou by si (někteří) uživatelé představovali. Je však

faktem, že při vzniku koncepce TCP/IP nebyl požadavek zabezpečení vznesen a autoři se soustředili hlavně na efektivnost přenosů. Malá míra bezpečnosti spočívá v tom, že přenášená data nejsou šifrována, ani jinak zabezpečena proti odposlechu a zneužití. Protokol IP nijak nešifruje to co přenáší. V době akademického Internetu malá úroveň bezpečnosti nevadila, avšak v době komerčního využití to vadí hodně! Změna by znamenala úpravu protokolu IP! Již dnes ale existují možnosti zvýšení míry zabezpečení a to šifrováním na aplikační úrovni kdy aplikace si sama zabezpečí data ještě před jejich odesláním. Jiný způsob je pomocí zabezpečených tunelů, kdy se vytváří zabezpečené virtuální kanály (tunely) a přenášená data (IP pakety) se zašifrují, vloží do normálních IP paketů a takto přenesou.

- filosofie TCP/IP nepočítá s mobilitou uživatelů tzn. Že nelze mít jednu IP adresu a cestovat s ní po světě. IP adresy jsou vázány na „geografické“ (topologické) umístění. Přímé využití mobilních komunikačních technologií je problematické např. GSM (musí se obcházet tím, že se GSM využívá stejně jako běžná telefonní síť )
- nedostatek adres, kdy autoři TCP/IP zvolili 32-bitový adresový prostor (IP adresy jsou 32-bitové). Přitom autoři pamatovali na existenci různých velkých sítí které potřebují různé počty adres a vznikly třídy A, B, C, ... Původní způsob přidělování adres z tohoto adresového prostoru nebyl příliš hospodárný. Autoři ani tak neudělali chybu, jako spíše nedocenili obrovitost svého úspěchu, neodhadli, jak ohromný zájem bude o připojování k Internetu. Problémem je hrozící vyčerpání IP adres který se začal řešit dočasnými opatřeními, která mají za úkol zpomalit úbytek adres. Tím je mechanismus CIDR (řeší i další problémy) tzv. privátní IP adresy. Zásadním koncepčním řešením, které by problém definitivně odstranilo je nalezení nové verze protokolu IP (IPnG, IP next Generation, resp. IPv6).
- charakter přenosu - přenosové protokoly TCP/IP jsou orientovány na přenos el. pošty, souborů apod. , mají „dávkový“ (nárazový) charakter, negarantují, za jak dlouho jsou data doručena, ani s jakou pravidelností budou doručovány jednotlivé části dat. To velmi vadí multimediálním přenosům, přenosům živého zvuku a obrazu. Zvyšováním přenosových kapacit lze nepříznivý efekt zmírnit, ale ne odstranit zcela. Díky tomu se stalo možné například telefonování po Internetu. Řešením jsou až specializované přenosové protokoly, podporující přenos v reálném čase RTP a RSVP

### 6.4 Srovnání TCP/IP a referenčního modelu ISO/OSI

- ISO/OSI a jeho součásti vznikají stylem „od složitějšího k jednoduššímu“, nejprve se požaduje hodně, a pak se musí ubírat, vznikají problémy s kompatibilitou „podmnožin“. K přijetí standardu není nutné ověření praktické realizovatelnosti. Naopak TCP/IP vzniká stylem „od jednoduchého ke složitějšímu“ nejprve se přijme jednodušší řešení, pak se ev. přidává. Existuje záruka kompatibility alespoň na úrovni „společného minima“, pro přijetí standardu je nutné ověření praktické realizovatelnosti dokonce i praktické provozní zkušenosti.

- Standardy ISO jsou prodávány a jsou opravdu hodně drahé. Uplatňuje se strategie: „*chci abys dodržel moje standardy, a musíš mi nejprve hodně zaplatit, abych ti vůbec řekl v čem spočívají*“ a výsledek tomu odpovídá. Naopak standardy TCP/IP (i související dokumenty) jsou dostupné volně a zdarma. Uplatňuje se strategie: „*když chci něco prosadit, musím k tomu maximálně usnadnit přístup*“ a tato strategie funguje.
- RM ISO/OSI je vhodný jako model pro studium sítí a to s vynecháním relační a prezentační vrstvy jsou protokoly „ušity na míru“ síťovému modelu. Pro praktické použití je jen velmi málo vhodný. Naopak TCP/IP je výhodný pro praktické použití, jako model pro studium sítí již není až tak výhodný, síťový model TCP/IP je „ušit na míru“ konkrétním protokolům.



### Kontrolní otázka

Jaké jsou základní koncepční odlišnosti modelu ISO/OSI a TCP/IP?



### Úkol k zamyšlení

Co považujete za nejdůležitější faktor, který vedl k tomu, že se v praxi výrazně více rozšířil model TCP/IP, než RM ISO/OSI?

## 7 TECHNICKÉ A PROGRAMOVÉ VYBAVENÍ SÍTÍ PC-LAN

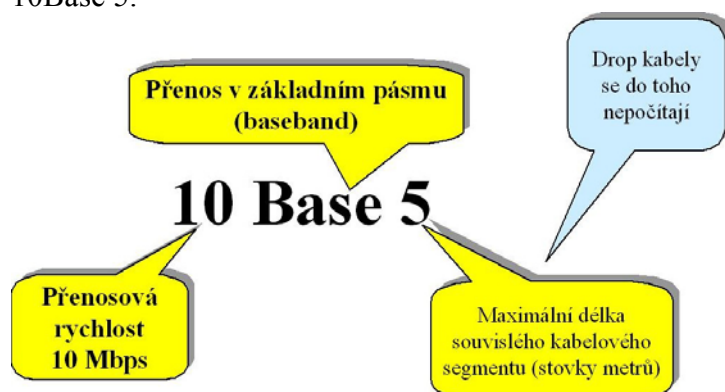
K zapojení počítače do místní sítě musíme mít k dispozici následující technické a programové komponenty :

- přenosové médium (kabel)
- síťovou kartu tzv. adaptér
- obslužný program pro adaptér tzv. driver
- síťový operační systém

Většina výrobců se snaží o vzájemnou síťovou kompatibilitu. Z tohoto důvodu dochází ke snaze vytvořit síť pokud možno modulárně, aby byla možná zaměnitelnost jednotlivých komponent více výrobců.

### 7.1 TECHNOLOGIE POČÍTAČOVÝCH SÍTÍ

Dříve, než objasníme jednotlivé technologie počítačových sítí, vysvětleme si způsob označování technologií. Na Obr. 7.1 je příklad označení standardu 10Base 5.



Obr. 7.1 Příklad standardního označování technologie

Sítě typu Ethernet se obecně dělí na:

„Klasický“- „desítkový“ Ethernet

- |           |  |
|-----------|--|
| 10Base-5  | Tlustý koaxiál – max. 500 m – nepoužívá se |
| 10Base-2  | Tenký koaxiál – max. 185 m                 |
| 10Base-T  | Kroucená dvojlinka – max. 100 m            |
| 10Base-FL | Mnohavidové optické kabely – max. 2 km     |

Fast - „stovkový“ Ethernet

- |            |   |
|------------|---|
| 100Base-TX | Kroucená dvojlinka – max. 100 m                   |
| 100Base-T4 | Nižší kategorie kroucené dvojlinky – nepoužívá se |
| 100Base-FX | Mnohavidové optické kabely – max 2 km             |

Gigabit Ethernet

- |             |                                       |
|-------------|---------------------------------------|
| 1000Base-SX | Mnohavidové optické kabely – až 550 m |
| 1000Base-LX | Jednovídné optické kabely – až 5 km   |
| 1000Base-CX | Stíněné kabely                        |

### 7.1.1 Technologie sítě Ethernet (IEEE 802.3)

Základy technologie, známé jako Ethernet, byly položeny začátkem 70. let. V roce 1980 byl standardizován konsorciem DEC, Intel a Xerox, standard je známý pod zkratkou DIX. Standard byl později rozšiřován o další média a nové způsoby provozu. Ethernet je přenosovou technologií zajišťující skutečný přenos dat v RM ISO/OSI a pokrývá fyzickou a linkovou vrstvu. V rámci TCP/IP spadá do vrstvy síťového rozhraní. Přitom může používat různá přenosová média jako koaxiální kabely, kroucenou dvoulinku, optická vlákna. Předpokládá logicky sběrníkovou topologii tj. má „sdílenou“ povahu. Teprve v poslední době se díky switchingu mění na „nesdílenou“ přenosovou technologii. Chování je „statistické“ tj. nezaručuje právo vysílat a funguje dobře s „rozumnou“ pravděpodobností. Dále se vyvíjel v stomegabitový a gigabitový Ethernet. Existuje více standardů jako například : *10BASE5*, *10BASE2*, *10BROAD36*, *StarLAN*, *10BASE-T*, *100BASE-TX* a *100BASE-FX*. V současné době, však mezi nejrozšířenější patří : *10BASE-T* a *100BASE-TX*.

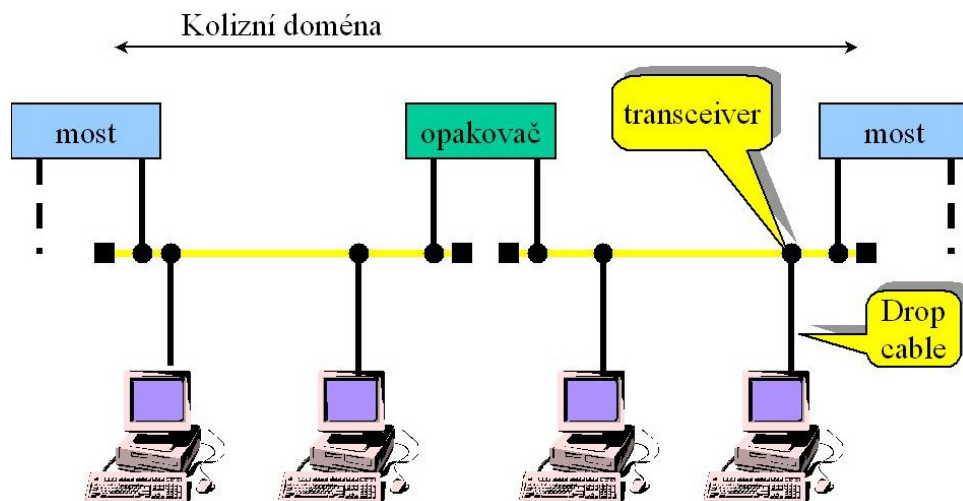
### 7.1.2 Technologie 10BASE 5

Nejstarší verze Ethernetu počítala s tzv. tlustým (žlutým) koaxiálním kabelem o průměru cca 1 cm, z něj se dělaly odbočky k jednotlivých uzlům, pomocí tzv. drop kabelů. Koaxiální kabel se buď rozpojil a znovu spojil přes tzv. transceiver, nebo byl „nabodnut“ zvláštním nožovým konektorem (tzv. vampire tap) – viz obr.7.2.



Obr. 7.2 Představa transceiveru

Rozhraní AUI = Attachment Unit Interface je rozhraním mezi transceiverem a ostatními obvody síťového adaptéru. Jde o rozhraní „na obou koncích drop kabelu“, používá 15-pinový konektor (Canon). Používá se i dnes a jsou jím vybavovány i takové síťové karty, které mají zabudovaný transceiver např. pro tenký koax. Kabel. Umožňuje to připojit ke kartě i jiné druhy transceiverů, např. pro optická vlákna. Představa topologie je uvedena na obr. 7.3



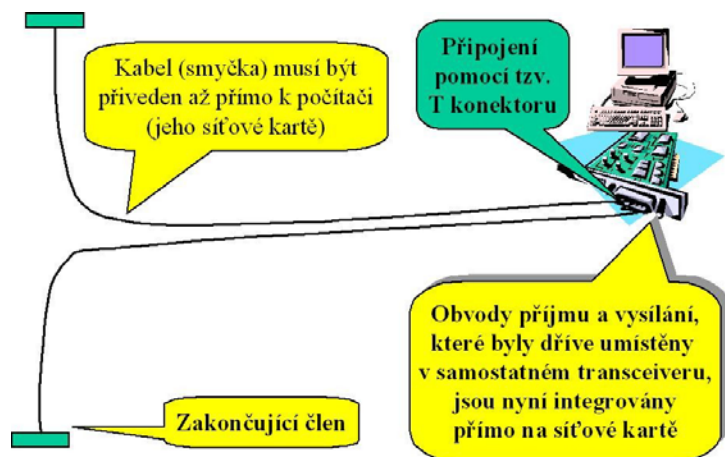
Obr. 7.3 Topologie tlustého koaxu

## 7.1.3 Technologie 10BASE2

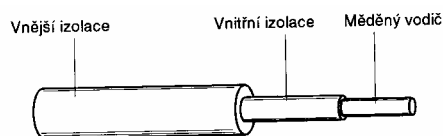
Tlustý koaxiální kabel byl drahý, málo ohebný, špatně se instaloval, Topologie rozvodů na bázi tlustého koaxiálního kabelu byla vcelku vhodná pro páteřní sítě, ale méně již pro připojování stanic. Místo tlustého koaxiálního kabelu se přešlo na tenký koaxiální kabel (průměru cca 0,5 cm), v provedení:

- s jednoduchým opletením
- s dvojitým opletením

Tenký koaxiální kabel je lacinější, ohebnější. Možnost jeho využití si vyžádala úpravu standardu, resp. nový standard 10Base2 který je odlišný hlavně na úrovni fyzické vrstvy. 10Base2 předpokládá max. délku kabelového segmentu 185 m (zaokrouhleno 2x100m).



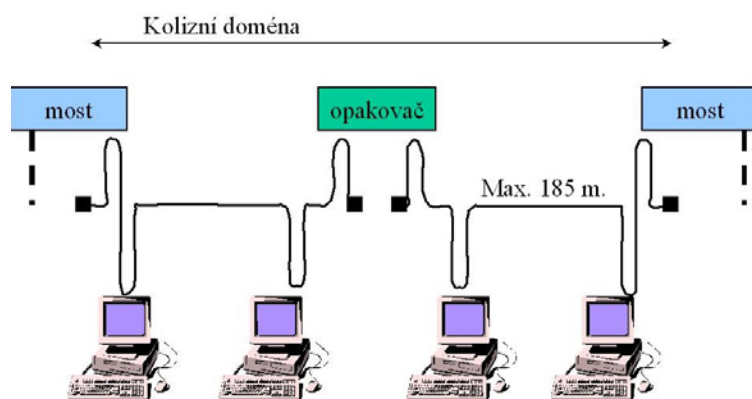
Obr. 7.4 Připojování k rozvodům z tenkého koaxiálního kabelu



Obr. 7.5 Tenký koaxiální kabel

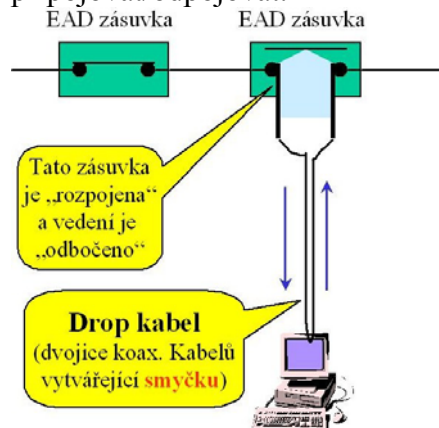


Obr. 7.6 Konektory tenkého koaxu - 10Base2



Obr. 7.7 Představa topologie 10Base2

Systém EAD (Ethernet Attachment Device) je konkrétní variantou (provedením) rozvodů na tenkém koaxiálním kabelu, umožňuje budovat „modulární“ rozvody takové, ke kterým se lze dynamicky připojovat/odpojovat.



Obr. 7.8 Připojování stanice k tenkému koaxiálu pomocí EAD zásuvky

### 7.1.4 Technologie 10BASE-T

Další vývoj Ethernetu byl podmíněn snahou o využití již existující rozvodů „telefonního typu“. Nejprve vzniknul standard 1Base5, umožňující dosáhnout až na 500 metrů, ale jen s rychlostí 1 Mbps!!! Dalším standardem se stal 10BaseT s rychlostí 10 Mbps a dosahem kabelu 100 m. Telefonní kabely, které



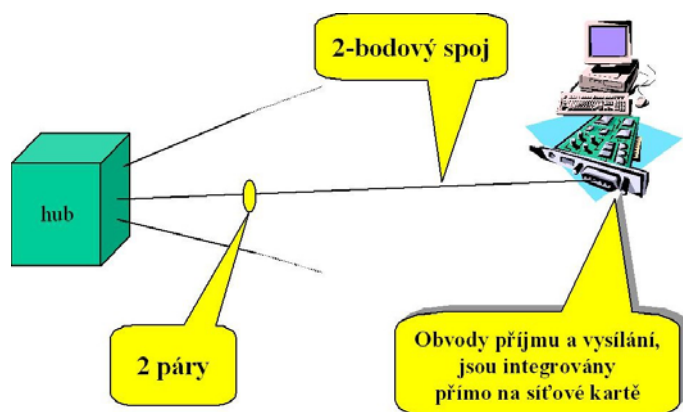
## Počítačové sítě

předpokládá standard 10BaseT, jsou tzv. kroucenou dvoulinkou (též: twist, UTP), na každý uzel jsou zapotřebí 2 páry a kvalita je tzv. voice grade („hlasové“) - dnes se říká kategorie 3.



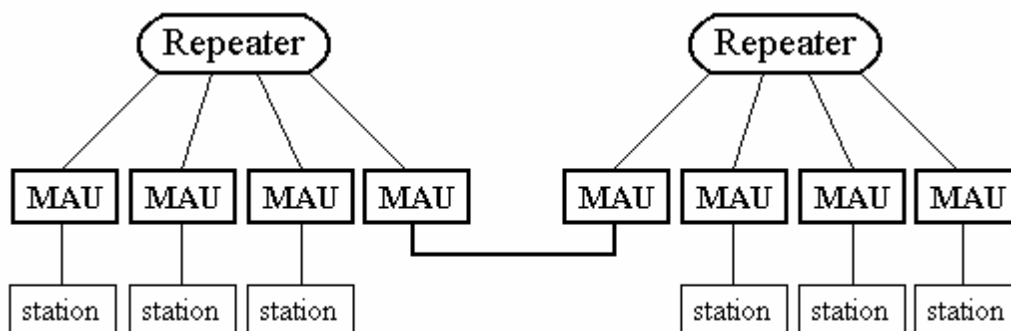
Obr. 7.9 Kabel UTP

Na kroucené dvoulinkě nelze dělat odbočky. Případné rozvětvení (rozbočení) je nutné dělat elektronickou cestou a kvůli tomu se používají rozbočovače (huby). Rozbočení (rozvětvení) může logicky fungovat na úrovni fyzické vrstvy, pak se hub chová jako opakovatel a nebo na úrovni linkové vrstvy, pak se hub chová jako most, v. switch.



Obr. 7.10 Připojování k rozvodům z kroucené dvoulinky (10BaseT)

Na obr.7.11 jsou naznačeny úseky UTP kabelu o délce do 100 m (přesněji do 90 m pevného rozvodu a dvakrát 5 m pohyblivý kabel pro připojení zařízení) propojují jednotlivé stanice s více-vstupovým opakovatelem (Multiport Repeater, koncentrátor). Ten je středem hvězdice tvořené skupinou až 8, 12, 16 nebo i více stanic.



Obr.7.11: Struktura sítě 10BASE-T

Ze čtyř vodičových párů kabelu UTP jsou využity dva, jeden pár přenáší signál od stanice k opakovateli, druhý přenáší signál ve směru opačném. Kabel UTP

musí splňovat podmínky na šířku pásma, charakteristickou impedanci a přeslech. Podmínky splňují kabely UTP Cat. 3 a s rezervou dnes běžnější UTP Cat. 5. Jako konektor slouží plochý konektor EIA RJ-45, který je 8-pólový a využity jsou jen 4 piny, kde:

pin č. 1: TransmitData+

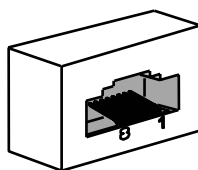
pin č. 2: TD-

pin č. 3: Receive Data+

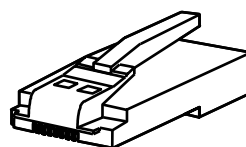
pin č. 6: RD-

ostatní: nevyužité

### zásuvka



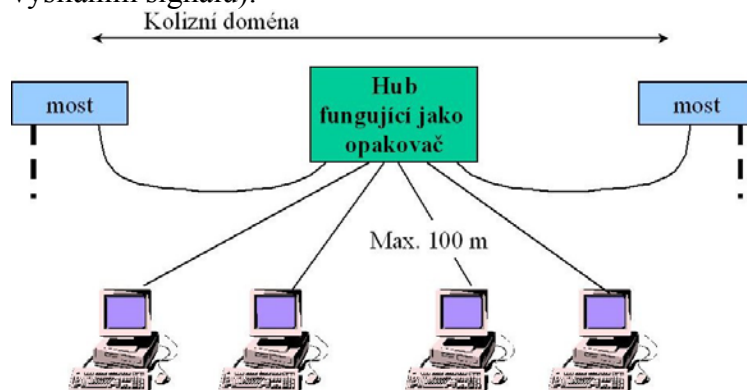
### zástrčka



Obr 7.12. Tvar zásuvky a zástrčky RJ 45

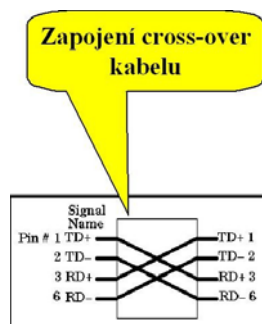
Opakovač předává signál přijatý od jedné ze stanic po elektrické úpravě ostatním stanicím, kromě stanice nebo opakovače, od nichž je přijímán. Stará se tak o vytvoření sdíleného kanálu. Příjem signálu při vlastním vysílání je pro stanici indikací kolize. Opakovače lze mezi sebou propojovat, buď opět kabely UTP, koaxiálním kabelem nebo optickými spoji. Řada výrobců nabízí vedle opakovačů s pevným počtem rozhraní i opakovače *modulární* (pro 8, 12, 16 UTP vedení a s moduly pro jiná média - koaxiální kabel, FOIRL nebo s univerzálním rozhraním AUI) a stohovatelné huby (Stackable Hub).

Sdílený kanál vytvářený více-vstupovým opakovačem 10BASE-T nebo strukturou z nich složenou a přináší proti sběrníkovému propojení počítačů podstatnou výhodu: odpojení stanice nemůže ovlivnit chod zbytku sítě. Logika moderních opakovačů 10BASE-T dovolí odizolovat i stanici, která by u sběrníkovému Ethernetu svou poruchou narušila funkci celé sítě (např. trvalým vysíláním signálu).



Obr. 7.13 Představa topologie (10BaseT)

Pro propojení hub-uzel je třeba tzv. patch kabel zapojený jako kabel 1:1 kde jsou vzájemně propojeny piny stejných čísel. V singulárních případech lze přímo propojit i dva koncové uzly mezi sebou, tj. bez použití hubu, ale je na to potřeba tzv. cross-over kabel



Obr. 7.14 Zapojení cross-over kabelu

### Zapojení kabelu UTP na RJ45

#### Standardní

- 1 – Oranžovo-bílý
- 2 – Oranžový
- 3 – Zeleno-bílý
- 4 – Modrý
- 5 – Modro-bílý
- 6 – Zelený
- 7 – Hnědo-bílý
- 8 – Hnědý

#### Překřížené (cross-over)

- 1 – Zeleno-bílý
- 2 – Zelený
- 3 – Oranžovo-bílý
- 4 – Modrý
- 5 – Modro-bílý
- 6 – Oranžový
- 7 – Hnědo-bílý
- 8 – Hnědý

### 7.1.5 Technologie 100BASE-TX

Výraznou technologickou modifikací hvězdicového Ethernetu *10BASE-T* je standard označovaný jako *100BASE-T* a zvyšující přenos na 100 Mb/s na kabelovém rozvodu UTP Cat. 5 nebo kabelech STP či optických vláknech.

Standard 100BaseT je označován jako Fast Ethernet a oproti 10Mbps verzi je 10x rychlejší a je zaveden mechanismus pro detekci rychlosti tzv. auto-negotiation of media speed. Beze změny zůstal formát linkových rámců, přístupová metoda a adresy. Fast Ethernet (100BaseT) dosáhl 10x násobného zrychlení 10x násobným zkrácením bitového intervalu a zkrácením maximálního dosahu kabelových segmentů. Zavedla se možnost používání různých druhů kabeláže tj. dvoulinky kategorie 5, dvoulinky kategorie 3 a optických vláken. Fyzická vrstva Ethernetu se rozdělila na dvě podvrstvy a to Medium Independent Interface (MII) a Physical Layer Device (PHY). Vrstva PHY (Physical Layer) nabízí 3 varianty pro pro různé druhy kabeláže tj. konkrétní řešení (standards) pro použití kroucené dvoulinky kategorie 5 a optických vláken. Pak 100 Base TX říká jak provozovat 100Mbps Ethernet nad 2 páry dvoulinky kategorie 5, 100 Base FX pro optická vlákna a 100 Base T4 říká jak provozovat 100 Mbps Ethernet nad 4 páry dvoulinky kategorie 3 („telefonní“).

Technologie rychlého Ethernetu je založena na efektivnějším využití přenosového média. Kódování *Manchester* je nahrazeno efektivnějším kódováním *4B5B*. To dovolí dosáhnout přenosové rychlosti 100 Mb/s (na médiu až 125 Mb/s). Vzdálenost mezi stanicí a koncentrátorem je do 100 m, optické vlákno dovolí jít až na 400 m.

## Počítačové síť

Rychlý Ethernet definuje 3 rozdílné realizace fyzického kanálu. Základem jsou kanály 100BASE-TX - 2 páry kabelu UTP nebo STP a 100BASE-FX - dvojice optických vláken.

Zvýšení rychlosti při zachování ostatních vlastností Ethernetu si však vyžádalo snížení maximální vzdálenosti. Pokud jde o vícevstupové opakovací, rychlý Ethernet definuje 2 odlišné typy :

**Class 1** - umožňuje použití různých fyzických rozhraní na vstupech a smí být mezi stanicemi jediný. Class 1 „dekóduje“ jednotlivé bity, překládá mezi různými druhy kódování (různými médii) a umožňuje přechod mezi různými přenosovými médii, např. 100BaseTX a FX v kolizní doméně smí být jen 1 protože je pomalý, způsobuje zpoždění signálu v délce několika bitů.

**Class 2** - pracuje se stejnými fyzickými rozhraními, mezi stanicemi smí být nejvýše 2 opakovací tohoto typu, navzájem propojené na vzdálenost 5 m. Class 2 „nedekóduje“ jednotlivé bity, pouze „vyhlazuje“ signál, nesnaží se jej interpretovat ani na úrovni bitů, dokáže propojit jen segmenty se stejným způsobem kódování TX a FX, nikoli T4. V kolizní doméně smí být až 2 tyto opakovací.

Omezení dosahu 100BaseT je dáno obecnými zásadami tj.:

- žádný segment z kroucené dvoulinky nesmí být delší než 100 metrů
- žádný optický segment nesmí být delší než 412 metrů
- drop kabely (MII kabely, mezi transceiverem a kartou) nesmí být delší než 0,5 m

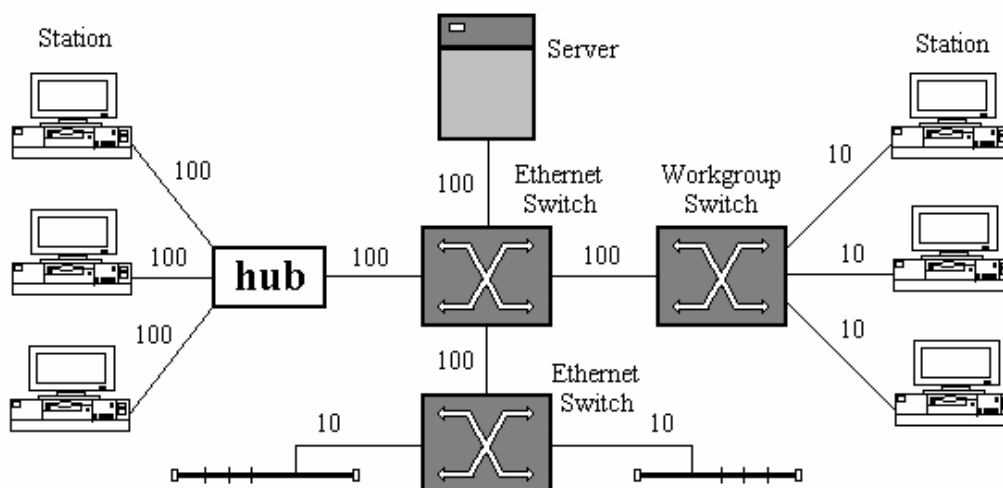
a dalším omezením uplatňujícím se při použití opakovaců

- nelze mechanicky počítat délky segmentů
- konkrétní hodnoty uvádí následující tabulka
- Max. velikost kolizní domény

POUŽITÝ OPAKOVAČ	TWIST	OPTICKÉ VLÁKNO	T4 + FX	TX+FX
Žádný	100m	412m	N/A	N/A
1x Class I	200m	272m	231m	260m
1x Class II	200m	320m	N/A	308m
2x Class II	205m	228m	N/A	216m

Pro řadu aplikací může stačit dvoubodové připojení pracovišť kanály o rychlosti 10 Mb/s (mikrosegmentace) k přepínači, na který jsou rychlými kanály připojeny servery a další části sítě. Pro náročné aplikace je k dispozici možnost sdílení rychlého kanálu 100 Mb/s, nebo mikrosegmentace s plným vyhrazením kanálů 100 Mb/s. Příklad možné topologie sítě na technologii 100BASE-TX uvádí obrázek 7.15.

## Počítačové sítě



Obr. 7.15: Topologie sítě 100BASE-TX

Kombinace zařízení se standardní rychlostí 10 Mb/s a zařízení pracujících se 100 Mb/s a navíc s odlišným využitím média (100BASE-TX a 100BASE-T4) a režimem provozu (poloduplex, plný-duplex) může přinést problémy se správou a konfigurací. Pro usnadnění konfigurace jsou zařízení umožňující práci oběma rychlostmi vybavena obvody dovolujícími automatickou konfiguraci při zahájení provozu. Mechanismus respektuje i fakt, že jedno ze zařízení nemusí být obvody pro automatickou konfiguraci vybaveno.

## 7.2 Technické vybavení sítí PC-LAN

Technické vybavení sítí je tvořené přenosovým médiem, síťovým adaptérem případně opakovači a mosty. V standardním případě je technickým vybavením realizovaná úroveň první a část druhé vrstvy. Tyto vrstvy vlastně určují topologii a přístupovou metodu sítě.

### 7.2.1 PŘENOSOVÉ MÉDIUM

Zajišťuje přenos dat mezi počítači sítě. Nejčastěji se používají tyto druhy přenosových medií :

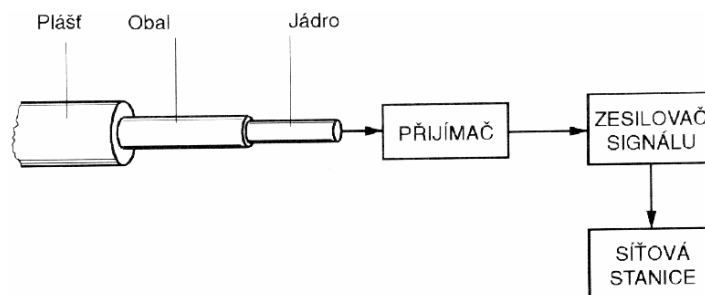
- **symetrický kabel** (kroucená dvojlinka) - hlavně u levných sítí s přenosovou rychlostí do 2Mbit/s (token ring)
- **stíněný symetrický kabel STP** - složený z měděných vodičů, které jsou obklopené izolačním nevodivým materiálem, použití je podobné jako u předchozího typu (Type 1 - Type 9)
- **nestíněný symetrický kabel UTP** - složený s dvou, čtyř, 22, 24 nebo 26 vodičů, kde páry jsou navzájem obtočené, použití je podle kategorie (Cat. 3 - Cat. 5)

Členění:

- kategorie 1: telefonní pro přenos řeči

## Počítačové sítě

- kat. 2: do 4Mb/s - 4 páry
- kat. 3: do 16 Mb/s - 4 páry s 9 závity/1m
- kat. 4: do 20 Mb/s
- kat. 5: do 100 Mb/s
- kat. 6: do 200 Mb/s
- vedení až 100m do rozbočovače
- 100BaseFX – vlákno Multimode – 400 m
- **stíněný kabel STP**
- **koaxiální kabel** - používá pro svou dobrou odolnost vůči rušení a vysokou přenosovou rychlost až do 10 Mb/s
- **optický kabel** - pro své výborné vlastnosti je čím dál více perspektivnější, hlavně při propojování vzdálenějších sítí, je však stále poměrně drahý



Obr. 7.16 Připojení optického kabelu

- **rádiové spojení** - není tolik rozšířené. Síťové karty obsahují vysílače a přijímače vř signálu o kmitočtu 900 MHz. Přenosová rychlost je 1 Mb/s

### 7.2.2 SÍŤOVÁ KARTA - ADAPTÉR

Zajišťuje realizaci 1. a částečně 2. vrstvy modelu ISO. Sběrnice karty je připojená na vnitřní sběrnici počítače, buď **ISA** (16 bitová - pomalá) nebo **PCI** (32 bitová - rychlejší).

#### Síťová karta zajišťuje tyto základní činnosti :

- ♦ převod vysílaných bitů do kódu přenášeném na vedení
- ♦ určuje topologii sítě a přístupovou metodu
- ♦ zajišťuje synchronizaci vzorkovacích impulsů podle přijímaného signálu fázovým závěsem
- ♦ provádí kontrolu správnosti přijatých paketů cyklickým kódem

#### Na síťové kartě se obvykle též nachází :

- paměť RAM - která slouží jako vyrovnávací paměť pro přijímané pakety obvykle má velikost 16-32 kB
- paměť PROM - anebo jen místo pro ni, tato paměť umožní stanici bez diskových mechanik natáhnout systém ze serveru (tzv. BOOT ROM)

- paměť EPROM - tato paměť je k dispozici jen u některých karet přičemž obsahuje vlastní operační systém sítě (NETBIOS)

### 7.3 Programové vybavení sítí PC-LAN

Z vrstevové struktury komunikace v síti vyplývá, že vrstvu 2 realizuje driver adaptéru a vyšší vrstvy jsou už plně pokryté operačním systémem. Programové vybavení sítě je tedy tvořené :

- 1) Driverem adaptéru
- 2) Síťovým operačním systémem

#### 7.3.1 DRIVER ADAPTÉRU

Jedná se o obslužný program, který inicializuje adaptér přičemž zajišťuje větší část druhé vrstvy. Tvoří obálku paketů a zajišťuje přístupovou metodu. Driver adaptéru zajišťuje taky přechod mezi hardwarem sítě a sítovým operačním systémem.

#### 7.3.2 SÍŤOVÝ OPERAČNÍ SYSTÉM

Pod tímto pojmem rozumíme programové vybavení, které zajišťuje dvě základní funkce :

- komunikaci jednotlivých uzlů sítě
- vytvoření a funkci serveru(ů)

Jedná se o programy, které zajišťují činnost sítě při styku uživatelů se sítí. Obvykle se liší sítový operační systém na stanici se sítovým operačním systémem na serveru.

### 7.4 Propojení sítí PC - LAN s ostatními sítěmi

Rozvoj počítačových sítí přinesl i potřebu jejich vzájemného propojení za účelem výměny dat. Snahou je, aby uživatel sítě měl přístup nejen ke své místní síti, ale i k dalším připojeným sítím LAN, případně byl propojený na veřejnou datovou síť.

Na propojení sítí existují různé technické prostředky, které členíme podle vrstvy modelu ISO v které zajišťují vlastní propojení.

## Počítačové sítě

	VRSTVA	Z pohledu „SLUŽBY“	Z pohledu „ÚČELU ČINNOSTI“	Z pohledu „DAT“
7.	<b>APLIKAČNÍ</b>	Zprostředkovává přístup ke komunikačním službám (elektronická pošta, přenos souborů).	Uživatelsky orientované služby	ZPRÁVA
6.	<b>PREZENTAČNÍ</b>	Transformuje data (formátuje, konvertuje) do/z tvaru srozumitelného aplikačnímu procesu.		
5.	<b>RELAČNÍ</b>	Navazuje, udržuje a ukončuje spojení a konverzaci koncových procesů.		
4.	<b>TRANSPORTNÍ</b>	Zajišťuje spolehlivý přenos zpráv mezi procesy ve vysílacím počítači a procesem v přijímacím počítači.	Přenosové služby	PAKET
3.	<b>SÍŤOVÁ</b>	Řídí směrování paketů v počítačové síti, její efektivní využívání.		
2.	<b>SPOJOVÁ</b>	Určuje rozdělení bitů do bloků (rámců), řídí tok rámců (aby pomalý přijímač nebyl „zahlcen“ rychlým vysílačem), provádí detekci chyb přenosu.		RÁMEC
1.	<b>FYZICKÁ</b>	Přenos nestrukturovaných dat přenosovým médiem.		BITY

Obr. 7.17 Pohledy propojování sítí

Vrstvy RM OSI	Propojovací zařízení
Aplikační	Brána (Gateway)
Prezentační	
Relační	
Transportní	
Síťová	Směrovač (Router)
Linková	Most, prepínač (Bridge, switch)
Fyzická	Opakovač (Repeater)

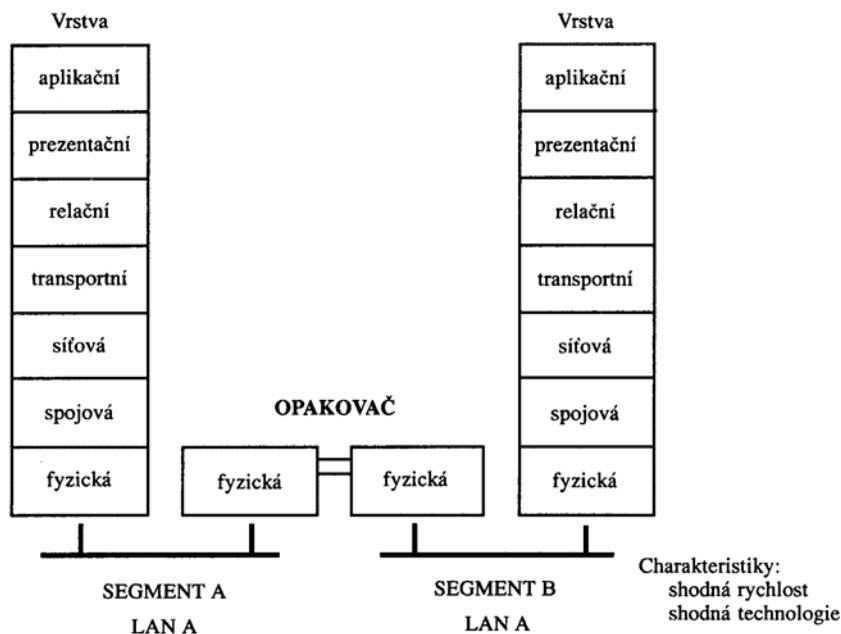
Obr. 7.18 Technické prostředky propojení počítačových sítí

### 7.4.1 Opakovač (Repeater)

Používá se na spojení vzdálených úseků sítě tzv. *segmentů*, anebo když chceme propojit navzájem stejné sítě LAN. Jeho úlohou je překlenout zvýšenou vzdálenost mezi účastníky sítě. Pracuje ve fyzické vrstvě, a proto jen regeneruje a zesiluje datový signál (jednotlivé bity).



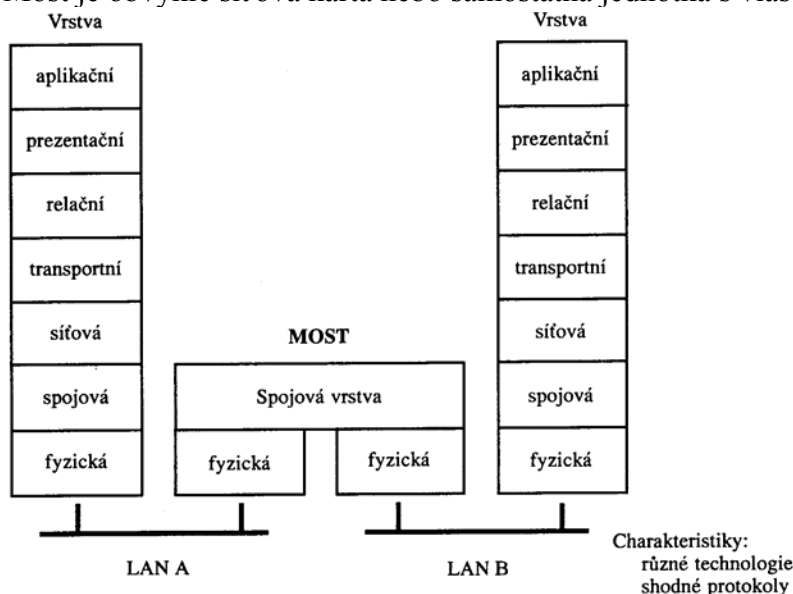
## Počítačové sítě



Obr. 7.19 Začlenění opakovače

### 7.4.2 Přepínač (Switch), most (Bridge)

Jeho úlohou je spojit dvě, anebo více sítí PC-LAN. Většinou se jedná o sítě se stejnou strukturou paketů v linkové vrstvě, tedy buď Ethernet nebo Token Ring. Takovéto mosty se nazývají homogenní. Mosty mohou být místní, anebo vzdálené, podle toho jestli jsou spojované sítě místní nebo vzdálené. Most je obvykle síťová karta nebo samostatná jednotka s vlastním napájením.

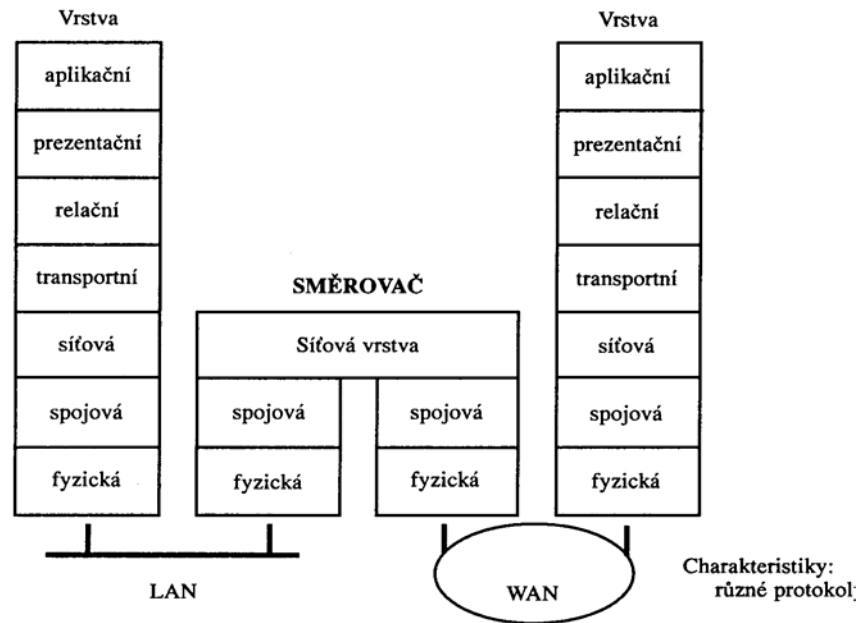


Obr. 7.20 Začlenění přepínače a mostu

### 7.4.3 Směrovač (Router)

Spojuje sítě v síťové vrstvě. Jednotlivé pakety se předávají mezi jednotlivými uzly sítě, což jsou vlastně *Routery*.

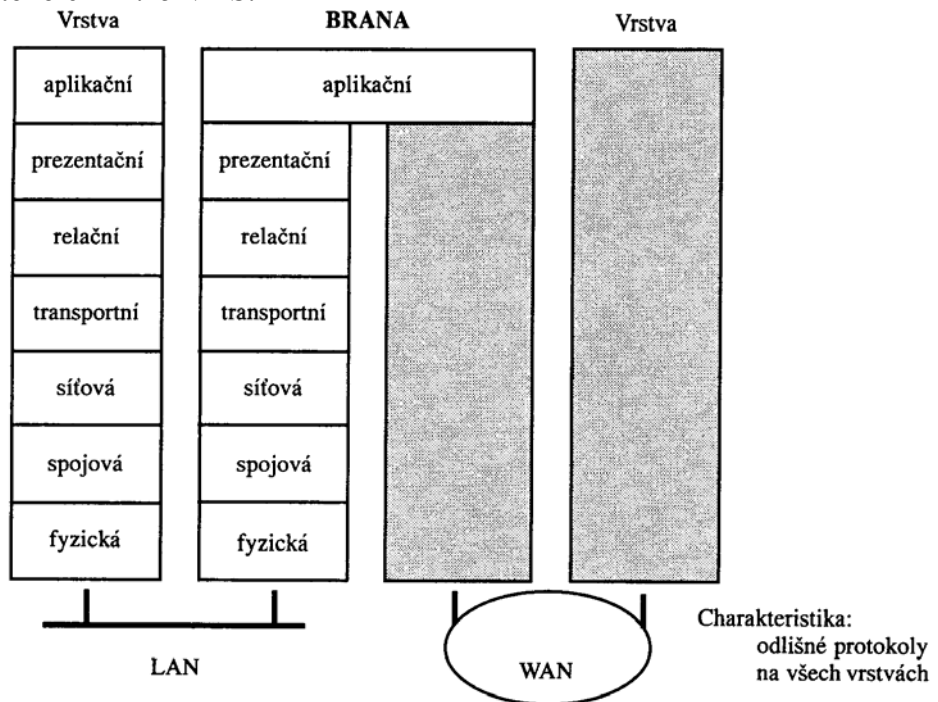
## Počítačové sítě



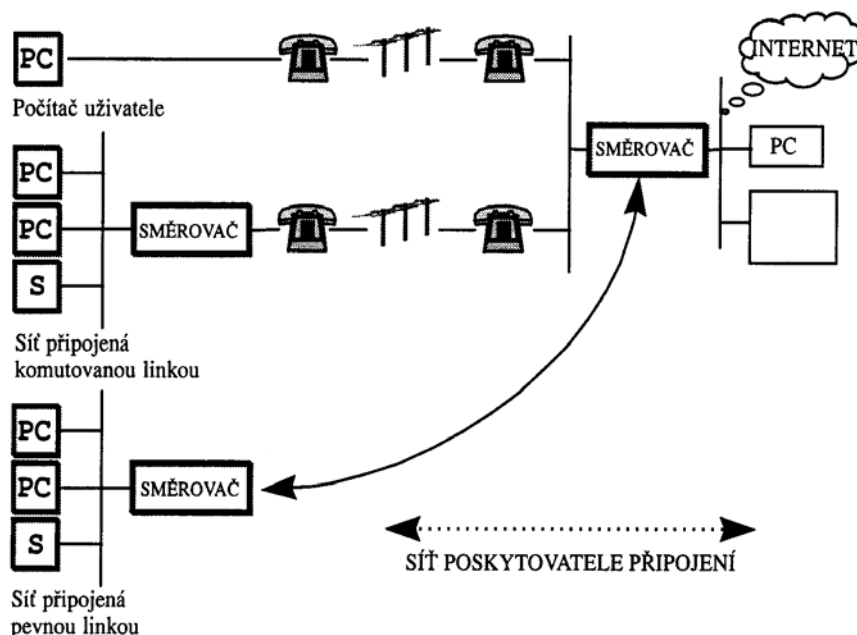
Obr. 7.21 Začlenění směrovače

### 7.4.4 Brána (Gateway)

Propojuje sítě s odlišnými operačními systémy, např. se síťovým protokolem X.25 VDS.



Obr. 7.22 Začlenění brány



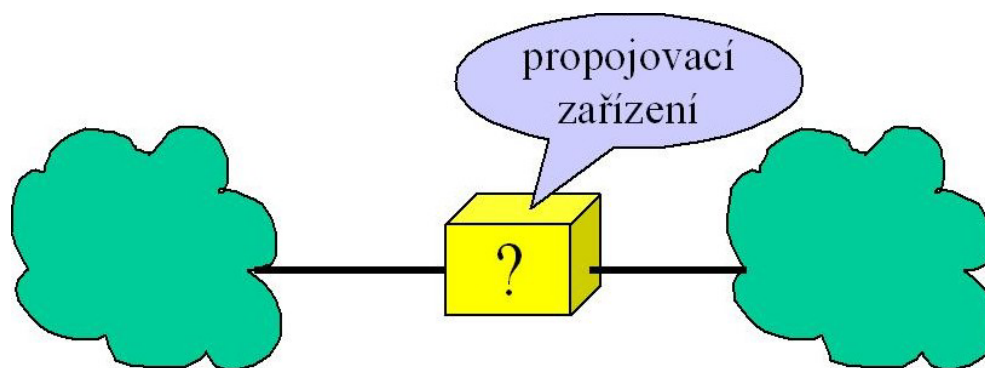
Obr 7.23 Začlenění směrovačů

## 7.5 Internetworking

Pro vzájemné propojování celých sítí i jednotlivých kabelových segmentů vznikl pojem **internetwork**, zkráceně internet a to s malým „i“, což chápeme obecně jakékoli propojení dvou či více částí a s velkým „I“ se jedná o jednu konkrétní síť („toho“ celosvětového Internetu).

Důvodem vzniku internetworkingu bylo překonání technických omezení/překážek, např. dosah kabelových segmentů je omezený (10Base2: 185 metrů), omezený je i počet uzlů které lze připojit ke kabelu. Optimalizací fungování sítě byla snaha regulovat tok dat, zamezení zbytečného šíření provozu a implementace nejrůznějších strategií a opatření (oprávnění k přístupu, správné směrování, peering, .....). Dalším důvodem pro internetworking byly fyzikální podstaty některých druhů kabeláže, hlavně kroucené dvoulinky a optických vláken, které lze použít jen jako dvoubodové spoje, mnohdy pouze jednocestné, nelze na nich dělat odbočky, „rozbočení“ musí být realizováno elektronickou cestou, prostřednictvím propojovacích prvků. Proto internetworking měl zpřístupnit vzdálené zdroje např. přístup ke vzdáleným FTP archivům, serverům Gopher, WWW serverům a to s využitím výpočetní kapacity vzdálených uzlů (vzdálené přihlašování). Tím se zvětší dosah poskytovaných služeb, užitná hodnota některých služeb je tím větší, čím větší je její potenciální dosah (např. elektronická pošta, Internetové telefonování, služby pro skupinovou diskusi, ...).

Obecná podstata internetworking-u je zřejmá z obr. 7.24

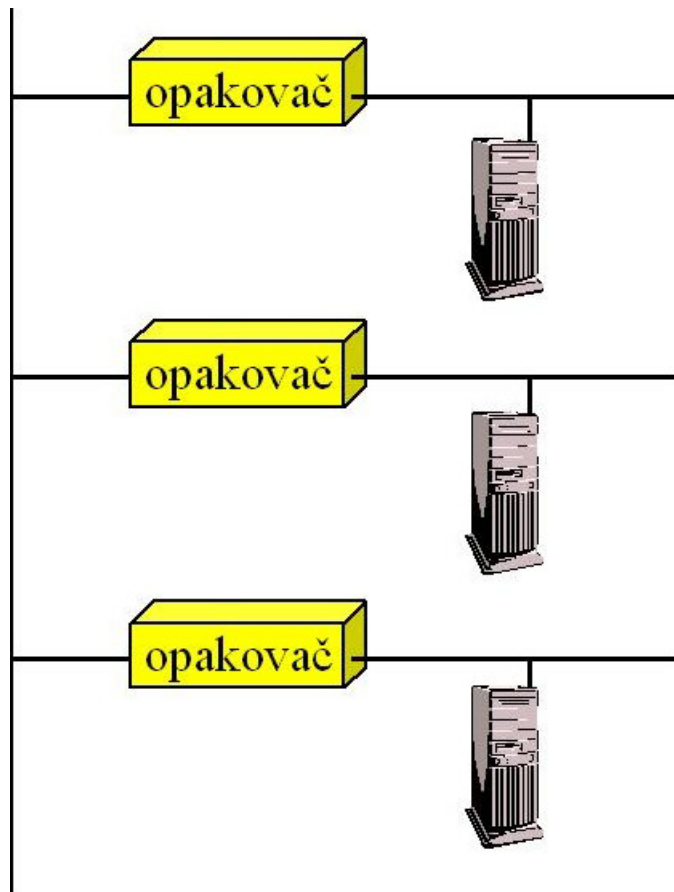


Obr. 7.24 Podstata internetworkingu

Rozdíl při různých metodách propojování je v tom, jakým způsobem pracuje propojovací zařízení. Může pracovat na úrovni fyzické až aplikační vrstvy a podle toho se také propojovací zařízení pojmenovává. V případě, že propojovací zařízení pracuje na fyzické vrstvě, hovoříme o opakovací (repeater), na linkové vrstvě o mostu (bridge), nebo přepínači (switch), na síťové vrstvě o směrovači (router) a aplikační vrstvě o bráně (gateway). Nejobecnějším aktivně fungujícím propojovacím zařízením je rozbočovač (hub). Bez apriorního určení úrovně na které pracuje, hub může fungovat jako opakováč, jako most i jako směrovač.

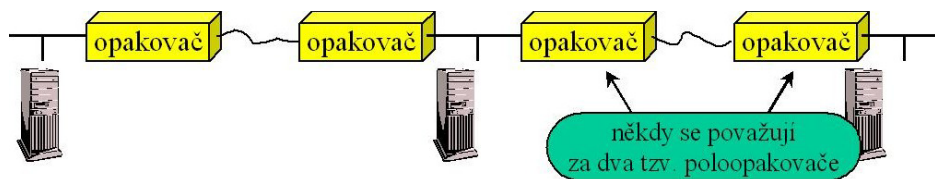
Při propojení na úrovni fyzické vrstvy je zřejmé, že propojovací zařízení (tzv. opakováč) si všímá pouze jednotlivých bitů tj. toho, co je přenášeno na úrovni fyzické vrstvy. Opakováč je pouze digitální zesilovač, který zesiluje a znovu tvaruje přenášený signál, kompenzuje zkreslení, útlum a další vlivy reálných obvodových vlastností přenosových cest. Opakováč „nevnímá“, že určité skupiny bitů patří k sobě a tvoří přenosový rámec, nedokáže rozpoznat ani adresu odesílatele a příjemce dat (rámce) a nemá k dispozici informace, které by mu umožnily měnit svoje chování podle toho, jaká data skrz něj prochází. Opakováč rozesílá („opakuje“) všechna data do všech stran (segmentů), ke kterým je připojen ale neví, co by mohl zastavit a nemusel šířit dál. Funguje v reálném čase až na malé zpoždění na svých vnitřních obvodech. Nemá žádnou vnitřní paměť pro bufferování dat a může propojovat jen segmenty se stejnou přenosovou rychlostí. Opakováč je nezávislý na protokolech linkové vrstvy, ale je závislý na specifikacích fyzické vrstvy, které typicky úzce souvisí s protokoly linkové vrstvy. Počet segmentů, které opakováč propojuje, není apriorně omezen. Opakováč v Ethernetu však musí šířit i kolize. Všechny segmenty, propojené opakováčem (opakovači), tvoří tzv. kolizní doménu a to z toho důvodu, aby i uzly v jiných segmentech poznaly, že ke kolizi došlo. Kolizní doména končí až na nejbližším mostu, switchi nebo směrovači. Proto nesmí být v Ethernetu libovolně mnoho opakováčů. Důvodem je i fungování Ethernetu (metoda CSMA/CD, která vyžaduje aby se kolize rozšířila „z jednoho konce na druhý konec“ nejdéle do pevně dané doby t), ze které plyne omezení na max. počet opakováčů. Nejjednodušší formulace pravidla zní, že mezi žádnými dvěma uzly nesmí být více jak dva opakováče. Proto se musí budovat „páteřní“ síť dle obrázku 7.25

## Počítačové sítě



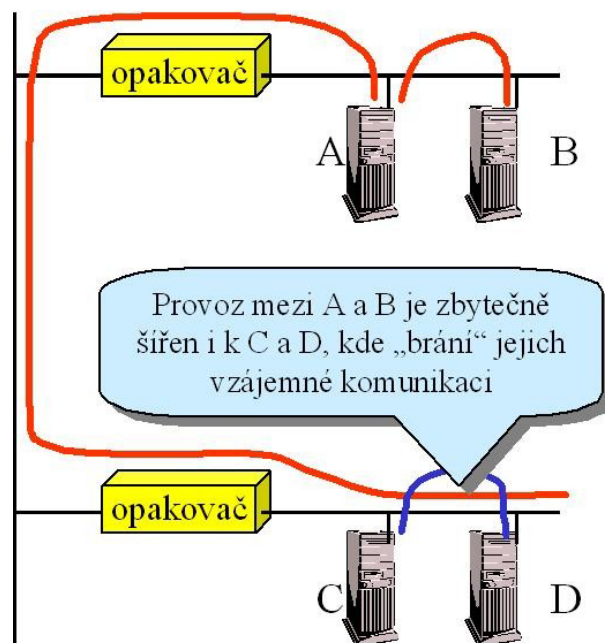
Obr. 7.25 Pátevní síť s opakovači

Exaktní formulace omezovacích pravidel se ustálila na max. 5 segmentech, tj. max. 4 opakovačích a max. 3 „obydlených“ segmentech. Ostatní jsou pouze propojovací, např. optické, a není k nim nic připojováno – viz. Obr. 7.26



Obr. 7.26 Zapojení opakovačů

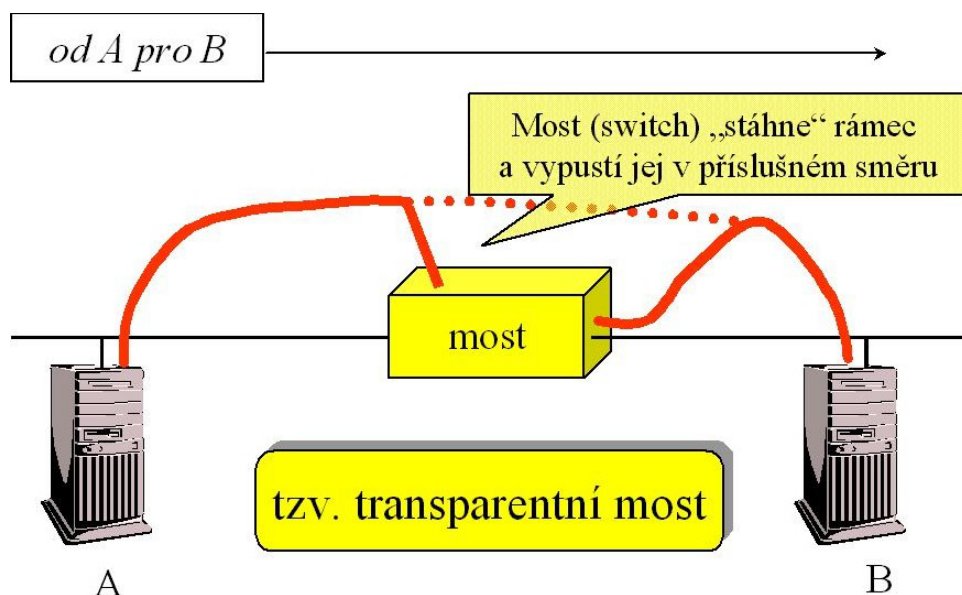
Opakovače jsou „hloupá“ zařízení, šíří do ostatních segmentů i to, co by mohlo zůstat někde lokální, plýtvají dostupnou přenosovou kapacitou – viz. Obr. 7.27



Obr. 7.27 Šíření dat s opakovači

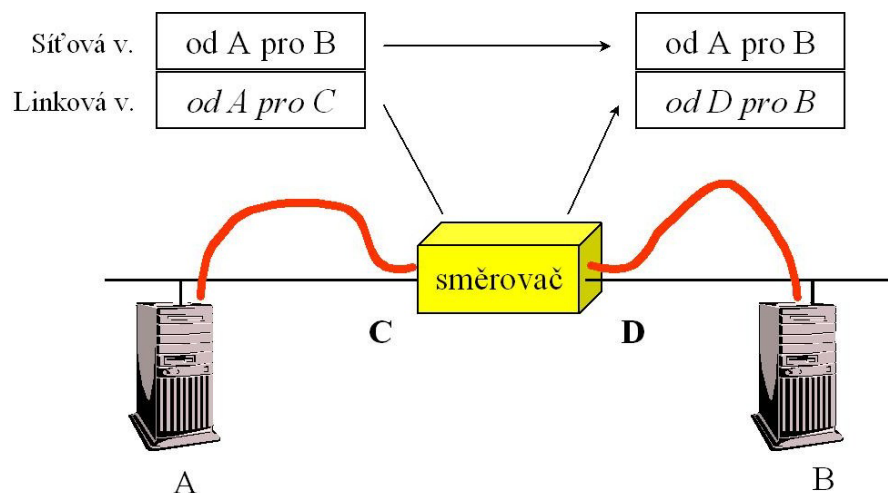
Při propojování sítí se chce dosáhnout filtrování (Filtering), tj. aby propojovací uzel dokázal poznat, co nemusí být šířeno dále a také to dále nešířil. Díky této schopnosti filtrování lze významnou měrou „lokalizovat“ provoz tj. cílené předávání (Forwarding). Aby se propojovací uzel mohl chovat inteligentně tj. posílal informace jen tam, kam to má být šířeno, a neposílat je jinde, musí alespoň trochu rozumět přenášeným datům tj. potřebuje znát adresu příjemce a adresu odesílatele. Tu může poznat z hlavičky rámce (nebo paketu, datagramu, zprávy). Propojovací uzel pak musí sám fungovat alespoň na úrovni linkové vrstvy tj. musí znát přenosové protokoly příslušné vrstvy, rozumět formátu datových bloků na příslušné úrovni a chápat význam informací, které jsou s přenosem spojeny (hlavně adresy). Aby propojovací uzel dokázal reagovat na adresy příjemce a odesílatele, nemůže už fungovat v reálném čase. Musí tedy nějakým způsobem bufferovat data, celé datové bloky nebo alespoň jejich části (takové, ze kterých jsou poznat adresy). Díky bufferování může propojovat segmenty s různými přenosovými rychlostmi, proto může být např. Ethernetový switch 100Mbps/10Mbps.

Propojovací uzel musí na úrovni linkové vrstvy propouštět a šířit data do všech segmentů (na úrovni síťové vrstvy nemusí propouštět). Co je propojeno na úrovni linkové vrstvy, tvoří jednu síť (jeden celek na úrovni síťové vrstvy). Na úrovni linkové vrstvy není propojovací uzel pro ostatní uzly viditelný, odesílatel neví o propojovacím uzlu, odesílaný rámec adresuje koncovému příjemci (v dané síti), rámec nese linkovou adresu (např. Ethernetovou) svého příjemce. Propojovací uzel funguje v tzv. promiskuitním režimu, a zachytává všechny datové rámce i takové, které mu nejsou adresovány. Za normálních okolností by mu neměly být přímo adresovány žádné rámce, propojovací uzel nemá vlastní adresu na úrovni síťové vrstvy (např. IP adresu).



Obr. 7.28 Viditelnost propojovacích uzlů

Na úrovni síťové vrstvy je propojovací uzel viditelný i pro ostatní uzly. Tyto si uvědomují jeho existenci a počítají s ní, přenášené pakety nesou v sobě síťovou adresu koncového příjemce, ale jsou odesílány na linkovou adresu propojovacího uzlu. Co je propojeno (odděleno) na úrovni síťové vrstvy, to tvoří samostatné sítě.



Obr. 7.29 Viditelnost propojovacích uzlů

Propojovací uzel musí mít dostatečné informace o skutečné topologii sítě: na úrovni linkové vrstvy (most) o svém nejbližším okolí v dosahu přímého spojení, k nejbližším směrovačům na úrovni síťové vrstvy (směrovač) o skutečné topologii sítě. Na úrovni aplikační vrstvy (brána) musí rozumět přenášeným datům.

**Získávání informací o topologii sítě**

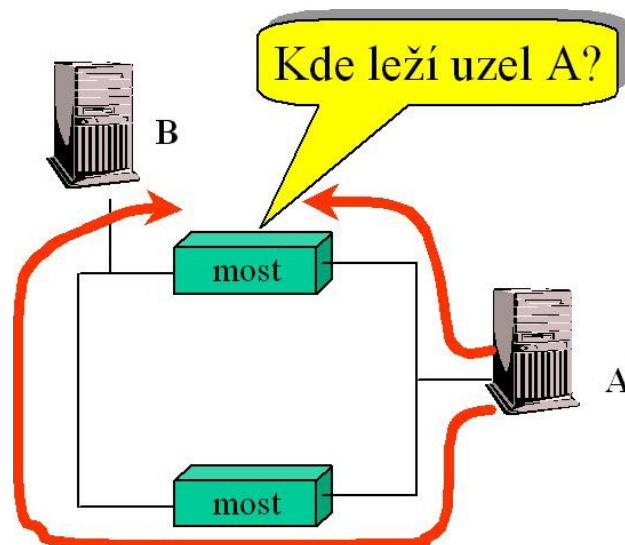
Most (i switch) musí znát své nejbližší okolí. Možnosti, jak se o něm dozví, jsou:

- statická konfigurace - informace se dodá na počátku, jednorázově

- dynamické zjišťování ve spolupráci s ostatními mosty což je zbytečně složité, není zapotřebí
- „samoučení“

Směrovač musí znát skutečnou topologii celé sítě. Objem informací je výrazně větší než u linkové vrstvy. Možnosti získávání těchto informací je pomocí statické konfigurace, statická konfigurace s dynamickým update a pomocí samoučících se mostů. Rozsah informací, které most potřebuje, je relativně malý (týká se jen nejbližšího okolí). Most je schopen fungovat i tehdy, když tyto informace nebude mít k dispozici. Pak bude fungovat jako opakovač, a rozešle všechno na všechny strany – tím to nebude efektivní, ale na krátkou dobu to lze připustit.

Ideální by bylo, kdy by si most sám získával potřebné informace ze svého okolí (učil se) a do doby než se „naučí“ fungoval neefektivně. Pak samoučící se most je „plug & play“ zařízení, které se vůbec nekonfiguruje. Princip učení je následující. Most (switch) začíná fungovat jako „tabula rasa“ tj. nemá žádné informace o topologii svého okolí a průběžně sleduje z jakých adres mu přichází jednotlivé rámce. Když dostane rámec od uzlu A pro uzel B ze směru X, odvodí si že „A leží ve směru X“, rámec rozešle do všech směrů (kromě X) a z případné odpovědi se „naučí“ umístění uzlu B. Příští rámec od A pro B již pošle cíleně jen do směru, ve kterém se B skutečně nachází. Existuje však překážka pro „samoučení“. Proces samoučení nebude fungovat, když v síti budou cykly (smyčky), pak most (switch) přijme jeden rámec z více různých směrů – viz. obr. Inteligentní mosty se dokáží vzájemně domluvit a cyklus přerušit. Aplikují algoritmus STA (Spanning Tree Alg.) a vytvoří kostru grafu.



Obr. 7. 30 Princip samoučení

V sítích Ethernet se používají výhradně samoučící se mosty (switche). V sítích Token Ring se používají mosty, fungující na principu „source routing“ - zdrojové směrování, směrování prováděné zdrojem. Podstatou „source routing“ je, že každý jednotlivý rámec si v sobě nese úplný itinerář tj. úplný seznam uzlů, přes které má projít. Tento „itinerář“ sestavuje odesílající uzel, proto „source“ routing. Tento způsob má blíže k síťové vrstvě než k vrstvě linkové a proto má v názvu „směrování“ (routing). Otázkou je, kde vezme odesílající uzel znalost o topologii sítě, na základě které sestaví úplný itinerář?



## Počítačové sítě

Před odesláním paketu (paketů) vyšle do sítě průzkumný paket (spíše rámec), který se šíří záplavově (jako lavina), až dorazí ke svému cíli. Po dosažení cíle se průzkumný paket vrací a nese v sobě údaj o cestě, kterou se k cíli dostal. Záplavové rozesílání není moc šetrné k přenosové kapacitě, ale najde skutečně „nejkratší“ cestu.

### Brány (gateways)

Mosty, switche a směrovače se nezajímají o datový obsah rámců resp. Paketů. Mohou propojovat jen takové systémy, které do rámců/paketů „balí“ stejná data tj. stejné systémy, eventuálně systémy lišící se v přenosových technologiích nižších vrstev. Pro spolupráci odlišných systémů (např. sítě na bázi TCP/IP a Novellských sítí na bázi IPX/SPX) je nutné rozumět přenášeným datům a provádět jejich konverzi. To je úkolem bran (gateways), brány jsou vždy aplikačně orientované, rozumí jen datům od určité aplikace.

Jaký je rozdíl mezi mostem a prepínačem.

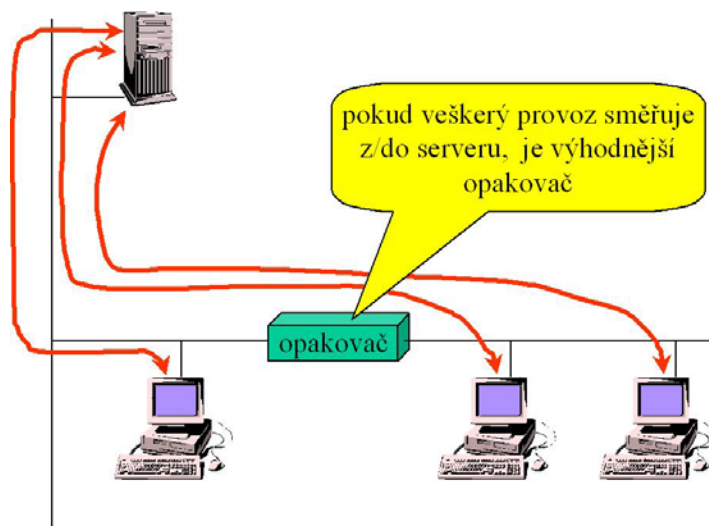
Most (bridge) propojuje několik málo segmentů (2, 3 apod.) a má za úkol omezit nepříznivé důsledky sdílení přenosové kapacity tj. zavést aspoň nějakou optimalizaci provozu. Je „starším“ řešením a jeho výkonnost (v přepojování rámců) je relativně malá.

Prepínač (switch) propojuje více segmentů (např. až desítky) a má za úkol poskytnout každé komunikující dvojici co možná nejvíce přenosové kapacity. Je „novějším“ řešením a jeho výkonnost (v přepojování rámců) je relativně vysoká.

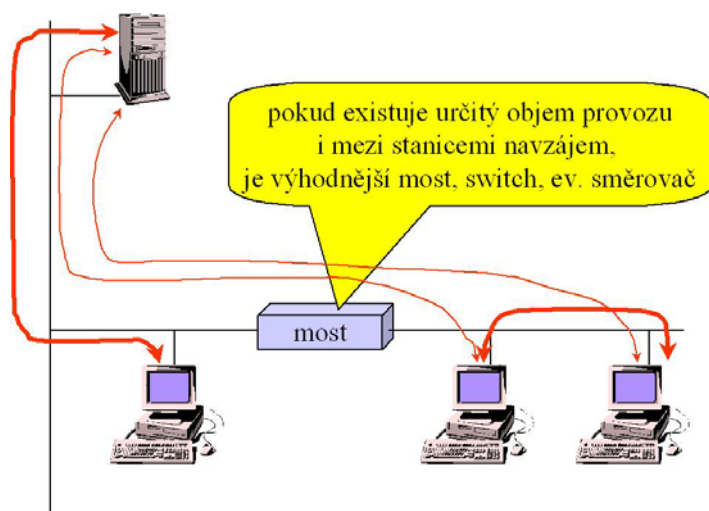
Mosty vznikly v době, kdy Ethernet používal koaxiální kabely a měl skutečně sběrníkovou topologii tj. byl technologií se sdíleným přenosovým médiem přičemž mosty se snažily udělat maximum pro využití dostupné kapacity. Později Ethernet přešel na použití kroucené dvoulinky, která není sdíleným médiem, ale Ethernet se k ní stále choval jako kdyby měla charakter sdíleného média. Přechodem na použití kroucené dvoulinky Ethernet získal stromovitou (hvězdicovou) topologii, kterou ale využíval jako „logicky sběrníkovou“. Proto prepínače jsou pokusem využít potenciál nesdílených spojů (rozvodů) na maximum. Ideální stav je možný jen při skutečně hvězdicové topologii, tj. jen při použití kroucené dvoulinky. Prepínače se liší v tom, kolik uzlů (portů) je možné připojit ke každému segmentu, kolik MAC adres si prepínač zvládne pamatovat na každém segmentu (1 port na segment = ideální stav, více portů na segment = méně-než-ideální stav), v celkové přepojovací kapacitě, zda postačuje pro ideální stav a v režimu fungování tj. na store&forward prepínače (mají větší průchozí zpoždění) a cut-through prepínače (mají menší zpoždění tzv. latency).

Prepínače se používají hlavně k tomu, aby se zvýšila celková propustnost sítě, a rozumně rozložily toky dat, aby se maximálně efektivně využila dostupná přenosová kapacita. Používají se tam, kde se dříve používaly mosty tj. hlavně

„uvnitř“ lokálních sítí, ale nikoli „na okraji“, kde se používají spíše směrovače jsou zaměřeny spíše na rychlost, a nikoli na ochranu, omezování přístupu apod. Je důležité si uvědomit, že přepínače nejsou univerzálně „lepší“ než opakovače.



Obr. 7.31 Příklad výhodnosti opakovače



Obr. 7.32 Příklad výhodnosti mostu



### Kontrolní otázky:

1. Jaký je nejčastější účel vzájemného propojování existujících počítačových sítí?
2. Jaký je princip fungování mostu?, přepínače, směrovače a na jakých vrstvách pracují?

### Úkoly k zamyšlení:

1. Jaký je nejdůležitější rozdíl mezi propojením sítí na linkové a na síťové vrstvě?
2. Může pracovat směrovač v režimu cut-through?



### Korespondenční úkoly:

1. Existují případy, kdy je vhodnější propojení uzlů na fyzické vrstvě než na vrstvě linkové?
2. Vyberte nejpoužívanější síťovou službu ve Vámi používané (lokální) síti (např. elektronická pošta, nebo sdílení souborů na serveru apod.) a podle jejího charakteru komunikace v síti rozhodněte, zda je v síti s takovouto převažující aplikací nejvhodnější jako propojovací prvek mezi stanicemi opakovač, most či přepínač.



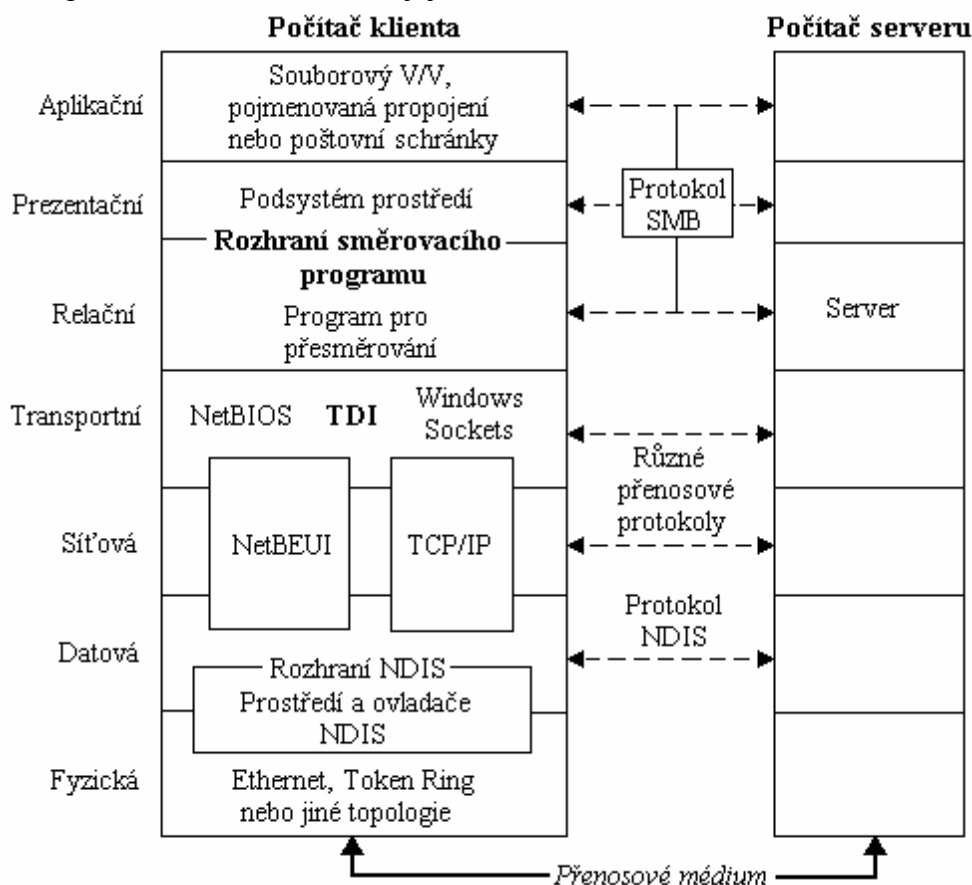
## 8 SÍŤOVÝ OPERAČNÍ SYSTÉM WINDOWS NT

Jako příklad popisu funkcí síťového modelu v operačním systému využijeme dosti známý OS WINDOWS NT.

### 8.1 Referenční model OSI ve Windows NT

Cílem síťového softwaru je vzít požadavek z aplikace na jednom počítači, předat je dalšímu počítači, vykonat požadavek na vzdáleném počítači a vrátit výsledek prvnímu počítači. To znamená, že požadavek bude několikrát v průběhu cesty přetransformován. Vysokoúrovňové požadavky, jako třeba „přečti x bytů ze souboru y na počítači z“, vyžadují, aby software určil, jak se dostat na počítač z a jakému komunikačnímu softwaru daný počítač rozumí. Když požadavek dosáhne druhou stranu musí být zkontrolován, zda je kompletní, dekodován a vyslán správné komponentě operačního systému, aby jej vykonal. Nakonec musí být odpověď zakódována, aby mohla být poslána zpět přes síť.

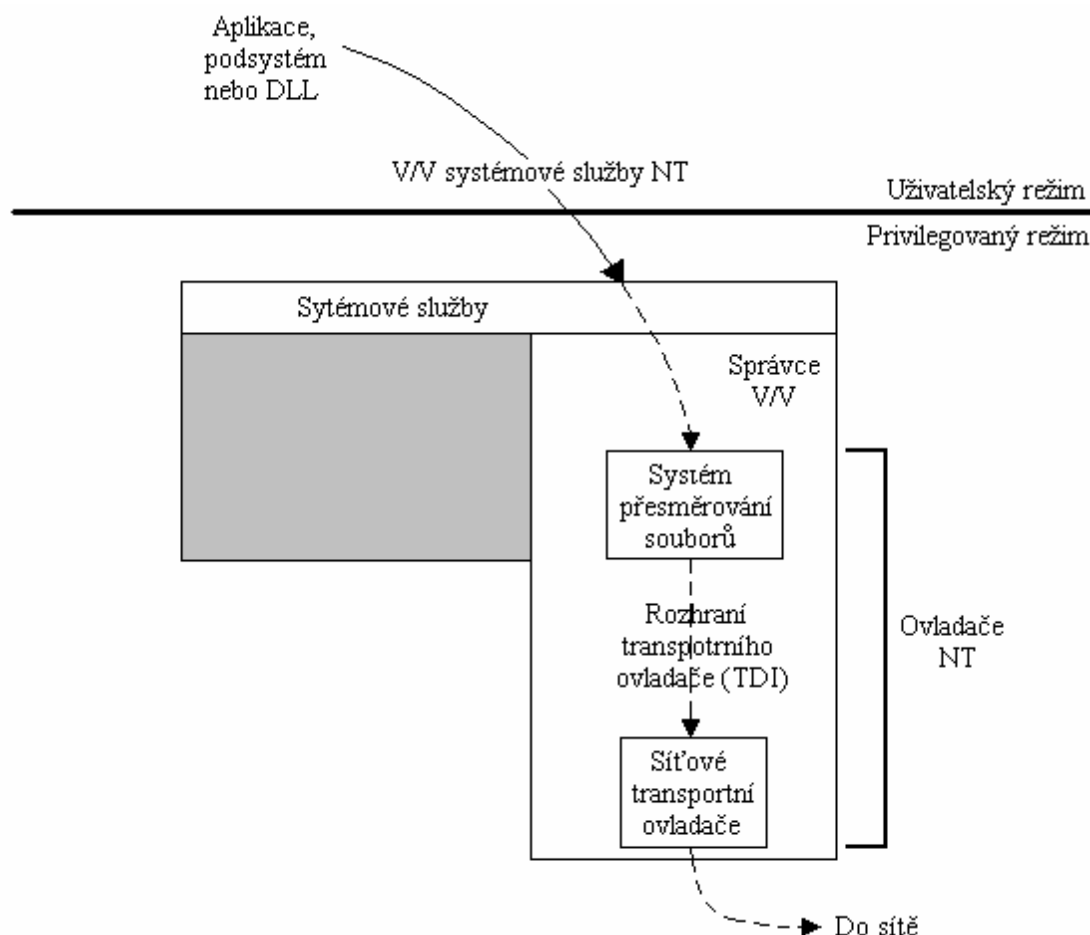
Referenční model OSI, který byl popsán výše, je však idealizované schéma, které jen málo systémů realizuje přesně, ale je často používán při diskusích o síťových principech. Každá vrstva na jednom počítači předpokládá, že „mluví“ na jiném počítači se stejnou vrstvou. Ve skutečnosti však musí síťový přenos projít každou vrstvou na klientském počítači, musí být přenesen přes síť a potom procházet vrstvami na cílovém počítači, až dosáhne vrstvy, která požadavku rozumí a která jej může realizovat.



obr. 8.1 Model OSI a komponenty sítě Windows NT

Účelem každé vrstvy v modelu je poskytovat služby vyšším vrstvám a abstrahovat způsob, jak jsou služby vykonávány na nižších vrstvách.

Následující obrázek ukazuje jak se chová V/V operace na straně klienta. Z toho lze usoudit, že vrstvy modelu OSI neodpovídají. Transportní software tak často překračuje několik hranic. Ve skutečnosti se často čtyřem spodním vrstvám říká dohromady „transport“ a softwarovým komponentům umístěných v horních třech vrstvách „uživatelé transportu“.



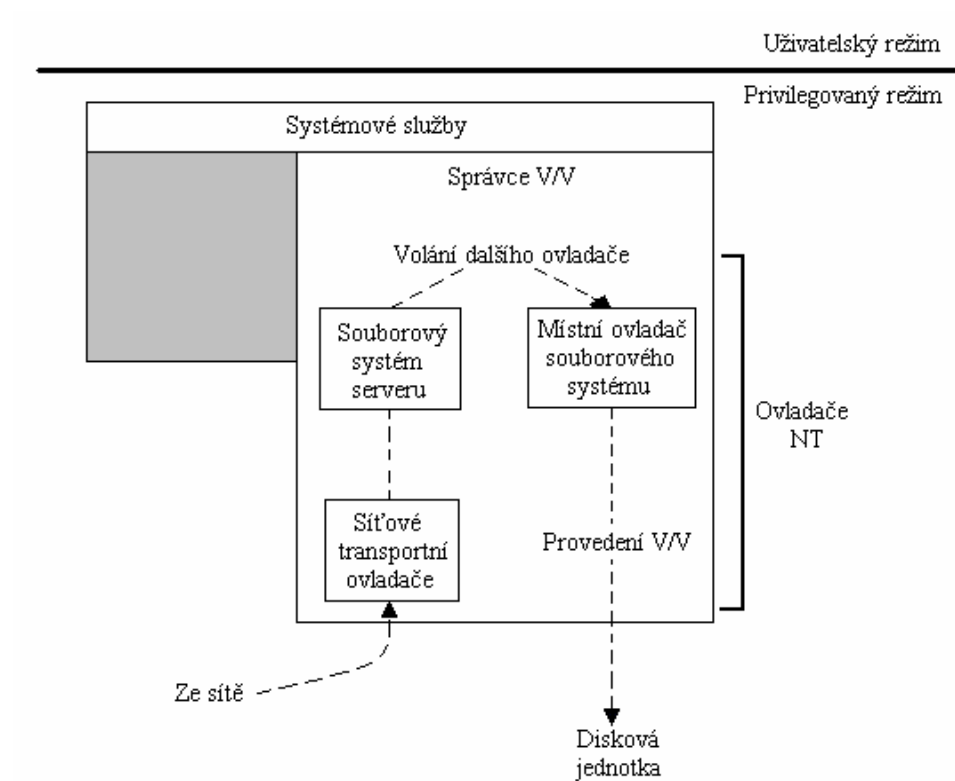
obr.8.2 Zjednodušený pohled na V/V síťovou operaci ze strany klienta

## 8.2 Zabudované síťové vlastnosti Windows NT

Software běžící v uživatelském režimu (například V/V rozhraní API pro Win32) vydá vzdálený V/V požadavek zavoláním integrované V/V služby systému NT. Po jistém úvodním zpracování správce V/V vytvoří paket V/V požadavku (IRP) a předá požadavek jednomu z registrovaných ovladačů systému správy souborů; v tomto případě programu pro přesměrování ve Windows NT. Program pro přesměrování pošle paket IRP ovladačům nižší vrstvy (transportním ovladačům), které jej zpracují a umístí do sítě. Když požadavek dorazí na místo určení ve Windows NT, je přijat ovladači transportu a pak prochází přes několik dalších ovladačů. Následující obrázek ilustruje

## Počítačové síť

přijetí síťového požadavku na zápis. Operace čtení by procházela stejnou cestou k serveru a data by se vracela opačnou cestou.



obr 8.3 Zjednodušený pohled na V/V síťovou operaci ze strany serveru

## 9 VLASTNÍ REALIZACE POČÍTAČOVÉ SÍTĚ

### 9.1 Rozvaha pro vytvoření počítačové sítě

Při rozhodování o vhodném hardwaru a softwaru musíme brát hlavní zřetel na to jaké hardwarové a softwarové vybavení máme nebo budeme mít k dispozici, kolik pracovních stanic (workstations) bude vzájemně propojených (z toho taky vyplývají nároky na rychlost, velikost a vybavenost serveru), jakou hodláme použít kabeláž, rychlost sítě (velikost datových toků a přenášených dat v síti), topologii sítě a v neposlední řadě i možnosti finančních prostředků, které můžeme na realizaci počítačové sítě použít a jestli se nám vložené investice skutečně vyplatí.

V poslední době je v praxi nepoužívanější hvězdicová (stromová) topologie podle specifikace Fast Ethernetu s využitím spojení pomocí nestíněného krouceného kabelu (UTP Cat. 5) a propojení přes hub (stohovatelné huby), která je výhodná z hlediska jednoduchosti a umožňuje jednoduché rozšiřování sítě, montáž i jednoduchou detekci možných chyb nebo poruch na síti.

Dále je potřeba zvolit architekturu sítě. Architektura *peer-to-peer* je sice jednoduchá a výhodná v malých kancelářích nebo pokud máme velmi omezené finanční prostředky, ale v případě více stanic než 5 nemá prakticky smysl a navíc neoplývá ani velkou bezpečností dat. Nejvýhodnější je proto architektura *klient-server*, která je orientovaná na centrální počítač.

Z použité kabeláže taky nutně vyplývá jaké síťové karty budeme používat. Pokud zřizujeme jednoduchou síť s uvažovanou rychlostí 10 Mb/s můžeme využít i starší typ pro sběrnici ISA, ale s ohledem na budoucí rozšiřování raději zvolíme typ pro sběrnici PCI, který umožňuje bezproblémovou komunikaci na obou rychlostech, a to 10 i 100 Mb/s. Navíc sběrnice ISA je v poslední době nahrazována rychlejšími sběrnicemi PCI a časem bude určitě zcela nahrazena. Také cenový rozdíl mezi oběma typy už v současné době není tak velký.

Při volbě hardwaru volíme raději firmy, které mají určitou tradici v síťových službách a se síťovými produkty, čímž si ušetříme řadu možných budoucích starostí s levnými neznačkovými produkty, které mohou být sice kvalitní, avšak jejich podpora od dalších výrobců, ať softwaru či hardwaru, nemusí být zrovna nejlepší.

Software sítě potom volíme podle toho, které aplikace budeme nejčastěji provozovat a jaké operační systémy mají jednotlivé stanice, aby byl síťový software na serveru plně kompatibilní v možnostech přenosu dat mezi serverem a klienty, a pokud možno dostatečně dimenzovaný na budoucí rozšiřování sítě, či počtu klientských stanic.

### 9.2 Parametry při měření kabelů s ukázkou protokolu

**Wire Map** - mapa zapojení

Při tomto testu je měřeno správné propojení jednotlivých pinů konektorů na obou koncích a stínění. Kromě správného propojení je také identifikována chyba označována jako *SPLIT PAIR*, kdy jsou vzájemně kroucený vodiče dvou různých párů.

## Počítačové síť

<b>Lenght (m)</b>	-	délka segmentu
		Zobrazuje délku kabelu. Při tomto měření je rovněž zjištěna vzdálenost k poruše, jako je zkrat, přerušení nebo změna impedance.
<b>Attenuation (dB)</b>	-	útlum
		Při měření útlumu je měřeno zeslabení signálu při průchodu kabelem.
Měření útlumu zjistí:		Útlum v jednom směru pro každý pár Nekvalitní propojení nebo ukončení Nevyhovující kabel Chyby v konektoru
<b>Local NEXT (dB)</b>	-	přeslech na blízkém konci
		Je míra signálu přecházejícího z jednoho páru do druhého v čtyřpárovém kabelu. Předpokládá se, že jen dva páry kabelu budou v daném čase použity pro přenos dat.
<b>PS-NEXT</b>	-	přeslech na blízkém konci
		Je součet NEXT přenesený na měřený pár ze všech přilehlých párů. Předpokládá se, že jsou využity všechny páry kabelu v daném čase pro přenos dat. Jedná se o naibdukovaný signál do 1 páru ze zbývajících 3 párů.
<b>FEXT (dB)</b>	-	přeslech na vzdáleném konci, obdoba NEXT, ale je ovlivněn útlumem kabelu
<b>PS-FEXT (dB)</b>	-	součet FEXT přenesených na měřený pár ze všech příhrhlých párů. Předpokládá se, že jsou využity všechny páry kabelu v daném čase pro přenos dat.
<b>ELFEXT (dB)</b>	-	obdoba ACR, vypočítává se z FEXT a útlumu.
<b>Return LOSS (dB)</b>	-	míra konzistence impedance kabelu. Velké odchylky impedance způsobují nežádoucí odrazy signálu nebo "echo", které mohou způsobovat interference přenášeného signálu.
<b>CABLE GRADING</b>	-	poskytuje prostředek pro kvalifikaci o kolik lepší je výkon linky ve srovnání požadavky testů dle TSB-67. Výkon linky je stupňován podle velikosti odstupe nejhoršího výsledku od limitu NEXTu. Každý vyšší stupeň (stupeň 1-7) reprezentuje 3dB zlepšení v NEXT/PS- NEXT hodnotě ve srovnání limitu. Používalo se v době kdy nebyla definovaná Cat5E a Cat6. V dnešní době měření ztratilo význam a nepoužívá se
<b>ACR (dB)</b>	-	rozíl mezi NEXT a útlumem při dané frekvenci
		$ACR (dB) = NEXT (dB) - Útlum (dB)$
		Ukazuje využitelnou šířku pásma linky. Minimální hodnota je 10dB. Pro CAT.5E se měří 100MHz, pro CAT.6 – 200MHz (doporučuje se 250MHz).
<b>NOISE (dB)</b>	-	šum
<b>DELAY (ns)</b>	-	zpoždění, která vzniká na páru
<b>DELAY SKEW (ns)</b>	-	zpoždění mezi páry
<b>IMPEDANCE (OHM)</b>	-	impedance páru

### Výsledky měření

<b>PASS</b>	-	vyhovující
<b>PASS*</b>	-	na hranici vyhovující / nevyhovující
<b>FAIL</b>	-	nevyhovující



# Počítačové síť



LANcat CABLE CERTIFICATION REPORT # 13

Circuit ID: Local Module Type: C5e Modular Plug  
 Cable Test Standard: Remote Module Type: C5e Modular Plug  
 Location: Serial Number: 0205404 V6.20D4  
 Date Tested: Cable NVP: 72.0%

TEST SUMMARY: PASS

Wire Map: PASS	Near End	1	2	3	6	4	5	7	8	Shield
										open
	Remote End	1	2	3	6	4	5	7	8	

Length:	PASS	Limit:	100m	
Delay:	PASS	Limit:	555ns	
Delay Skew:	PASS	Limit:	50ns	
Pair	Length	Delay	Delay Skew	Comments
1,2	53m	249ns	4ns	OPEN
3,6	52m	245ns	0ns	OPEN
4,5	53m	248ns	3ns	OPEN
7,8	52m	245ns	0ns	OPEN

Attenuation:	PASS	1,2	3,6	4,5	7,8
		PASS	PASS	PASS	PASS
Attenuation	dB	10.8	10.6	10.7	10.5
Limit	dB	24.1	24.1	24.1	24.1
Margin	dB	+13.3	+13.5	+13.4	+13.6
Frequency	MHz	101.0	101.0	101.0	101.0

Local NEXT:	PASS	12/36	12/45	12/78	36/45	36/78	45/78
		PASS	PASS	PASS	PASS	PASS	PASS
NEXT	dB	49.7	39.6	66.3	41.5	42.4	49.8
Limit	dB	36.3	31.3	55.8	30.7	32.7	40.7
Margin	dB	+13.4	+8.3	+10.5	+10.8	+9.7	+9.1
ACR	dB	43.0	29.8	64.7	31.3	33.8	44.9
Frequency	MHz	43.2	84.8	2.8	92.0	70.4	23.7

Remote NEXT:	PASS	12/36	12/45	12/78	36/45	36/78	45/78
		PASS	PASS	PASS	PASS	PASS	PASS
NEXT	dB	42.7	42.9	43.2	42.7	49.0	44.5
Limit	dB	32.2	32.9	31.9	30.7	40.0	34.7
Margin	dB	+10.5	+10.0	+11.3	+12.0	+9.0	+9.8
ACR	dB	33.8	34.2	34.1	32.5	43.7	37.1
Frequency	MHz	74.7	68.1	78.3	92.0	26.2	53.7

PSNEXT:	PASS	1,2	3,6	4,5	7,8
		PASS	PASS	PASS	PASS
Power Sum	dB	39.4	44.9	37.8	38.9
Limit	dB	28.3	33.2	28.2	29.6
Margin	dB	+11.1	+11.7	+9.6	+9.3
Frequency	MHz	84.8	43.2	85.2	70.8

Remote PSNEXT:	PASS	1,2	3,6	4,5	7,8
		PASS	PASS	PASS	PASS
Power Sum	dB	40.5	39.4	42.3	39.8
Limit	dB	29.9	29.2	29.9	28.9
Margin	dB	+10.6	+10.2	+12.4	+10.9
Frequency	MHz	68.1	74.9	68.1	77.4

Return Loss:PASS		1,2	3,6	4,5	7,8
		PASS	PASS	PASS	PASS
Return Loss	dB	21.1	20.2	21.2	20.5
Limit	dB	11.4	10.0	10.9	11.8
Margin	dB	+9.7	+10.2	+10.3	+8.7
Frequency	MHz	72.4	99.6	81.2	65.2

Rem Return Loss:PASS		1,2	3,6	4,5	7,8
		PASS	PASS	PASS	PASS
Return Loss	dB	22.2	23.6	20.3	19.5
Limit	dB	11.3	11.3	11.3	11.8
Margin	dB	+10.9	+12.3	+9.0	+7.7
Frequency	MHz	72.6	73.5	72.9	65.2

ELFEXT: PASS		12/36	12/45	12/78	36/45	36/78	45/78
		PASS	PASS	PASS	PASS	PASS	PASS

## Počítačové síť

ELFEXT	dB	70.0	77.2	69.7	80.1	35.7	82.4
Limit	dB	50.7	58.1	51.6	58.1	17.5	58.1
Margin	dB	+19.3	+19.1	+18.1	+22.0	+18.2	+24.3
Frequency	MHz	2.1	0.9	1.9	0.9	98.7	0.9
		36/12	45/12	78/12	45/36	78/36	78/45
		PASS	PASS	PASS	PASS	PASS	PASS
ELFEXT	dB	39.3	37.1	62.2	81.6	39.7	82.7
Limit	dB	20.6	17.9	43.7	58.1	21.8	58.1
Margin	dB	+18.7	+19.2	+18.5	+23.5	+17.9	+24.6
Frequency	MHz	69.3	94.2	4.8	0.9	60.0	0.9
Remote ELFEXT:	PASS	12/36	12/45	12/78	36/45	36/78	45/78
		PASS	PASS	PASS	PASS	PASS	PASS
ELFEXT	dB	39.3	36.5	61.0	81.6	40.0	82.9
Limit	dB	20.6	17.7	42.6	58.1	21.8	58.1
Margin	dB	+18.7	+18.8	+18.4	+23.5	+18.2	+24.8
Frequency	MHz	69.3	96.5	5.5	0.9	60.0	0.9
		36/12	45/12	78/12	45/36	78/36	78/45
		PASS	PASS	PASS	PASS	PASS	PASS
ELFEXT	dB	49.8	77.5	69.9	80.2	35.4	82.2
Limit	dB	30.4	58.1	51.6	58.1	17.5	58.1
Margin	dB	+19.4	+19.4	+18.3	+22.1	+17.9	+24.1
Frequency	MHz	22.3	0.9	1.9	0.9	98.7	0.9
PSELFEXT:	PASS	1,2	3,6	4,5	7,8		
		PASS	PASS	PASS	PASS		
Power Sum	dB	65.7	46.2	74.7	66.7		
Limit	dB	48.1	27.2	55.0	48.6		
Margin	dB	+17.6	+19.0	+19.7	+18.1		
Frequency	MHz	2.0	22.7	0.9	1.9		
REM PSELFEXT:	PASS	1,2	3,6	4,5	7,8		
		PASS	PASS	PASS	PASS		
Power Sum	dB	64.5	35.2	75.0	37.4		
Limit	dB	46.9	16.5	55.0	18.8		
Margin	dB	+17.6	+18.7	+20.0	+18.6		
Frequency	MHz	2.3	78.0	0.9	60.0		

Operator: Date:

Comments:

END REPORT #13

## 10 Rodina protokolů TCP/IP, architektura TCP/IP

### 10.1 Vznik rodiny protokolů TCP/IP

Historie protokolové sady TCP/IP začíná v 60. letech 20. století, kdy na akademických pracovištích zejména v USA vznikla koncepce „přepojování paketů“ jako alternativa dosud převládající koncepce „přepojování okruhů“ převzaté z telekomunikačních sítí.

Princip **přepojování paketů**, který byl popsán v kapitole 1.2 a 7.5, je vcelku prostý, avšak pro jeho důležitost jej zde znovu připomeneme. Místo toho, aby se nejprve vytvořil okruh (zpravidla virtuální), po němž se data posílají, se při využití koncepce přepojování paketů data v blocích (tzv. paketech) předávají síti, která se podle adresy příjemce postará o jejich doručení, a odesílatel se nezabývá tím, kudy síť data doručí příjemci. Princip přepojování paketů byl pro provoz datových sítí navržen až v 60. letech 20. století. Do té doby se používal výlučně princip přepojování okruhů běžný v tehdejších telekomunikačních sítích, kde převažuje dodnes.

Výzkumem v té době převratné koncepce přepojování paketů se věnovalo několik akademických pracovišť především v USA, která získala pro praktické ověření výsledků tohoto výzkumu finanční podporu agentury ARPA. Tak vznikla síť ARPANET, která byla po ukončení výzkumných úkolů předána do užívání akademickým pracovištím (nejen těm, která se podílela na výzkumu).

V síti ARPANET se používal protokol NCP, což byl experimentální protokol, který se příliš nehodil pro rutinní používání. Proto byly brzy zahájeny práce na jeho nahrazení protokolovou sadou TCP/IP. Její vývoj byl opět financován agenturou ARPA. V průběhu 70. let 20. století byla protokolová sada TCP/IP z větší části vyvinuta a implementována do některých operačních systémů, především Unixu. Protože velká část těchto prací byla financována vládou USA (prostřednictvím různých agentur), musel být jejich výsledek (protokolová sada TCP/IP) dán k dispozici veřejnosti. Celý Internet (ve který se původní síť ARPANET mezitím přeměnila) přešel na používání protokolů TCP/IP na počátku roku 1983.

### 10.2 Architektura TCP/IP

Architektura TCP/IP se dosti zásadně liší od referenčního síťového modelu TCP/IP. Odlišnost je dána především okolnostmi vzniku popsanými výše, neboť při vzniku TCP/IP měla hlavní slovo akademická sféra nezatížená stereotypy pocházejícími z telekomunikačních sítí. Tím byly dány hlavní zásady TCP/IP:

- budování „zdola nahoru“ a důraz na praxi (nejprve se vytvoří funkční protokoly, a ty se teprve následně po ověření funkčnosti začlení do struktury celé architektury);
- snaha vytvářet jen ta řešení, která objektivně chybí (na rozdíl od ISO/OSI, který se snaží do svých standardů zahrnout všechna existující řešení a vytvořit tak komplexní sadu standardů);
- eliminování potřeby centrálního prvku sítě (síť by měla být schopna provozu i při výpadku kteréhokoli prvku);

## Počítačové sítě

- snaha přizpůsobit se předpokládaných výpadkům částí sítě a jejich nespolehlivosti.

Z těchto výchozích odlišností pak vyplývají odlišnosti v konkrétní stavbě architektury TCP/IP:

- nižší počet vrstev než ISO/OSI ;
- ponechání větší možnosti volby na všech vrstvách;
- přednost mají nespojovaná a nespolehlivá komunikace, při uplatnění principu „best effort“ (maximální snahy).

Z výše uvedených důvodů se architektura TCP/IP ustálila na 4 vrstvách, jak znázorňuje tabulka 10.1.

TCP/IP		ISO/OSI
aplikační vrstva		aplikační vrstva prezentační v. relační v.
transportní vrstva		transportní vrstva
síťová vrstva (IP vrstva)		síťová v.
vrstva síťového rozhraní		linková (spojová) v. fyzická v.

Tabulka 10.1: porovnání vrstev TCP/IP a ISO/OSI

### Nespojovaná a nespolehlivá komunikace

O referenčním síťovém modelu ISO/OSI víme, že dává přednost **spojově orientované komunikaci**, tedy komunikaci, kdy se před zahájením přenosu vytvoří virtuální kanál od odesílatele k příjemci, a ten se pak využije pro přenos datových paketů. Po ukončení přenosu se tento virtuální kanál zruší.

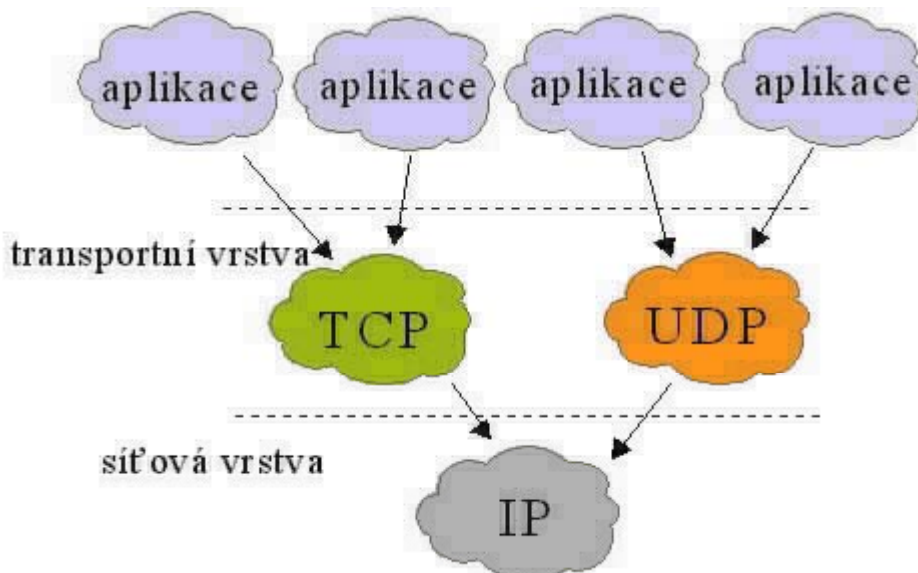
Naproti tomu v TCP/IP se v hojně míře (jak uvidíme dále, tak především na síťové vrstvě) uplatňuje pravý opak, **nespojová komunikace**. Proč? Počítá se totiž, jak bylo výše řečeno, s výpadky a nespolehlivostí různých částí sítě. Pokud by se používala spojově orientovaná komunikace, bylo by nutné při každém výpadku spojení obnovovat, a obnovení je spojeno s nezanedbatelným režijním provozem. Totéž platí při změnách v síti (např. její topologie). V případě nespojové komunikace taková režie nevzniká.

Je zřejmé, že to má svá omezení: nespojová komunikace se hodí pro tzv. shlukové přenosy s nízkou průměrnou přenosovou rychlostí, kdy se občas přenášejí větší objemy dat, nicméně po většinu času se nepřenášejí téměř žádná data. Nehodí se však určitě pro případ trvalých toků většího množství dat.

V TCP/IP se (na nižších vrstvách) rovněž dává přednost nespolehlivé komunikaci. Důvodem je skutečnost, že i zajištění spolehlivosti vyžaduje určitou režii, a pokud by touto režii byla zatížena např. síťová vrstva, nemohla by se jí vyšší vrstva vyhnout, i pokud by spolehlivý přenos nevyžadovala. Navíc vzhledem ke skutečnosti, že je prakticky velmi obtížné docílit 100% spolehlivosti, mohlo by se stát, že nabízená úroveň spolehlivosti je pro některou aplikaci i tak nedostatečná, a tak by si musela spolehlivost zajistit sama, čímž by docházelo k dalšímu zvyšování režie a plýtvání přenosovou

kapacitou. Je nutné upozornit, že i přesto, že nižší vrstvy fungují nespojovaně a nespolehlivě, mohou nad nimi vyšší vrstvy fungovat spolehlivě a v případě potřeby i spojovaně.

Z toho důvodu bylo v TCP/IP zvoleno řešení, které umožňuje si aplikaci zvolit, zda spolehlivost vyžaduje či nikoli. Pokud ano, použije spolehlivou spojovanou službu transportní vrstvy, reprezentovanou protokolem TCP, v opačném případě použije nespolehlivou nespojovanou službu protokolu UDP (viz obrázek 10.1).



Obr. 10.1: Volba služby transportní vrstvy aplikací

Rozhodnutí tvůrců protokolové sady TCP/IP umožnit alternativní službu až na transportní vrstvě a ponechat služby síťové vrstvy pouze nespolehlivé a nespojované znamená, že spolehlivost se bude zajišťovat pouze na koncových uzlech. Na mezilehlých uzlech (nejčastěji směrovačích), které zahrnují pouze funkce vrstvy síťového rozhraní a vrstvy síťové (viz obrázek 10.2), bude jedinou možnou službou nespolehlivý a nespojovaný přenos. Toto rozhodnutí je dosti zásadní pro celou koncepci budování TCP/IP sítí, protože to znamená, že síťová infrastruktura včetně mezilehlých uzlů (směrovačů) bude pracovat co nejjednodušeji a zároveň s maximálním omezením režie. Veškeré dodatečné služby (mezi něž zajištění spolehlivosti patří) budou implementovány pouze na koncových uzlech.

Toto rozhodnutí je koncepčního charakteru zejména pro další rozvoj služeb TCP/IP sítí. Pokud totiž například nějaká budoucí verze protokolu TCP zdokonalí mechanismus zajištění spolehlivosti (či dokonce nahradí protokol TCP jiným), nebude nutné kvůli tomu provádět žádné změny na síťové infrastruktuře, postačí pouze změny v samotných koncových uzlech, kde se díky častější obměně hardware i operačních systémů provádějí snáze.

## Počítačové síť



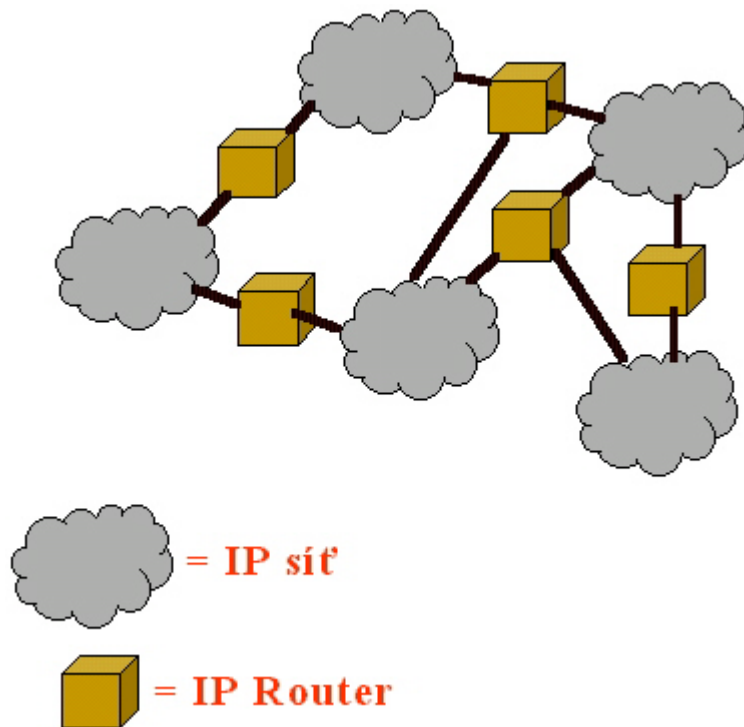
Obr 10.2: rozdělení vrstev na mezilehlých uzlech (směrovačích) a na koncových uzlech

Řečeno jinými slovy to znamená, že „intelligence“ potřebná k provádění složitějších operací (k nimž zajištění spolehlivosti patří) a spolu s ní potřebný výpočetní výkon je umístěna především na koncových uzlech a nikoli v síti. Zde spočívá jeden z důležitých rozdílů proti telekomunikačním sítím, které umísťují inteligenci naopak především do sítě.

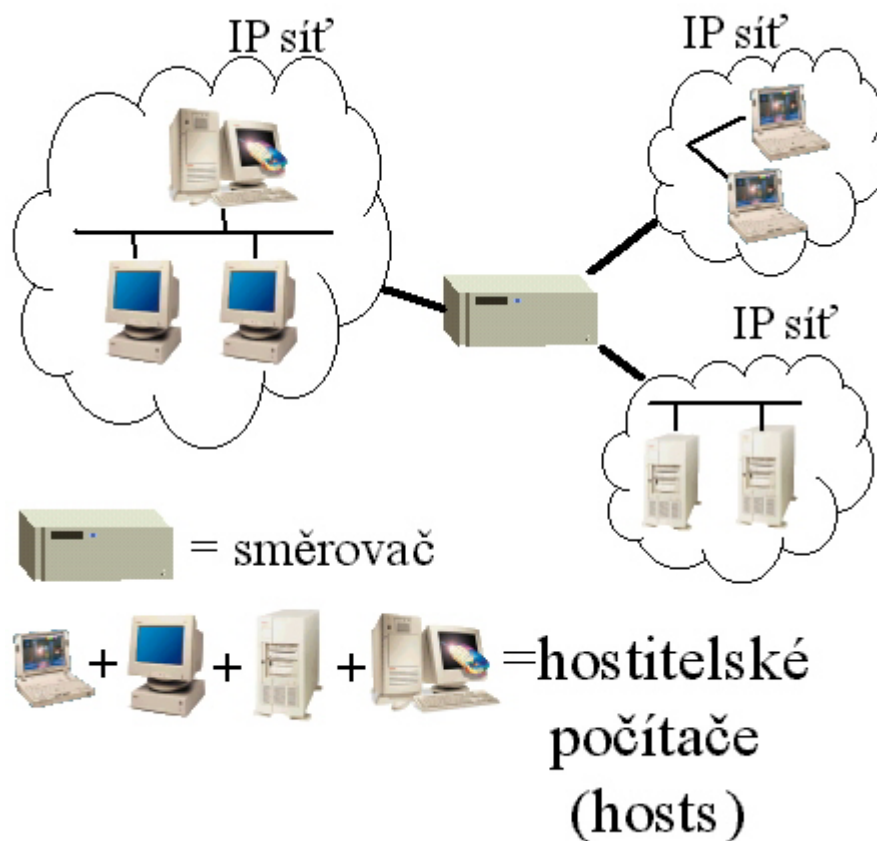
Princip maximální snahy zmíněný výše ovšem znamená, že v případě požadavků přesahujících kapacitu některé části sítě může přetížená část sítě (například směrovač) požadavky krátit, a to buď pozdržením jejich vyřízení, nebo i úplným zahazením některých paketů. Důležité přitom je si uvědomit, že protokolová sada TCP/IP nemá v sobě zabudovány žádné prioritizační mechanismy, takže při krácení požadavků jsou všechny požadavky kráceny stejnou měrou. Není tedy možné např. při zahazování paketů; nejdříve zahazovat pakety elektronické pošty nebo FTP a tak omezit třeba opoždění datového toku u služeb, které jsou na jeho pravidelnosti více závislé (třeba WebTV apod.). Je zřejmé, že tato vlastnost TCP/IP poněkud brání rozvoji především multimediálních služeb. Naproti tomu lze ukázat, že síť fungující na principu „best effort“ je mnohem efektivnější pro veškeré datové přenosy. Je třeba si uvědomit i skutečnost, že právě tato efektivnost pomohla Internetu k tak rychlému rozvoji.

Při vzniku protokolové sady TCP/IP se počítalo s tím, že jednotlivé dílčí sítě budou propojeny pomocí směrovačů, přičemž z požadavku robustnosti (zachování funkce při výpadku některé části sítě) vyplývá, že jsou přípustná (a dokonce žádoucí) redundantní propojení. Pochopitelně nutno podmínkou provozuschopnosti sítě je souvislost grafu sítě, tedy existence neméně jednoho spojení mezi každými 2 uzly. Typickou malou část sítě si můžeme představit podle obrázku 10.3. Bližší představu o propojení sítí získáme pomocí obrázku 10.4.

## Počítačové sítě



Obr 10.3: Představa IP sítě.



Obr 10.4: IP síť a jejich propojení.

Jak je vidět na obr. 10.4 a jak již bylo zmíněno výše, jsou IP sítě tvořeny dvěma druhy uzlů. Na jedné straně většina uzlů jsou koncové uzly neboli podle původní terminologie TCP/IP **hostitelské počítače** (host computers), což jsou

nejčastěji pracovní stanice, servery, ale i tiskárny či faxy se síťovou kartou apod. Společným rysem těchto uzlů je **připojení právě do jedné sítě**. Zpravidla v početní menšině pak jsou uzly druhého typu, totiž **směrovače** (routery, dříve se pro ně používal termín gateway), které zajišťují propojení mezi 2 nebo více sítěmi.

I zde platí určitá pravidla pro nesměšování těchto typů uzlů. Podobně jako v lokálních sítích pravidlo, že server by neměl sloužit jako zároveň pracovní stanice, protože jeho funkci serveru (která je přirozeně důležitější, protože obsluhuje více uživatelů) to může ohrožovat, tak i v IP sítích platí, že směrovač by neměl plnit další funkce, a to z podobného důvodu. Nejspíše by připadala v úvahu funkce serveru, neboť zejména v menších sítích se můžeme často setkat se směrovači postavenými na bázi PC (nejčastěji s OS Linux). Přesto se s takovým uspořádáním (nazývá se „multihomed host“) někdy setkáváme. U specializovaných směrovačů vestavěných typicky do stojanového modulu současné využití pro jiné funkce pochopitelně zpravidla nepřichází v úvahu, neboť zpravidla pracují se speciálním operačním systémem, který něco takového neumožňuje.

### 10.3 Rozdělení TCP/IP do vrstev

#### Vrstva síťového rozhraní a síťová vrstva

Koncepce nižších vrstev protokolové sady TCP/IP pramení ze stavu při vzniku ARPANETu, kdy bylo nutné propojit technologicky naprosto různé sítě. Proto bylo přijato rozhodnutí, že síťová vrstva vytvoří **jednotné** rozhraní mezi nejrozličnějšími technologiemi používanými ve vrstvě síťového rozhraní a vyššími vrstvami TCP/IP. Autoři TCP/IP se nesnažili vytvořit své vlastní řešení i pro vlastní přenos dat, ale zde spoléhali na existující prověřená řešení zavedených výrobců (např. Ethernet, ATM, TokenRing, FDDI, ISDN, Frame Relay apod.).

Z výše uvedeného stanoviska vyplývá, že aby bylo možné překrýt různé přenosové technologie jednotnou síťovou vrstvou, musí nutně síťová vrstva využívat pouze těch služeb různých technologií ve vrstvě síťového rozhraní, které jsou schopny nabídnout všechny existující technologie. Z těchto důvodů není např. možné využít spolehlivého přenosu určité technologie (např. ATM). Ze stejných důvodů dokonce některé technologie působí při spolupráci s TCP/IP problémy, protože v TCP/IP se pro určité činnosti počítá s možností všesměrového vysílání na vrstvě síťového rozhraní, což některé technologie (opět např. ATM) nenabízejí. Z výše uvedených důvodů se na síťové vrstvě používá přenosový protokol, který je nespolehlivý a nespojovaný. Z těchto důvodů se zavádí jednotné adresování pomocí 32-bitových adres, která mají v sobě část identifikující síť, a část identifikující uzel v rámci sítě. Vzhledem k zavedení virtuálních IP adres, které neobsahují žádnou informaci o adresách z vrstvy síťového rozhraní, je nutné vytvořit převodní mechanismy mezi adresami fyzickými (linkovými) a virtuálními IP adresami. Síťová vrstva musí pochopitelně mít k dispozici také některé informace o vlastnostech vrstvy síťového rozhraní, která pracuje pod ní. Jde především o parametr MTU (Maximum Transfer Unit), který určuje maximální velikost rámce, který může vrstva síťového rozhraní přenést.

Jak již bylo uvedeno výše, protokolová sada TCP/IP nedefinuje protokoly



síťového rozhraní, ale pouze to, jak existující propojit síťovou vrstvu s vrstvou síťového rozhraní, kde jsou používány různé protokoly (též označované jako přenosové technologie). Tyto technologie samotné však nejsou protokolovou sadou TCP/IP definovány.

Určitou výjimku představují protokoly SLIP a PPP určené pro provoz na dvoubodových spojích. Důvod jejich vzniku je nasnadě: uživatelé potřebovali jednoduché protokoly pro provoz na dvoubodových spojích (často využívaných pro připojení malých sítí nebo jednotlivých počítačů k TCP/IP sítím), a tyto protokoly nebyly v době vzniku SLIP k dispozici.

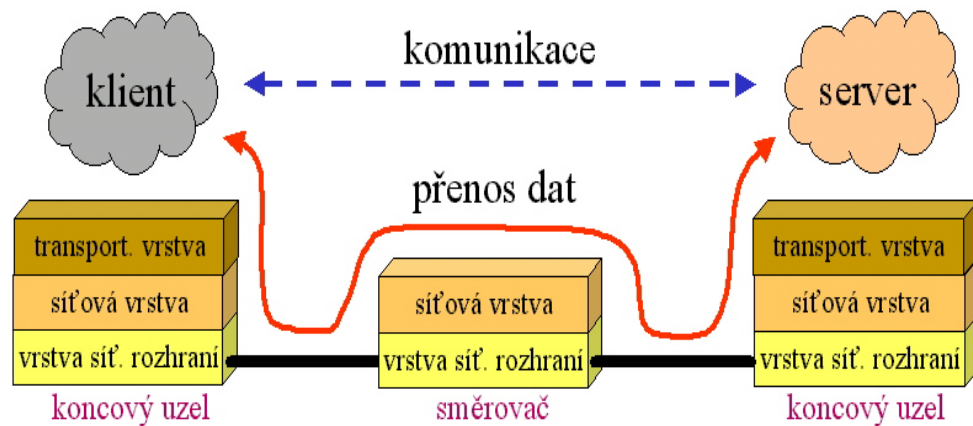
### **Vyšší vrstvy - transportní a aplikační**

Funkce transportní vrstvy v TCP/IP jsou podobné jako v referenčním modelu ISO/OSI, odlišné je však uspořádání vrstev nad ní. Na rozdíl od 3 vrstev ISO/OSI je zde pouze jediná vrstva, totiž aplikační. Znamená to, že všechny funkce, které jsou v referenčním modelu ISO/OSI svěřeny vrstvě aplikační, prezentační a relační, musí v TCP/IP zajistit samotná aplikační vrstva. Je to ostatně v souladu s filosofií TCP/IP, která se snaží zatěžovat režii jen ty entity, které využívají služby tuto režii vyžadující. Znamená to tedy, že zatímco v ISO/OSI musí např. služby prezentační vrstvy, která zabezpečuje např. konverze dat, využívat všechny aplikace, v TCP/IP si tyto služby zajišťuje aplikace sama, proto ta, která tyto služby nepotřebuje, nenese jejich režii.

Na druhé straně má tento přístup nevýhodu v tom, že některé služby (např. právě již zmiňované konverze dat), se díky tomu programují vícekrát. Ovšem to plně platilo jen do doby začlenění protokolu NFS mezi aplikační protokoly sady TCP/IP. Jeho samostatně použitelnou součástí totiž byly také protokoly RPC (Remote Procedure Call) a XDR (eXternal Data Representation).

Mezi nejdůležitější aplikace v aplikační vrstvě (bývají často označovány jako služby) patřila původně elektronická pošta (protokoly SMTP, POP, později IMAP), přenos souborů (protokol FTP) a vzdálené přihlašování (telnet). původním aplikacím plně vyhovoval princip maximální snahy, neboť již ze své podstaty nevyžadují. Později k nim přibýly další aplikace (např. sdílení souborů pomocí protokolu NFS, prezentace obsahu stránek pomocí WWW, chat apod.), pro něž již princip maximální snahy již představoval určité omezení funkčnosti, ne však zásadního charakteru. V případě dostatečné přenosové kapacity i tyto aplikace fungují bez problémů.

Je nutno si uvědomit, že všechny aplikace v TCP/IP jsou založena na výpočetním modelu klient/server, tedy na jedné straně stojí klientský program poskytující dané služby (např. WWW server), proti němu stojí na druhé straně příslušný klient (v uvedeném příkladě web prohlížeč). Toto uspořádání aplikací je znázorněno na obr. 10.5.



Obr 10.5: Uspořádání aplikací v TCP/IP

Během vývoje TCP/IP se objevily i aplikace, pro které je fungování aplikační vrstvy v TCP/IP nevhodné. Jedná se jednak o aplikace vyžadující distribuci identických dat od 1 zdroje k více příjemcům (např. přenos rozhlasového, televizního či video signálu apod.). Je zřejmé, že distribuce většího množství dat pomocí dvoubodových spojení klient-server je neefektivní, při větším množství klientů dokonce může být i technicky nemožná.

Některé z výše zmíněných aplikací narážejí v TCP/IP na další problém, kterým je neexistující podpora tzv. Quality of Services (QoS) neboli kvality služeb, což znamená, že v TCP/IP není možné garantovat např. omezenou velikost přenosového zpoždění, jeho omezené kolísání apod. To představuje zásadní omezení především pro aplikace, které potřebují přenášet v reálném čase např. videosignál, zvuk apod.

Zřejmě nejčastěji zmiňovaným problémem TCP/IP je nedostatečná bezpečnost. Zde se však jedná spíše o nepochopení, než o skutečný problém. Autoři TCP/IP neměli za úkol žádné bezpečnostní mechanismy vytvořit. Vycházeli z toho, že pokud nějaká aplikace bude zabezpečení požadovat, musí si je zabezpečit sama. Důvodem k tomuto přístupu byla mimo jiné již zmiňovaná filosofie minimalizace režie, tedy snaha neklást režii za zabezpečovací mechanismy na ty uživatele, kteří je nevyžadují.

Problém nastal v okamžiku, kdy začali běžné aplikace TCP/IP (např. elektronickou poštu) používat uživatelé bez dostatečné osvěty i pro přenos citlivých dat. Nebyli si přitom vědomi skutečnosti, že svěřit obchodní dopis běžnému e-mailu je z hlediska zabezpečení obsahu před čtení nepovolanou osobou totéž jako jej poslat na korespondenční lístek, což by jistě učinil málokdo. Na rozdíl od výše zmiňovaných problémů s distribučními aplikacemi či chybějící podporou QoS lze bezpečnostní mechanismy do aplikací na aplikační vrstvě poměrně snadno začlenit.

## 10.4 Další aspekty TCP/IP: standardizace, dohled nad fungováním Internetu

### Standardizace v TCP/IP

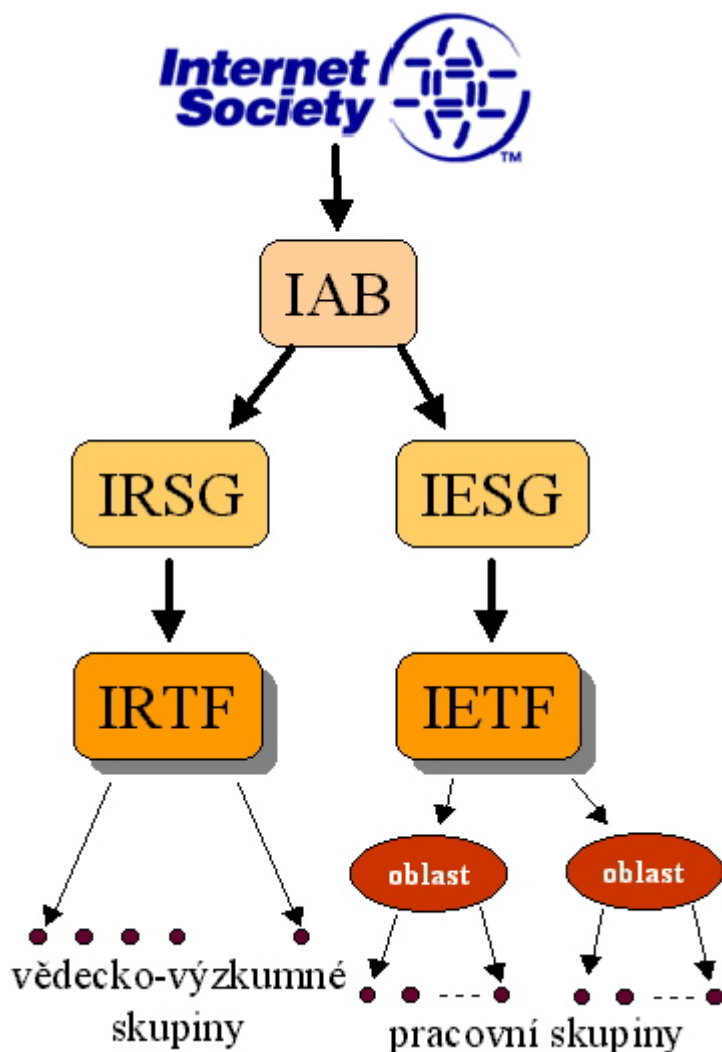
V počátcích vývoje protokolové sady TCP/IP se jako víceméně univerzální komunikační nástroj vyvinuly dokumenty RFC pojmenované podle původního

určení Request for Comment (žádost o komentář).

Dokumenty RFC, jakmile jsou oficiálně vydány, se nikdy nemění. Při vydání je každému RFC dokumentu dáno pořadové číslo. Dokumenty RFC jsou volně dostupné, jejich přehled je k dispozici např. na adrese [http://www.ietf.org/iesg/1rfc\\_index.txt](http://www.ietf.org/iesg/1rfc_index.txt). Dokumenty RFC se dělí na 2 skupiny:

- standard (dokumenty popisující standardy TCP/IP)
- off-track (ostatní dokumenty, zejména. informativní, experimentální, prototypové, historické)

V případě, že je třeba popsat určitou skutečnost novým (např. v definici nové verze některého protokolu), se původní RFC dokument označí jako zastaralý (obsolete), ale nemění se. Nový text je vydán v novém RFC dokumentu. Vzhledem k tomu, že RFC dokumentů je velké množství (početně výrazně převažují off-track dokumenty), je orientace v nich např. při vyhledávání určitého standardu poněkud obtížná. Proto vznikly tzv. dokumenty STD, které obsahují vždy aktuální RFC dokument popisující příslušný standard. Ten však přitom však zůstává součástí řady RFC dokumentů.



Obrázek 10.6: Struktura standardizačních orgánů

Standardizace je řízena standardizačními orgány zastřešenými organizací

ISOC, jejichž struktura je uvedena na obrázku 10.6. Nejdůležitější výkonnou roli v tomto procesu hraje organizace Internet Engineering Task Force (IETF). Samotný proces a standardizace má 3 fáze, kterými musí projít každý normotvorný dokument RFC. Tyto fáze se označují:

Proposed Standard (návrh standardu, podmínkou jsou 2 nezávislé implementace);

Draft Standard (vyžadují se provozní zkušenosti);

Internet Standard (konečná podoba standardu).

Faktické vytváření návrhů standardů bylo dříve zajišťováno pracovními skupinami ustavovanými IETF, dnes jsou pracovní skupiny ustavovány pouze pro dohled nad samotným procesem standardizace a vytváření a předkládání návrhů je plně v režii soukromých firem, které považují za prestižní, pokud se některá technologie vyvinutá původně jako proprietární řešení (pouze pro zákazníky dané firmy či uživatele jejich výrobků) stane součástí veřejných standardů RFC.

### Vztah TCP/IP a Internetu

Vztah protokolové sady TCP/IP a Internetu je mnohostranný, avšak pro správné pochopení procesů, které se v oblasti TCP/IP a Internetu odehrávají, je tento vztah důležité pochopit. Protokolová sada TCP/IP je „technologie“, která vznikla v Internetu či přesněji spolu s ním, jak bylo popsáno výše. Přitom však použití TCP/IP není vázáno na Internet, dnes existuje mnoho sítí používajících technologii TCP/IP, které k Internetu nejsou připojeny. Protože však je Internet největší světovou sítí využívající TCP/IP, určuje vývoj v Internetu vývoj celého TCP/IP.

### Dohled nad fungováním Internetu

K tématu TCP/IP nepochybně patří i dohled nad fungováním Internetu, neboť způsob fungování Internetu ovlivňuje také fungování jiných TCP/IP sítí, zejména těch, které jsou nebo později budou s Internetem propojeny. Subjektem, který má fungování Internetu na starost, je organizace ICANN. Jde o sdružení podnikatelských, akademických i jiných subjektů z celého světa, převážně organizací, které v roce 1998 nahradilo organizaci IANA a převzalo její činnost.

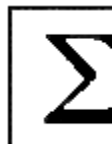
Hlavní činnosti, které ICANN zajišťuje, jsou:

- koordinace technické správy systému doménových jmen Internetu (DNS)
- koordinace přidělování IP adres z adresního prostoru
- koordinace přiřazování čísel protokolů (tzv. dobře známé porty)
- správa systému kořenových DNS serverů

Kromě těchto úkolů se ICANN zabývá i jinými otázkami a problémy dlouhodobějšího charakteru s cílem zachování a zlepšování svobodné konkurence na Internetu při zachování jeho provozní stability.

### Shrnutí

V této úvodní kapitole jsme se seznámili se vznikem protokolové sady TCP/IP, s její základní filosofií (otevřenost, snaha o jednoduchost), a dále s vrstevnatým



## Počítačové sítě

uspořádáním TCP/IP do 4 vrstev (vrstva síťového rozhraní, vrstva síťová, transportní v., aplikační v.) a jejich funkcemi. Rovněž již víme, jakým způsobem vznikají standardy TCP/IP a jakou mají formu, a kdo se stará o každodenní fungování Internetu.

### Kontrolní otázky

1. Co znamená princip „best effort“ a na jaké vrstvě se uplatňuje?
2. Jaké jsou hlavní přednosti TCP/IP?
3. Jaké má TCP/IP nedostatky?

### Pojmy k zapamatování:

- Princip přepojování paketů
- ARPANET
- Princip maximální snahy (best effort)
- Nespolehlivá nespojová komunikace
- 4 vrstvy TCP/IP
- RFC
- IETF
- ICANN

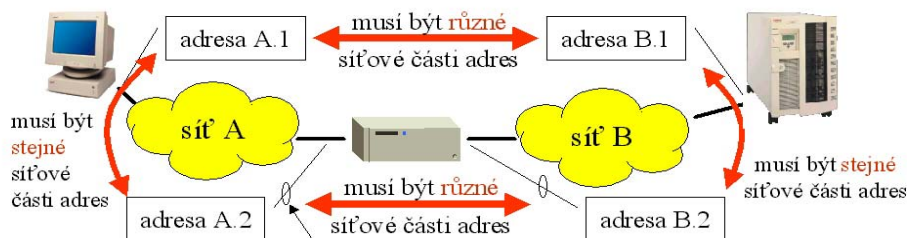
## 11 IP adresy

### 11.1 Struktura IP adres, třídy IP adres, distribuce adres

IP adresy jsou adresy používané na síťové vrstvě a vyšších vrstvách protokolové sady TCP/IP. K tomu, aby mohly sloužit svému účelu, kterým je identifikace uzlů v rámci celé sítě TCP/IP (např. celého Internetu), musí být zajištěna jedinečnost přidělených adres v rámci této sítě.

IP adresy v sobě nenesou žádnou explicitní informaci o adrese fyzické neboli linkové a proto se nijak navzájem neliší IP adresy, které patří uzlům připojenými k sítím s odlišnou přenosovou technologií. Velikost IP adresy je 32 bitů.

IP adresy jsou fyzicky jednoduté, přestože obsahují část označující adresu sítě, a část identifikující uzel v dané síti. Hranici mezi oběma částmi adresy tvoří určitá bitová pozice. Je přitom zřejmé, že pokud by bylo umístění této hranice pevně stanoveno, docházelo by k velkému plýtvání adresami, neboť hranice by zřejmě musela být nastavena tak, aby vyhovovala i velkým sítím, a proto by malé sítě, kterých je početně jistě velká většina, adresami velmi plýtvaly. Vzhledem ke koncepci IP adres totiž není možné, aby měly 2 různé sítě síťovou část adresy shodnou (viz obr. 11.1)



Obrázek 11.1: Přidělení IP adres v jednotlivých sítích

#### Příklad

Například při pevném rozdělení IP adresy na polovinu (16 bitů pro adresu sítě a 16 bitů pro adresu uzlu) by v síti o 1000 uzlech zůstalo nevyužito přes 64 tisíc adres (přesně 64 536), což je přes 98 procent.

#### Úkol k textu

Zkuste si sami spočítat, jak velká je ztráta adres při 5000 uzlech v síti.

Z výše uvedeného vyplývá, že hranice mezi síťovou částí a uzlovou částí IP adresy musí být do jisté míry pohyblivá. Původní koncepce IP adres počítala s

tím, že informace o tom, kde se zmíněná hranice mezi síťovou a uzlovou částí adresy nachází, ponese adresa určitým způsobem v sobě, a proto byly definovány 3 tzv. třídy adres:

- Adresy třídy A, které mají síťovou část dlouhou 8 bitů a uzlovou část 24 bitů, jsou určeny pro největší síť. Rozlišovacím znakem je, že jejich první bit je nulový.
- Adresy třídy B, které mají síťovou část dlouhou 16 bitů a uzlovou část rovněž 16 bitů, jsou určeny pro středně velké síť. Rozlišovacím znakem je, že jejich první bit má hodnotu 1 a druhý bit 0.
- Adresy třídy C, které mají síťovou část dlouhou 24 bitů a uzlovou část 8 bitů, jsou určeny pro malé síť. Rozlišovacím znakem je, že jejich první dva bity mají hodnotu 1 a třetí bit 0.

Upozorňujeme zde na to, že vzhledem k tomu, co bylo uvedeno výše, se IP adresy přidělují po celých blocích, a proto se často pod pojmem „adresa třídy ...“ míní celý blok adres se stejnou síťovou částí, tedy např. v případě adresy třídy C všech 256 možných adres, přestože jednotné číslo by naznačovalo, že se jedná o jednotlivou adresu.

### K zamyšlení

Zkuste si spočítat, jak velkou část adresového prostoru IP adres zabírají adresy třídy A, adresy třídy B a adresy třídy C. Proč myslíte, že je rozdělení určeno tak, že se všechny 3 třídy o adresní prostor nedělí rovnoměrně?

Pro snazší práci lidí s IP adresami (pro účely různých úkonů spojených s konfigurací sítě apod.) je třeba, aby si IP adresy bylo možné zapamatovat. Binární vyjádření není pro lidskou paměť (na rozdíl od paměti počítačové) vhodné, proto se obvykle používá jiný způsob zápisu IP adres. Ten je založen na tom, že se IP adresa rozdělí na 4 byty (skupiny po 8 bitech) a ty se převedou na desítkové číslo. Takto vzniklá 4 desítková čísla se zapíší vedle sebe oddělené tečkami.

### Příklad

IP adresa binárně vyjádřená číslem **110000010101000111111100000000** má symbolické vyjádření **192.168.255.0**.

### Úkol k textu

Převed'te binární IP adresu **00001010000000010000000100110111** na symbolický tvar.

### Distribuce adres

Vzhledem k požadavku jedinečnosti IP adres v celé IP síti je nutno ve velkých sítích zajistit distribuci adres. Tento problém je nejzávažnější v Internetu, neboť se jedná o nejrozsáhlejší IP síť obepínající dnes již prakticky celý svět. Musí proto existovat organizace, která bude přidělování IP adres řídit. Touto organizací je ICANN zmíněná v lekci 1. ICANN přiděluje IP adresy hierarchicky prostřednictvím pověřených regionálních organizací. Pro Evropu jde o organizaci RIPE. RIPE přiděluje bloky IP adres buď přímo (zpravidla nadnárodním poskytovatelům služeb Internetu, tzv. ISP), nebo prostřednictvím

národních organizací s působností v určitém státě.

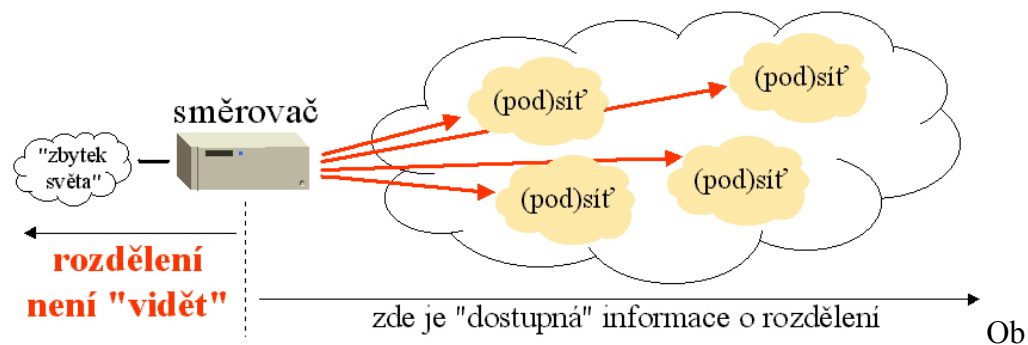
## 11.2 Omezený rozsah adresního prostoru

V důsledku nečekaně prudkého nárůstu uživatelů Internetu od počátku 90. let začalo došlo k tomu, že počet IP adres se začal jevit jako nedostatečný. Je zřejmé, že celkový počet možných kombinací IP adres (přes 4 miliardy) by byl ještě dlouho dostatečný, ale způsob distribuce a přidělování adres byl i přes rozdělení adres do tříd poměrně neefektivní. Proto byla hledána různá řešení tohoto problému.

Principiálním řešením by zřejmě bylo přechod na IP protokol verze 6, který kromě jiných zlepšení přináší prodloužení adresy na 128 bitů a tím mnohonásobné rozšíření adresního prostoru. Bohužel k přechodu na IP protokol verze 6 dosud nedošlo, zejména kvůli některým nedorozuměním otázkám týkajících se nutnosti změny software na velkém množství směrovačů ve vzájemné koordinaci.

### Subnetting

Z výše uvedených důvodů se dosud používají především dočasná řešení orientovaná na zefektivnění využívání stávajícího adresního prostoru. Jedním z takových řešení, které se používá zejména pro malé sítě, je vytváření tzv. podsítí (**subnetting**). Princip subnettingu spočívá v tom, že se hranice mezi sítíovou a uzlovou částí adresy posune směrem k nižším bitům (neboli se zvětší sítíová část adresy na úkor části uzlové) a díky tomu se může jedna síť rozdělit na několik menších sítí. Vzhledem k mechanismu rozdělení na podsítě založeném na posunu hranice mezi oběma částmi adresy, není možné provádět dělení libovolně, ale pouze po mocninách čísla 2. Protože ne všechny mechanismy pro práci s IP adresami nejsou na subnetting připraveny, bylo nutné pro tento účel zavést tzv. masky. Dělení na podsítě se provádí izolovaně, tedy tak, že informace o něm nejsou šířeny do zbytku sítě, jehož se toto rozdělení netýká. Schéma vytváření podsítí je zobrazeno na obr. 2.2.



rázek 2.2: Princip subnettingu

Smysl subnettingu spočívá v tom, že umožňuje využití jedné sítíové adresy (přesněji řečeno jedné skupiny adres určité třídy) pro více menších sítí. Bez subnettingu by bylo nutné pro každou takovou síť použít samostatnou adresu příslušné třídy, čímž by docházelo k plýtvání.

### Příklad

Například pro 4 sítě po 50 uzlech je možno využít díky subnettingu celkem jednu adresu třídy C rozdělenou na čtvrtiny. Bez subnettingu by bylo nutné





použít 4 adresy třídy C.

---

Je však nutno poznamenat, že subnetting lze použít pouze pro sítě navzájem blízké v tom smyslu, že mají **jediný společný vstupní bod**, který je propojuje se zbytkem sítě. Je to vynuceno tím, že informace o rozdělení do podsítí je lokalizována (není šířena dále do sítě). V případě více vstupních bodů by nebylo možné rozhodnout o tom, který se má použít.

### Privátní IP adresy

Dalším řešením je využití privátních IP adres. Privátní adresy se používají pro ty uzly v IP sítích, které nepotřebují přímo komunikovat s uzly mimo síť. Na první pohled se může zdát, že ve většině sítí takových uzlů mnoho nebude, neboť např. elektronickou poštou s obchodními partnery komunikuje dnes i mnoho řadových zaměstnanců v mnoha podnicích. Právě pro práci s elektronickou poštou jako zřejmě nejpoužívanější službou (spolu s WWW) však stanice přímou komunikaci se stanicemi mimo lokální síť nepotřebuje, neboť poštovní server, který je zpravidla umístěn uvnitř téže lokální sítě, veškerou komunikaci mimo síť realizuje sám, a stanice s klientem komunikuje pouze s tímto serverem. V případě služby WWW tomu tak není, ovšem zde lze zase snadno použít tzv. proxy serveru, který pro pracovní stanice uvnitř sítě službu zprostředkuje. Toto uspořádání se velmi často používá, protože mimo jiné umožňuje efektivně kontrolovat využívání služby WWW a také v menší míře přispívá ke zvýšení bezpečnosti sítě zneviditelněním jejích stanic zvenčí.

Privátní IP adresy se tedy mohou používat mnoha i poměrně rozsáhlých IP sítích, které pak vystačí s přidělením běžných (veřejných) IP adres pro několik málo uzlů. Je zřejmé, že základní podmínkou fungování privátních adres je zamezení šíření směrovacích informací ven ze sítě používající privátní adresy na jejích hranicích. Potom lze v takové síti jako privátní adresy použít adresy v zásadě libovolné. Přesto vzniklo doporučení, které rozsahy adres se mají používat jako privátní. Jedná se o tyto adresy:

- 1 adresu třídy A, konkrétně o rozsah 10.0.0.0 - 10.255.255.255;
- 16 adres třídy B, konkrétně o rozsah 172.16.0.0 - 172.31.255.255
- 256 adres třídy C, konkrétně o rozsah 192.168.0.0 - 192.168.255.255

Důvodem vzniku tohoto doporučení byla skutečnost, že ne vždy musí být zajištěno skutečné zamezení šíření směrovacích informací o vnitřku sítě s privátními adresami hned na hranici sítě (například kvůli nesprávně nakonfigurovanému směrovači). Pokud bude použito privátních adres z výše uvedených doporučených rozsahů, zastaví šíření nesprávně zaslaných paketů každý další směrovač, neboť má informaci o tom, že jde o privátní IP adresy, které nemá směrovat. V případě použití jiných adres jako privátních to pochopitelně neplatí.

### Network Address Translation (NAT)

Spolu s privátními adresami se často používá mechanismu NAT, který za chodu překládá adresy používané uvnitř sítě na adresy používané mimo síť. Je definován v RFC 1631.

### Classless InterDomain Routing (CIDR)

Dalším mechanismem pomáhajícím zpomalit úbytek IP adres, je mechanismus CIDR. Jde v zásadě o komplementární postup k subnettingu (z tohoto důvodu se také někdy označuje jako supernetting), který umožňuje, aby se sítím přidělovaly vždy vhodně velké rozsahy adres, neboť umožňuje posun hranice mezi síťovou částí adresy (nyní označovanou jako „prefix“) a uzlovou částí adresy. Prakticky tedy CIDR nahrazuje systém rozdělení IP adres do tříd A, B a C.

CIDR je založen na agregaci sousedních adres (ve smyslu podobnosti čísel, přesněji shodnosti nejvyšších bitů) ve směrovacích tabulkách. Nutnou podmínkou pro použití mechanismu CIDR je, aby sousední adresy byly přiděleny sousedním sítím ve smyslu hierarchie připojení. To si vynucuje, aby se do přidělování adres zapojili poskytovatelé, kteří jediné jsou schopni toto zajistit.

### Shrnutí

V této lekci jsme se seznámili se základními vlastnostmi IP adres, jejich rozdělením do tříd a prostředkům dočasného omezení jejich vyčerpávání.

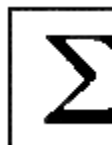
---

### Úkoly k textu

1. Napište libovolnou IP adresu třídy A.
  2. Napište masku podsítě pro IP adresu třídy C.
- 

### Korespondenční úkoly

1. Zjistěte si (při aktivním připojení k Internetu) IP adresu svého počítače (Win9x/ME – utilita winipcfg, Win NT/2000/XP – utilita ipconfig), napište ji a zařaďte ji do příslušné třídy adres.
2. Rozmyslete si, zda potřebuje Vaše PC, které je připojeno do lokální sítě a používá pouze elektronickou poštu, přímo komunikovat s Internetem (tedy: potřebuje IP adresu)? Svou odpověď zdůvodněte.

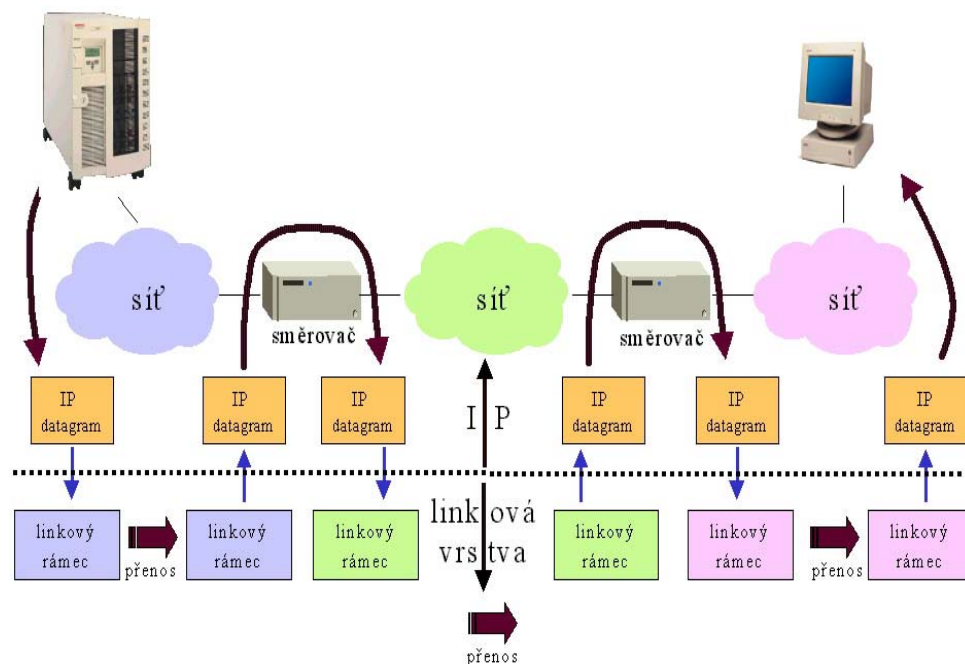


## 12 IP protokol, vlastnosti síťové vrstvy

### 12.1 IP protokol

IP protokol je jediný přenosový protokol síťové vrstvy v architektuře TCP/IP. Jak již jsme se zmínili dříve, jedná se o přenosový protokol univerzální, který je schopen fungovat nad téměř libovolnou přenosovou technologií používanou ve vrstvě síťového rozhraní. Proto nevyužívá specifika jednotlivých přenosových technologií a požaduje od nich pouze služby na společné minimální úrovni kvality, tedy nespojové a nespolehlivé.

Protokol IP je tedy nespolehlivý a nespojovaný. Pracuje s proměnlivou délkou paketu (používá se též termínu „datagram“). Protože jde o univerzální přenosový protokol síťové vrstvy, je implementován ve všech uzlech, tedy ve směrovačích i v hostitelských počítačích. V současné době se používá jeho verze 4. Existuje návrh verze 6, o němž jsme se zmínili v minulé lekci v souvislosti s rozšířením adresního prostoru, ten se však dosud nepoužívá. Fungování protokolu IP je znázorněno na obrázku 12.1.



Obr 12.1: Fungování protokolu IP

Protokol IP v rámci svého fungování rozhoduje o volbě směru pro další zaslání paketu, zajišťuje předávání paketů vrstvě síťového rozhraní pro odeslání.

#### Formát IP paketu (IP datagramu)

Jak již bylo zmíněno výše, délka datagramu je proměnná. Maximální délka je 64 kB, tedy 65536 bytů. Tuto maximální hodnotu si většina sítí snižuje podle toho, jakou maximální velikost rámce jim umožňuje zasílat vrstva síťového rozhraní. Minimální hodnota maximální velikosti paketu je 576 bytů, což odpovídá nejméně 512 bytům dat přenášeným v paketu.

Paket obsahuje hlavičku, která má rovněž proměnlivou délku, její minimální délka však je 20 bytů. Nejvýznamnější položky hlavičky jsou:

- údaj o délce hlavičky a délce paketu (v prvních 4 bytech);
- identifikační číslo paketu (slouží pro potřeby fragmentace, nikoli pro číslování pořadí paketů), a příznaky pro fragmentaci. O fragmentaci se dozvíme dále.
- životnost paketu (položka TTL), označení přenášeného protokolu, kontrolní součet hlavičky
- IP adresu odesílatele a IP adresu příjemce

Hlavička může být doplněna o další nepovinné položky a vždy je doplněna tzv. výplní na délku dělitelnou 32 bity.

### Fragmentace

Jak již bylo uvedeno výše, je délka IP paketu proměnlivá, a maximální délka IP paketu může být v různých sítích různá. Proto se nelze vyhnout situaci, kdy na směrovač přijde paket o velikosti, která je větší než maximální velikost přenesitelná v rámci sítě určené směrovačem jako odchozí. Potom má směrovač 2 možnosti: buď paket rozdělit na části (tzv. fragmentovat), nebo jej zahodit (to učiní pouze tehdy, pokud je příznakem v hlavičce zakázána fragmentace daného paketu).

Fragmentace probíhá tak, že směrovač paket rozdělí do více paketů, které budou mít stejné identifikační číslo, a lišit se budou pouze datovou částí a údajem v hlavičce, který se nazývá OFFSET a stanoví posun počátku datové části paketu od počátku datové části původního paketu. Kromě toho bude v hlavičce všech fragmentů kromě posledního fragmentu nastaven příznak MORE FRAGMENTS označující, že paket je fragmentován a že ještě následují další fragmenty.

Skládání fragmentů do paketu pak provádí až cílová stanice. Proto nelze vyloučit situaci, kdy bude potřeba již fragmentovaný paket dále fragmentovat. Jak je vidět z popisu fragmentace v předchozím odstavci, nemělo by to způsobit žádné problémy. Jediným problémem, který je třeba ošetřit, je situace, kdy na cílový uzel některý fragment nedorazí (resp. nedorazí včas). V takovém případě uzel zahodí všechny fragmenty (a vygeneruje zprávu ICMP - viz dále).

## 12.2 Protokol ICMP

Protokol IP může v některých případech (např. při přetížení směrovače apod.) některé pakety zahodit. Přestože je definován jako nespolehlivý, snaží se o této situaci zpravidla informovat odesílatele. K tomu slouží protokol ICMP (Internet Control Message Protocol). Protože je IP protokol univerzálním přenosovým protokolem síťové vrstvy, přenášejí se zprávy protokolu ICMP vložené do IP paketů.

Zahození paketů nesoucích ICMP zprávu se však již nesignalizuje, protože by hrozilo zahlcení sítě oznámeními o zahození. Obecné pravidlo přitom zní, že ICMP zprávy se negenerují v případě, že je zahozen paket z důvodu chybného kontrolního součtu hlavičky. Důvod je prostý: adresa odesílatele paketu (a tedy příjemce případně vygenerovaného ICMP paketu), která je součástí hlavičky, v tomto případě nemusí být spolehlivá, neboť chybný kontrolní součet mohla způsobit právě její změna, ke které cestou

došlo.

Protokol ICMP definuje vlastní paket, který se vkládá do IP paketu. Jeho formát je jednoduchý: 4 bytová hlavička, která obsahuje v položce TYPE typ ICMP zprávy (jeden z předdefinovaných označených určeným číslem), upřesňující položku CODE a kontrolní součet paketu. Dále následuje datová část, která obvykle nese část původního paketu.

Nejvýznamnější situace, které jsou protokolem ICMP signalizovány, jsou popsány v následujících odstavcích.

### Time exceeded

Zpráva ICMP Time exceeded označuje zacyklení paketu. To je zjištěno podle položky životnosti (TTL) v hlavičce paketu, která se snižuje o 1 při každém průchodu přes směrovač (u odesílatele je TTL nastavena na implicitní hodnotu, která bývá 64, 128, případně i jiná). Pokud hodnota položky TTL na některém směrovači dosáhne nuly, paket se zahodí a směrovač vygeneruje zprávu ICMP Time exceeded, kterou odešle odesílateli zahozeného paketu. V tomto případě má položka TYPE hodnotu 11 a CODE hodnotu 0.

Druhou situací, kdy se signalizuje paketem ICMP Time exceeded, je situace, kdy během nastavené prodlevy pro čekání na všechny fragmenty některý fragment fragmentovaného paketu nedorazil k příjemci. Pak se zahodí všechny fragmenty a uzel vygeneruje zprávu ICMP Time exceeded, kterou odešle odesílateli zahozeného paketu. V tomto případě má položka TYPE hodnotu 11 a CODE hodnotu 1.

Pomocí zpráv ICMP Time exceeded je zpravidla realizována i utilita **traceroute**, která se používá ke zjištění aktuálně používané cesty k určitému uzlu. Využívá zasílání paketu s ICMP zprávou s TYPE=30 a TTL, kterou postupně zvyšuje od jedné až do dosažení cílového uzlu.

### Destination unreachable

Zpráva ICMP Destination unreachable signalizuje další situace, kdy byl zahozen paket. Zejména jde o situace nedostupné sítě, nedostupného uzlu, neexistující adresy či portu, překročení maximální velikosti paketu při zakázané fragmentaci atd. Tyto situace se opět odlišují různými hodnotami položky CODE.

Další situací, kdy je signalizováno zahazování paketů, je zpráva ICMP **Source quench**. Touto zprávou signalizuje směrovač odesílateli, že je zahlcen a musí proto zahazovat jeho pakety.

Dalším typem ICMP zprávy je dvojice ICMP **Echo request** a ICMP **Echo reply**. Ty se používají k diagnostickým účelům (zejména ke zjištění doby odezvy určitého uzlu a počtu přechodů přes směrovače na cestě k němu). Využívá jich například utilita ping.

S dalšími druhy ICMP paketů se seznámíme v následující části věnované směrování.

Na závěr musíme ještě poznamenat, že v praxi se z důvodů snah o zamezení šíření informací o uspořádání sítě např. v rámci autonomního systému mnohdy zasílání ICMP zpráv potlačuje.

## 12.3 Rozpoznávání adres

Protože IP adresy jsou abstraktní, tedy nemají žádnou souvislost s fyzickými adresami používanými na vrstvě síťového rozhraní, je nutné mít k dispozici mechanismus, který umožňuje tyto 2 druhy adres mezi sebou převádět. Především je nutný převod z IP adresy na adresu fyzickou, neboť ten je třeba při každém odeslání paketu. Vrstva síťového rozhraní totiž musí dostat požadavek k zaslání paketu na určitý uzel identifikovaný fyzickou adresou, neboť s IP adresami neumí pracovat. Opačný převod je třeba v některých speciálních případech.

Způsob, který se v dané (lokální) síti použije pro rozpoznávání adres, závisí na vlastnostech vrstvy síťového rozhraní, tedy na použité přenosové technologii. Nejčastěji se používá decentralizované zjišťování pomocí dotazu. Protože se však dotaz musí zaslat všem stanicím na síti, musí mít vrstva síťového rozhraní možnost zasílat všesměrové rámce (tzv. broadcasty).

Pro Ethernet se používá protokolu ARP, který používá právě dotazování. Funguje tak, že dotazující uzel vloží do všesměrového rámce dotaz na fyzickou (MAC) adresu uzlu, který má danou IP adresu. Ten uzel, který svou adresu pozná, odpoví vložením své MAC adresy do rámce a jeho odesláním zpět, tentokrát již pouze dotazující stanici.

Vzhledem k tomu, že překlad adres je třeba při každém odeslání paketu, není efektivní pokaždé provádět skutečné dotazování. Proto je do protokolu ARP zabudováno cacheování. Znamená to, že výsledky ARP dotazů jsou po určenou dobu (typicky cca 20 minut) na uzlech uloženy do vyrovnávací paměti (cache) a při každém požadavku na překlad adresy se nejprve ověří, zda není informace o adrese z předchozích dotazů uložena zde. Teprve v případě, že zde není adresa nalezena, se provede skutečný dotaz.

## 12.4 Směrování

Směrování je činnost, při níž je třeba určit nejvhodnější cestu po síti od odesílatele k příjemci podle předem určeného kritéria. Jeho součástí kromě výpočtu optimální cesty je i získávání, šíření a uchovávání směrovacích informací nutných k určení cesty.

Dva základní druhy směrování jsou:

- směrování **statické**, kdy se směrovací informace mění pouze ručně, nikoli automaticky;
- směrování **dynamické**, kdy je zajištěna automatická aktualizace informací potřebných k určení trasy.

Podle kritéria způsobu určení trasy rozlišujeme:

- směrování zdrojové (celou trasu určuje odesílatel);
- směrování skokové (trasa se určuje průběžně, na každém směrovači se upřesňuje).

V TCP/IP sítích se převážně používá skokové dynamické směrování, pouze ve speciálních případech též směrování statické.

Existují 2 základní algoritmy dynamického směrování:

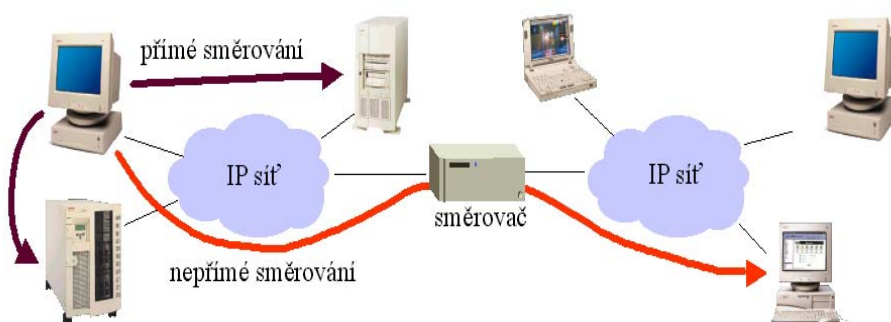
- směrování na základě vektoru vzdáleností (**vector-distance routing**);
- směrování na základě stavu linky (**link-state routing**).

Směrování provádějí všechny uzly v síti, tedy hostitelské počítače (stanice, servery apod.) i směrovače. Odlišnost spočívá pouze v tom, že směrovače se aktivně účastní i aktualizace směrovacích informací, kdežto hostitelské počítače pouze pasivně (tyto informace pouze přijímají).

Směrování se musí provést při každém odeslání paketu ještě před předáním požadavku na odeslání paketu vrstvě síťového rozhraní. Základní rozlišení, které se přitom musí provést, je rozlišení, zda se jedná o směrování přímé nebo nepřímé.

O **přímé směrování** se jedná tehdy, když je odesílatel i příjemce paketu ve stejné síti, tedy když mají shodnou síťovou část IP adresy neboli prefix. V takovém případě není třeba rozhodovat o volbě směru, stačí pouze požádat vrstvu síťového rozhraní o doručení paketu na odpovídající fyzickou adresu zjištěnou pomocí mechanismu rozpoznávání adres.

V případě, že je paket určen příjemci v jiné síti než kde se nachází odesílatel, jedná se o **nepřímé směrování**. V tom případě již je nutné určit odchozí směr, přesněji řečeno IP adresu odchozího směrovače. tento směrovač (resp. jeho jedno síťové rozhraní) se nachází ve stejné síti jako odesílatel, a tím se případ nepřímého směrování převede opět na směrování přímé. Schematické znázornění obou případů je znázorněno na obrázku 12.2



Obr 12.2: Přímé a nepřímé směrování

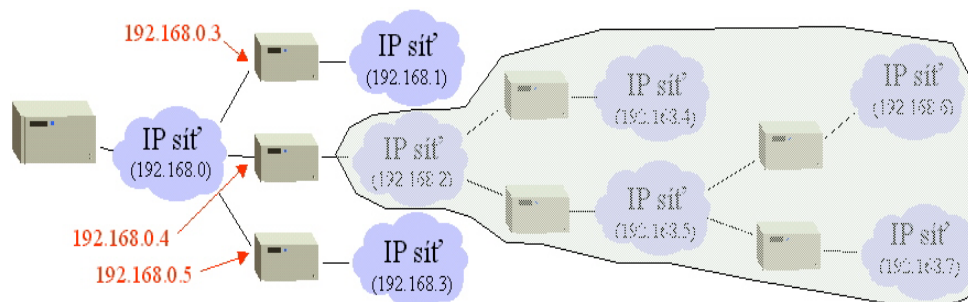
### Směrovací tabulky

Původní koncepce směrovacích tabulek byla taková, že každý směrovač bude mít (nejméně) jeden záznam pro každou síť, do které posílá pakety (v praxi to jsou většinou všechny připojené sítě, protože jen těžko lze a priori vyloučit komunikaci s některou ze sítí). To by ovšem vedlo k nutnosti práce s obrovským množstvím dat na každém směrovači, nehledě na jejich údržbu. Vzhledem k tomu, že ve směrovací tabulce se vždy nachází pouze odchozí směr pro každou síť, lze nutný objem směrovací tabulky výrazně zmenšit.

O jednom ze způsobů optimalizace směrovacích tabulek jsme se již zmínili v předchozí kapitole. Jde o mechanismus CIDR, který umožňuje sdružování sousedních adres do bloků. Nejeefektivnější a zřejmě nejčastěji

## Počítačové sítě

používaný způsob je však definice implicitního odchozího směru. V případě, že určíme, že pakety pro všechny sítě s výjimkou některých konkrétních sítí, které jsou z hlediska připojení zpravidla „blízko“, mají být posílány přes jeden směrovač (zpravidla ten, který má nejpřímější spojení s nadřazenými směrovači až k páteřním spojům v Internetu), stačí ve směrovacích tabulkách definovat explicitně těch několik výjimek a ostatní záznamy sdružit pod implicitní cestu (**default route**). To ilustruje obrázek 12.3 spolu s tabulkou 12.1.



Obr 12.3: Směrování s implicitní cestou

cílová síť/prefix	posílej přes
192.168.0/24	směruj přímo
192.168.1/24	192.168.0.3
192.168.3/24	192.168.0.5
všechno ostatní	192.168.0.4

Tabulka 12.1: Směrování s implicitní cestou – směrovací tabulka

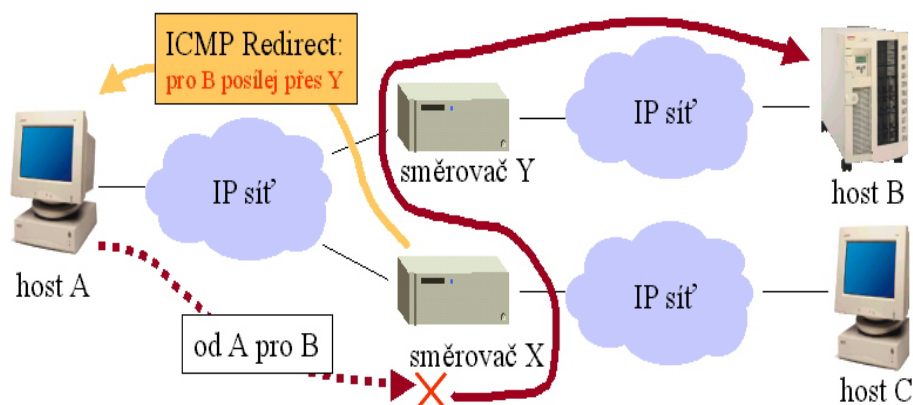
Existuje i opačná možnost definovat pro určitý konkrétní uzel specifickou cestu (tzv. host-specific route), to se však používá pouze ve speciálních případech. Díky algoritmu výběru směrovacích informací z tabulky, kdy přednost má vždy informace s nejvyšším počtem shodných bitů (odleva, tedy od nejvyšších bitů) mezi cílovou adresou a prefixem uvedeným ve směrovací tabulce se takové host-specific route uplatní, i když je pro ostatní uzly v téže síti definována třeba jiná cesta. V případě, že není definována implicitní cesta a v tabulce se nenajde odpovídající záznam, skončí směrování chybou a vygeneruje se hlášení ICMP Destination unreachable.

Další ICMP zprávy, které se směrováním úzce souvisejí, jsou:

- **ICMP Router solicitation** (dotaz na všechny směrovače, zasílá se IP broadcastem do celé sítě);
- **ICMP Router advertisement** (směrovač jím oznamuje svou existenci, buď jako odpověď na ICMP Router solicitation, nebo samostatně);
- **ICMP Redirect**, které se zasílá odesílateli paketu, pokud směrovač zjistí, že existuje vhodnější cesta pro daný paket, než kterou ji odesílatel zaslal. Odesílatel by si pak měl příslušným způsobem upravit odpovídající záznam ve své směrovací tabulce. Samotný paket se samozřejmě nevrací odesílateli, ale předá se dalšímu směrovači, který je pro odeslání vhodnější (viz obrázek 12.4).

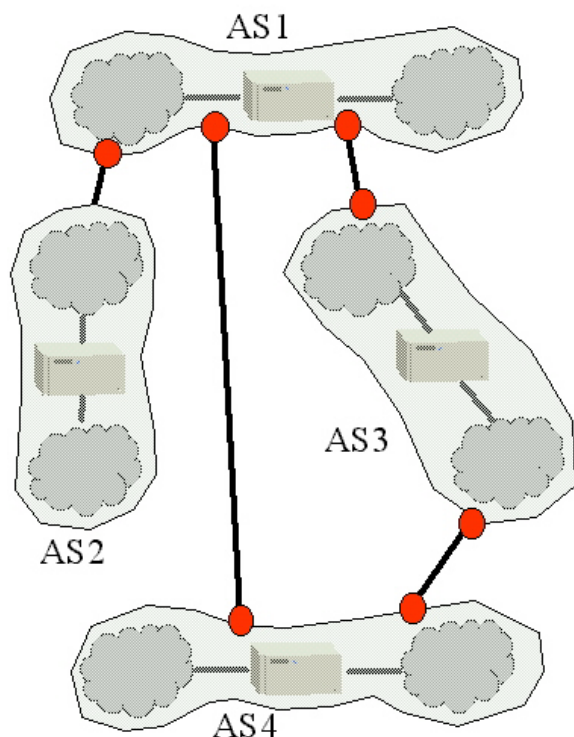


## Počítačové sítě



Obr 12.4: Použití ICMP Redirect

Správné směrování závisí na šíření směrovacích informací, což je při dnešním rozsahu Internetu značně obtížný problém. Vzhledem k tomu, že vnitřní směrovací informace o samostatných částech sítě s jedním vstupním bodem není třeba do zbytku sítě předávat, došlo na základě této myšlenky k rozdělení Internetu na větší množství tzv. **autonomních systémů**. Tím došlo k výraznému zmenšení objemu směrovacích informací, které je třeba po síti přenášet. Později bylo umožněno i to, aby autonomní systémy měly i více než jeden vstupní bod, takže jejich struktura může vypadat např. tak, jak je uvedeno na obr. 12.5



Obr 12.5: Autonomní systémy v Internetu

Touto změnou byl umožněn tzv. peering, neboli propojení sítí různých poskytovatelů připojení (nebo obecněji libovolných provozovatelů sítí) mezi sebou na úrovni nižší než přes páteřní spoje Internetu. V důsledku toho již nedochází k tomu, že např. paket zasílaný ze sítě jednoho poskytovatele v Ostravě do sítě jiného poskytovatele rovněž v Ostravě putoval až do páteřní

sítě a pak zpět, jak tomu často bývalo v minulosti.

### Shrnutí

V této kapitole jste se seznámili s vlastnostmi síťové vrstvy TCP/IP. Zejména je důležité si zapamatovat, že zde pracuje jediný přenosový protokol IP, který zakrývá specifika různých přenosových technologií pracujících na vrstvě síťového rozhraní. Dále je třeba si uvědomit, význam protokolu ICMP a klíčový význam směrování pro fungování rozsáhlých sítí na bázi TCP/IP.

---

### Kontrolní otázky:

1. Jaké jsou základní vlastnosti protokolu IP?
  2. V jakých situacích se používá protokol ICMP?
  3. Jaký protokol se používá k přenosu chybových a dalších hlášení protokolu ICMP?
  4. Co je to fragmentace paketů a kdy k ní může dojít? Kde se řeší její důsledky?
  5. K čemu slouží protokol ARP a jak pracuje?
  6. Jaký je rozdíl mezi směrováním, které provádí uzel (stanice), a tím, které provádí směrovač?
  7. Jaké druhy ICMP paketů generují směrovače při směrování?
- 

### Korespondenční úkoly

1. Na počítači připojeném k Internetu vyhledejte utility ping a traceroute (ve Windows se nazývá tracert!) a vyzkoušejte si jejich použití. Např. ve Windows musíte spustit příkazový řádek a v něm zadat např. „ping www.osu.cz“ resp. „tracert www.osu.cz“). Výsledek uložte do souboru a pošlete tutorovi. Pokud nebudete úspěšní, popište tutorovi co nejpřesněji verzi Vašeho operačního systému a požádejte jej o radu.
2. Použije se při odesílání zprávy pomocí elektronické pošty přímé nebo nepřímé směrování? Odpověď zdůvodněte!
3. Představte si IP síť s 1000 zhruba stejně velkých sítí (do 250 uzlů) a vypočtěte, kolik záznamů by musela mít směrovací tabulka v každém směrovači v této síti, pokud by nebyla použita optimalizace směrovacích tabulek. Předpokládejte přitom, že každá síť je připojena pouze k jedné další síti (tedy má jeden odchozí směrovač). Mezilehlé směrovače (přímo nepřipojené k žádné ze sítí) můžete zanedbat.

## 13 Protokoly transportní vrstvy

### 13.1 Funkce transportní vrstvy

Obecnou funkcí transportní vrstvy je přizpůsobovat možnosti vrstev nižších (reprezentovaných službami síťové vrstvy) požadavkům vrstev vyšších. Konkrétně v TCP/IP jde především o přizpůsobení nespolehlivé a nespojované služby protokolu IP (síťové vrstvy) častému požadavku mnoha aplikací na spolehlivý spojovaný přenos. Těmto požadavkům se umí transportní vrstva TCP/IP přizpůsobit, přitom však dává aplikacím možnost volby, takže ty aplikace, které požadují spolehlivost, použijí protokol TCP, kdežto ty, které preferují rychlost a spolehlivou službu nepotřebují, použijí protokol UDP.

Naopak požadavku na garanci určitého maximálního zpoždění přenosu či garanci jiného parametru přenosu dosud transportní vrstva TCP/IP vyhovět neumí. Přitom tyto požadavky se stále častěji objevují spolu s rostoucími požadavky uživatelů a aplikací na přenosy multimediálních dat. Na řešení tohoto problému se pracuje, avšak protože jde o značný zásah do filosofie TCP/IP, uspokojivé řešení zachovávající výhody TCP/IP dosud neexistuje.

Vedle přizpůsobení je další funkcí transportní vrstvy rozlišování paketů či datových toků podle jejich příslušnosti k určitým procesům v aplikační vrstvě. Tato funkce se nazývá multiplexování a demultiplexování. Bez ní by nebylo možné například provozovat (mít spuštěný) na stanici současně program pro práci s elektronickou poštou a prohlížeč WWW, což by dnes, v době nadvlády víceúlohových operačních systémů na stanicích bylo jistě nepohodlné. Ještě horší by bylo, že na jednom fyzickém serveru by nemohl běžet proces sloužící jako WWW server zároveň např. s FTP serverem, a dokonce by ani jeden WWW server nemohl zároveň komunikovat s více uživateli. K tomuto rozlišení se v transportní vrstvě TCP/IP používají tzv. **porty**.

Port je přechodovým bodem mezi aplikační vrstvou a transportní vrstvou, k nimž se mohou aplikace podle potřeby asociovat. Přitom jedna aplikace může využívat více portů, ale pochopitelně jeden port nesmí být současně používán více aplikacemi.

Porty jsou identifikovány svými čísly, jejichž tvar je nezávislý na platformě (vždy se jedná o celá kladná čísla). Tato čísla představují relativní adresu v rámci uzlu. Programy obvykle přistupují k portům prostřednictvím tzv. socketů, které jsou součástí příslušného programátorského rozhraní (API).

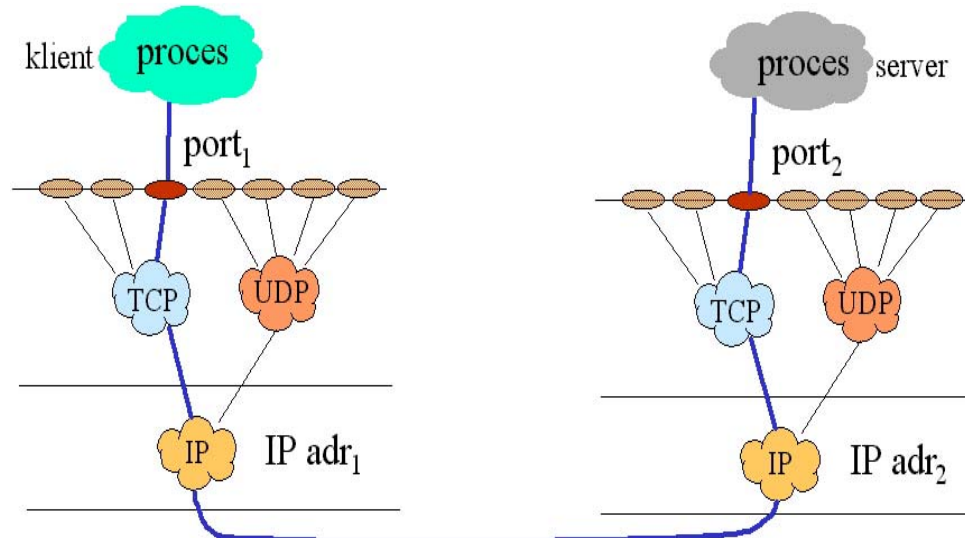
Význam některých portů je pevně dán, neboť je přidělila IANA (předchůdce ICANN) příslušným aplikacím. Jde o tzv. dobře známé porty (v rozsahu 1-1023), na nichž jsou poskytovány standardní služby. Jejich přidělení je uvedeno v RFC 1700, později je přidělení aktualizováno pouze on-line.

Z výše uvedeného tedy vyplývá, že spojení mezi aplikacemi na dvou uzlech je definováno pěticí údajů, tedy:

- transportní protokol (TCP nebo UDP);
- IP adresa odesílatele;
- port odesílatele;

- IP adresa příjemce;
- port příjemce.

Aplikační spojení je znázorněno na obrázku 13.1.



Obr 13.1 Představa aplikačního spojení

## 13.2 Typy služeb transportní vrstvy

Jak bylo uvedeno výše, transportní vrstva dává aplikaci vybrat, který typ služby bude používat. Proto jsou na transportní vrstvě definovány 2 odpovídající typy služby a ještě jeden speciální typ. Jde o typy:

- **stream**, kdy transportní vrstva vytváří iluzi bytového proudu. V tomto případě jsou data přijímána a vydávána po bytech, členění na bloky je pro aplikaci transparentní, tedy se provádí pouze interně pro potřeby přenosu. Přenos je spolehlivý, je garantováno pořadí dat, do služby je začleněno řízení toku. Tuto službu nabízí protokol TCP.
- **datagram**, kdy transportní služba vytváří iluzi blokového přenosu. Data jsou členěna do bloků (datagramů), přenášena jsou nespolehlivě, bez garance pořadí, ztrát či duplicit. Tato služba neobsahuje řízení toku. Tuto službu nabízí protokol UDP.
- **raw**, což je speciální režim umožňující přímý přístup ke službám nižších vrstev. Používá se pro testovací účely, pro utility typu PING apod.

## 13.3 Protokoly transportní vrstvy

### Protokol UDP

Protokol UDP představuje prakticky pouze velmi jednoduchou nadstavbu nad protokolem IP. Nemění charakter jeho služeb, neboť UDP poskytuje nespolehlivou a nespojovou službu. Navíc zajišťuje pouze multiplexing/demultiplexing. Definice protokolu UDP zahrnuje možnost vytváření kontrolního součtu celého paketu, avšak jedinou reakcí tohoto protokolu při doručení paketu s kontrolním součtem je zahození paketu. Vytváření kontrolního součtu lze vypnout.

## Počítačové sítě

Protokol UDP používají ty aplikace, které potřebují co nejrychlejší doručení dat a nejsou přitom závislé na tom, zda data budou doručena všechna. Vyšší rychlost UDP proti TCP je dána tím, že UDP není zatížen režii spojenou s vytvářením a rušením spojení, potvrzování doručení a dalšími mechanismy zabezpečení.

Vlastnosti UDP již byly zmíněny výše, proto jen zopakujeme. Jde o protokol:

- nespolehlivý;
- nespojový;
- vytvářející iluzi blokového přenosu (maximální velikost bloku je  $2^{16}$ -20-8 bytů; první údaj je max. velikost IP paketu, 20 je minimální délka hlavičky IP paketu a 8 je délka hlavičky UDP datagramu, jak uvidíme za chvíli).

Protokol UDP může být použit i pro rozesílání všesměrových (broadcast) paketů nebo paketů pro více příjemců (multicast), což u TCP není možné (protože vytváří spojení, které nemůže spojit více než 2 body). Komunikace pomocí protokolu UDP je bezstavová, takže si komunikující strany nemusí pamatovat předchozí historii komunikace a mohou kdykoli (přesněji řečeno po odeslání nebo přijetí kteréhokoli UDP datagramu) komunikaci přerušit nebo v ní znovu pokračovat.

UDP datagram je velmi jednoduchý. Zahrnuje pouze 8-bytovou hlavičku obsahující zdrojový port, cílový port, délku datagramu a kontrolní součet (jak bylo zmíněno výše, jeho generování je volitelné. Kontrolní součet se zde generuje z celého UDP datagramu (hlavičky i dat) a navíc ještě z tzv. **pseudohlavičky**. Pseudohlavička je virtuální struktura, která nikde reálně neexistuje, a používá se právě jen pro generování kontrolního součtu. Obsahuje ve 12 bytech nejdůležitější údaje z hlavičky IP paketu, především zdrojovou a cílovou IP adresu. Pokud se kontrolní součet generuje, kontroluje se při doručení datagramu a v případě neshody vypočteného součtu s údajem v hlavičce se datagram zahodí bez jakéhokoli oznámení odesílateli (na rozdíl od IP protokolu).

### Protokol TCP

Protokol TCP poskytuje službu spojovaného charakteru. Znamená to, že pro přenos dat se nejprve ustaví spojení mezi odesílatelem a příjemcem (vždy jedinými, tedy vždy se jedná o dvoubodové spojení), po něm se přenesou data a po přenosu dat se spojení ukončí. Standardně jde o spojení duplexní, tedy obou směrné. Služba protokolu TCP vytváří pro aplikace iluzi bytového toku. Zajišťuje plnou spolehlivost a řízení toku, při kterém se odesílatel snaží přizpůsobit schopnostem příjemce.

Spojovaná služba protokolu TCP je pouze iluzí pro aplikaci, protože IP protokol na síťové vrstvě funguje pochopitelně stále nespojovaně. Proto musí protokol TCP ošetřit všechny stavy, k nimž může na síťové vrstvě dojít, jako např. restart uzlu, ztrátu dat způsobenou nespolehlivostí přenosové infrastruktury, změnu pořadí dat apod. Mezilehlé uzly (směrovače) o protokolu TCP nevědí, protože funguje až na transportní vrstvě.

Spolehlivost v protokolu TCP je zajišťována především pomocí techniky tzv. kontinuálního potvrzování. Příjemce generuje po přijetí TCP paketu (který

se obvykle nazývá **TCP segment**) kladná potvrzení. Odesílatel monitoruje dobu obrátky (tedy dobu od odeslání TCP segmentu do přijetí potvrzení) a podle váženého průměru doby obrátky a jejího rozptylu počítá dobu, po kterou čeká na potvrzení. Výsledkem je, že doba, po kterou se čeká na potvrzení, je o něco vyšší než průměrná doba obrátky, přičemž velikost zvýšení čekací doby nad průměrnou dobu obrátky je úměrná rozptylu doby obrátky.

Tento postup vcelku uspokojivě reaguje jak na prodlužování doby obrátky, tak na její zkracování, a to bez ohledu na skutečnou velikost průměrné doby obrátky, která je pochopitelně vyšší v rozlehlých sítích a výrazně nižší naopak v lokálních sítích.

Protokol TCP používá nesamostatného potvrzování, pro které se používá určené pole v hlavičce protisměrného TCP segmentu. Tento způsob potvrzování se nazývá piggybacking.

Jak bylo uvedeno dříve, vytváří služba protokolu TCP pro aplikaci iluzi bytového proudu, tedy aplikace předává data protokolu TCP po bytech. TCP protokol si tato data sám „bufferuje“ neboli seskupuje do skupin odpovídajících množství volného místa a po naplnění velikosti bufferu (jehož velikost závisí na parametru MTU). Aby bylo možné data odeslat v případě potřeby okamžitě (např. při předávání dat ve formě souboru po dosažení jeho konce), má aplikace možnost vyžádat si okamžité odeslání obsahu bufferu i před jeho úplným naplněním.

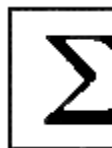
Vzhledem k tomu, že protokol TCP nepracuje s bloky, musí zajistit číslování pozice v bytovém proudu. K tomu se používá 32-bitové číslo. Počáteční hodnota se volí náhodně, tedy neplatí, že pozice prvního byte má číslo 1.

Protokol TCP řídí tok dat tak, aby odesílatel nezahlcoval příjemce a nedocházelo kvůli tomu ke ztrátě dat. Toho se dosahuje použitím tzv. metody okénka. Okénko udává velikost volného místa pro příjem dat. Tato velikost okénka je signalizována spolu s každým potvrzením o přijetí TCP segmentu. Odesílatel vysílá jen tolik dat, kolik odpovídá aktuálně volnému okénku, a vysílání obnoví až tehdy, když mu bude příjemce signalizovat zvětšení volného okénka.

V případě, že během zasílání řady TCP segmentů dojde ke ztrátě některého ze segmentů, a tedy odesílatel neobdrží potvrzení doručení tohoto segmentu, přejde odesílatel z kontinuálního potvrzování na potvrzování jednotlivých segmentů. To znamená, že odesílatel odesílá jednotlivé segmenty až po obdržení potvrzení doručení předchozího segmentu, namísto aby odesílal tolik segmentů, kolik odpovídá velikosti volného okénka. Ke kontinuálnímu potvrzování přechází odesílatel postupným zdvojnásobováním množství odesílaných dat a v případě včasného doručení potvrzení se takto pokračuje až do dosažení velikosti volného okénka.

### Shrnutí

V této kapitole jste se seznámili s vlastnostmi transportní vrstvy. Zejména je důležité si zapamatovat, že zde jsou k dispozici dva alternativní přenosové protokoly, TCP a UDP, které se liší svými vlastnostmi. Protokol TCP poskytuje spolehlivé spojové služby, zatímco protokol UDP služby nespolehlivé nespojové. Mezi protokoly TCP a UDP si mohou aplikace vybírat



ten, který jim lépe vyhovuje.

---

### Kontrolní otázky:

1. Proč je nutné pro různé služby použít různé porty transportní vrstvy?
  2. K čemu v praxi slouží socket?
  3. Jaké vlastnosti má protokol UDP?
  4. Na jakém principu pracuje protokol TCP?
  5. Jaké mechanismy používá TCP protokol k zajištění spolehlivosti?
  6. Co znamená použití 3fázového handshake protokolem TCP?
- 

### Korespondenční úkol:

Posuďte na základě vlastností, které má služba FTP, zda by měla používat protokol TCP nebo UDP.

## 14 Služby aplikační vrstvy

### 14.1 Vlastnosti aplikací v TCP/IP

Aplikace v TCP/IP jsou založeny na modelu klient/server. Znamená to, že jejich funkce je rozdělena mezi klientskou část, která se zpravidla spouští na pracovní stanici, a serverovou část, která je v provozu na určitém konkrétním aplikačním serveru. Součástí aplikační vrstvy jsou pouze ty části aplikací, které jsou nutné pro fungování určité služby, nikoli však uživatelské rozhraní. V případě klientské části pro elektronickou poštu jsou součástí aplikační vrstvy funkce pro odesílání zpráv a jejich příjem, ne však např. funkce pro správu složek apod. Standardizovány jsou pochopitelně jen ty části aplikací, které jsou součástí aplikační vrstvy.

Na počátku vývoje protokolové sady TCP/IP (tedy v dobách počátků Internetu a jeho předchůdce ARPANETu) se používaly 3 typy aplikačních služeb, a to:

- přenos souborů (pomocí protokolu FTP);
- vzdálené přihlašování (pomocí protokolu telnet);
- elektronická pošta.

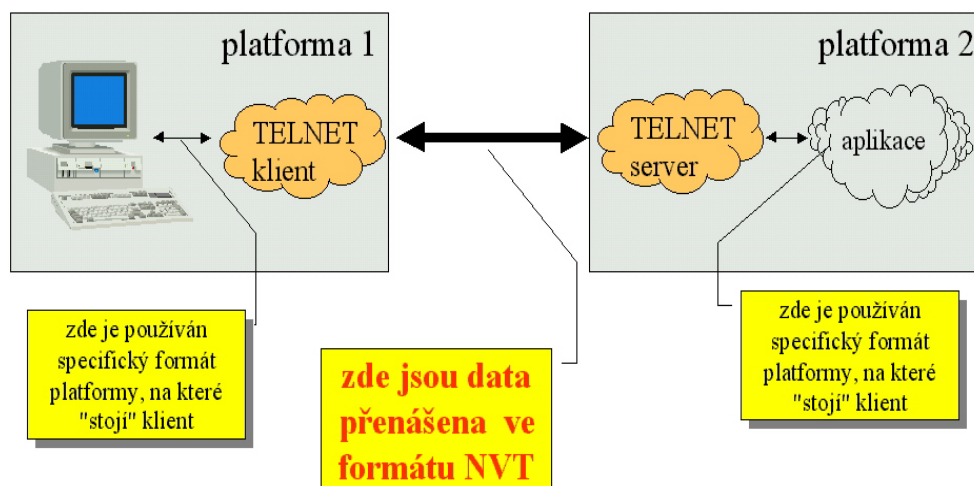
Později se objevily i další aplikace, z nichž se do dnešní doby udrželo především sdílení souborů pomocí protokolu NFS a zejména dnes nejrozšířenější služba WWW (World Wide Web).

### 14.2 Telnet

Protokol telnet se používá ke vzdálenému přihlašování. Znamená to, že umožňuje použití aplikací na vzdáleném počítači. Je to možné pro aplikace, které podporují výpočetní model host/terminál a rovněž operační systém musí podporovat terminálové relace, neboli přesměrování vstupů výstupů na konkrétní terminál. To umožňují kromě jiných např. operační systémy unixového typu, naproti tomu to není možné v operačních systémech jako např. MS-DOS, Windows 9X apod.

Protokol telnet slouží k přihlašování mezi platformami různých operačních systémů. Proto se telnet musí vyrovnat s odlišnostmi různých terminálů i různých serverů. Toho dosahuje tak, že zavádí pevně daný mezistupeň nazývaný Network Virtual Terminal (NVT), který definuje formát přenášených dat, rozsah kláves, tvar příkazů apod. Schéma fungování protokolu telnet je zobrazeno na obr. 14.1.





Obr 14.1: Schéma funkce protokolu telnet

NVT jakožto společné minimum schopností všech terminálů odpovídá funkcím jednoduchého řádkového terminálu, tedy data jsou členěna jen na řádky, přenášena po znacích, komunikace je poloduplexní (i když používá protokolu TCP). Telnet však obsahuje možnosti rozšíření, na nichž se mohou server s klientem dohodnout.

Příkladem takového rozšíření může být např. přenos znaků s ASCII kódem vyšším než 127 (standardně se používají jen 7-bitové ASCII znaky).

### Kontrolní otázky:

1. K čemu slouží protokol telnet?
2. Jaké operační systémy na serveru nelze použít pro vzdálené přihlašování přes telnet?
3. Lze při práci pomocí protokolu telnet používat znaky české abecedy?
4. Je možné při vzdáleném přihlášení k serveru protokolem telnet provést změnu hesla?
5. Obsahuje protokol telnet nějaký prostředek pro zabezpečení proti odposlechu hesla?

## 14.3 Protokoly FTP a NFS

Zatímco protokol telnet slouží ke vzdálenému přihlašování, kdy se spouští program na vzdáleném počítači a pokud zpracovává soubory, zpravidla se tyto soubory rovněž nacházejí na tomto vzdáleném počítači. Naproti tomu pokud potřebujeme zpracovávat vzdálený soubor pomocí programu, který se nachází na lokálním počítači, je třeba využít jiného mechanismu. K tomu je možno použít přenos souborů nebo sdílení souborů.

Rozdíl mezi sdílením souborů a přenosem souborů spočívá v několika rysech, které shrnuje tabulka 14.1.

Sdílení souborů	Přenos souborů
transparentní řešení pro uživatele	řešení netransparentní - uživatel si uvědomuje skutečnost, že soubor se nachází na vzdáleném počítači
uživatel nemusí znát umístění souboru	uživatel musí znát umístění souboru
uživatel si soubor nemusí explicitně zpřístupnit	uživatel si soubor musí explicitně zpřístupnit

Tabulka 14.1: Porovnání sdílení souborů a přenosu souborů

Pro sdílení souborů se v protokolové sadě TCP/IP používá nejvíce protokol NFS, kdežto pro přenos souborů se používá ponejvíce protokol FTP.

### Protokol FTP

Protokol FTP je jedním z nejstarších protokolů v protokolové sadě TCP/IP, neboť pochází dokonce ještě z období před vznikem protokolové sady TCP/IP. Byl používán již nad protokolem NCP, což byl, jak víme z 1. kapitoly, první protokol používaný v ARPANETu.

Vzhledem k tomu, že v době vzniku protokolu FTP byly mezi různými operačními systémy mnohem větší odlišnosti než dnes, se musel se všemi těmito odlišnostmi již od počátku FTP umět vyrovnat. Příkladem takové odlišnosti může být velikost slova nebo reprezentace znaků používaného v daném operačním systému. Dnes se většina takových odlišností eliminovala, odlišnost znázornění znaků (především znaků národních abeced) však přetrvává. Proto také během vývoje protokolu FTP většina schopností vyrovnávat rozdíly mezi platformami vymizela, pouze schopnost konvertovat textové soubory při přenosu mezi různými platformami zůstala zachována.

Protokol FTP definuje 2 základní **režimy** práce, **textový** (při něm se provádějí konverze znaků v přenášeném souboru), a **binární** (v něm se konverze neprovádí, je pro přenos zcela transparentní).

FTP podobně jako telnet zavádí pro potřeby přenosu jednotný formát dat. Také u FTP se v případě potřeby mohou komunikující strany dohodnout na přenosu v jiném formátu.

Protokol FTP přenáší soubory bez ohledu na jejich vnitřní strukturu. Soubor je implicitně přenášen jako souvislý proud dat (tzv. stream mode). Alternativně umožňuje protokol FTP použít tzv. blokový režim, při němž se mezi bloky vkládají tzv. záložky, k nimž je možno se po eventuálním přerušení spojení vrátit. Tím je možno ušetřit čas a přenosovou kapacitu, neboť při přerušení přenosu např. v polovině souboru je možno navázat přenos od poslední záložky a není nutno začínat přenos znovu od počátku souboru. Pro využití této schopnosti je pochopitelně nutné, aby blokový režim podporoval jak FTP server, tak FTP klient. Ojedinele se používá také tzv. komprimovaný režim přenosu, který eliminuje opakující se znaky.

Protokol FTP je zpravidla implementován tak, že jeho funkce jsou rozděleny mezi 2 aplikační entity:

- Protocol interpreter (interpret protokolu);

## Počítačové sítě

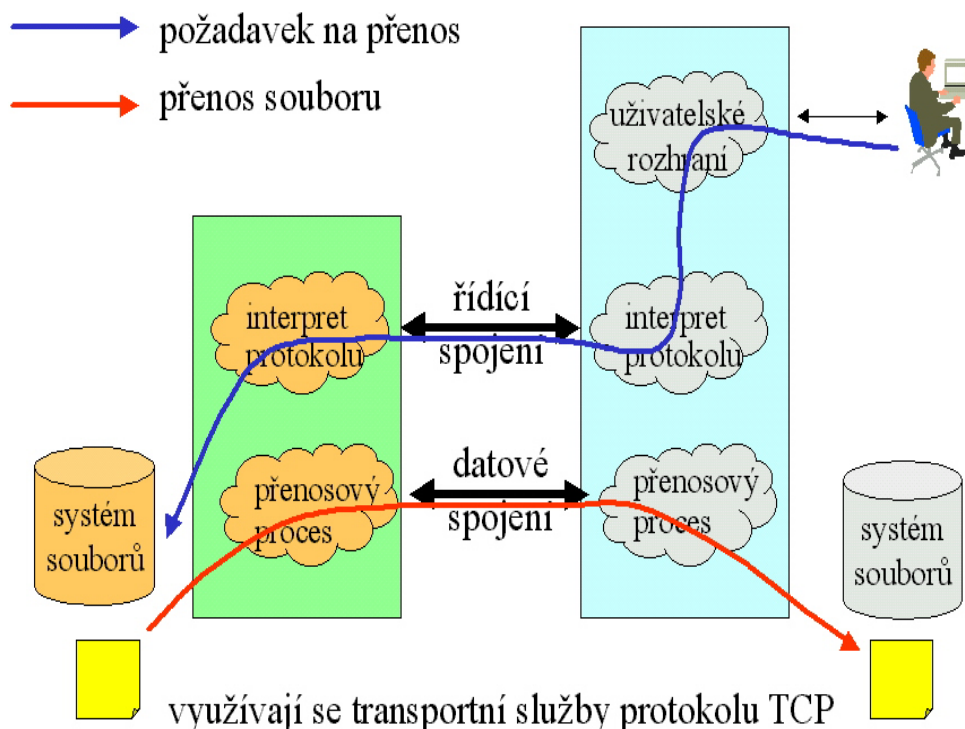
- Data transfer process (proces přenosu dat).

Interpretr protokolu existuje trvale (vytvoří se ihned po spuštění programu, tedy FTP klienta či FTP serveru). Proces přenosu dat se vytváří až na základě požadavku na přenos určitého souboru a po jeho ukončení zaniká.

Pro komunikaci se používají 2 spojení, spojení řídicí určené pro přenos příkazů, a spojení datové, jehož prostřednictvím se realizuje přenos souborů. Oddělení datového a řídicího spojení je výhodné především proto, že pomocí řídicího spojení je možno řídit přenos i během přenosu dat (např. je možné předčasně ukončit přenos, pokud se výrazně zpomalí, je možno signalizovat konec souboru apod.).

Řídicí spojení přetrvává po celou dobu spojení klienta s určitým serverem. Toto spojení navazuje vždy klient na dobře známý port serveru 21. Datové spojení se ustavuje až na základě požadavku klienta. datové spojení navazuje server, a to bez ohledu na směr přenosu souboru (tedy i v případě, že se bude přenášet soubor z klienta na server). Pro datové spojení používá server port 20.

V případě, že to klient požaduje, může navázat datové spojení namísto serveru, musí to však serveru předem oznámit nastavením tzv. pasivního režimu. V případě klienta nacházejícího se za firewallem, který požadavky na otevření FTP spojení zvenčí sítě neakceptuje, je to pro správné fungování FTP protokolu nutné. Schéma funkce protokolu FTP je ilustrováno na obrázku 14.2.



Obr 14.2: Schéma funkce protokolu FTP

Protokol FTP má zabudován mechanismus pro přihlašování uživatelů, tedy server si je vědom toho, který uživatel se přihlašuje, a po korektním přihlášení mu podle jeho oprávnění zpřístupní příslušné soubory a adresáře. K lokálním souborům přistupuje FTP server vždy jménem uživatele, který spustil FTP klienta, který spojení s daným serverem inicioval. Je však nutno poznamenat, že podobně jako protokol telnet ani FTP nijak nechrání uživatelské jméno a

heslo před odposlechem při přenosu po síti (nepoužívá tedy např. šifrování hesla).

Tzv. anonymní FTP servery, které se často používají pro zveřejnění souborů, nejsou při vyžadování uživatelského jména a hesla výjimkou. Používá se však konvence pojmenování uživatele anonymous a jako heslo je zpravidla požadována e-mailová adresa.

Pro účely řízení přenosu definuje protokol FTP vlastní řídicí jazyk. Příkazy řídicího jazyka jsou pro snazší práci uživatelů v řádkových FTP klientech nahrazeny odpovídajícími, avšak snáze zapamatovatelnými příkazy tzv. uživatelského jazyka. Klient zajistí jejich překlad do řídicího jazyka a jejich zaslání serveru. Podobně je tomu i v případě grafických FTP klientů, ovšem zde uživatel nemusí příkazy vůbec zadávat v textové formě.

Řídicí jazyk obsahuje 3 skupiny příkazů, které jsou uvedeny dále spolu s příkladem příkazů uživatelského jazyka:

- Příkazy pro řízení přístupu (např. otevření spojení - příkaz open, zadání hesla - příkaz user, atd.);
- příkazy pro nastavení parametrů (např. nastavení textového režimu - příkaz ascii);
- výkonné příkazy (např. přenos souboru na server - put, přenos souboru ze serveru - get, změnu aktuálního adresáře - cd, vytvoření adresáře, smazání při přejmenování souboru či adresáře apod.).

FTP server na příkazy odpovídá 3-znakovými odpověďmi složenými z číslic. První číslice signalizuje třídu odpovědi (1 - dočasná kladná, 2 - trvalá kladná, 3 - prozatímní, 4 - dočasná záporná, 5 - trvalá záporná odpověď). Další 2 číslice odpověď přesněji specifikují. Klient tyto odpovědi zpravidla vypisuje spolu s jejich textovou interpretací.

Za zmínku stojí i dnes již pomalu ustupující zjednodušená varianta protokolu FTP, která se nazývá TFTP (Trivial FTP). Ta se používá hlavně k zavedení operačního systému do bezdiskových stanic a terminálů. Její omezení spočívají především v tom, že nezná pojem uživatele, nepodporuje tedy přihlašování, a nepodporuje relativní cesty k souborům pomocí aktuálního adresáře, všechny cesty se tedy musí zadávat explicitně celé.

### Protokol NFS

Pro jednodušší práci se soubory, než jakou umožňuje protokol FTP, vznik protokol NFS (Network File System). Tento protokol nepochází z původní protokolové sady TCP/IP, byl vyvinut mnohem později jako proprietární řešení firmou Sun. Ta jej později předložila ke standardizaci v rámci TCP/IP a dnes je popsán standardem RFC 1094. Přestože vznikl v prostředí Unixu, není vázán na žádný konkrétní operační systém a dnes již existují implementace NFS serveru i NFS klienta pro všechny rozšířenější operační systémy.

Základní vlastností protokolu NFS je **bezstavovost**. Znamená to, že server si nemusí pamatovat průběh předchozí komunikace s klientem. K tomu je nutné, aby všechny přípustné požadavky klienta byly vůči serveru uzavřené, tedy aby ponechaly server po provedení požadavku ve stejném stavu, jako před jeho započítím. Proto mohou být NFS klientem vyžadovány pouze tzv.

**idempotentní** operace, které jsou vícenásobně opakovatelné se stejným výsledkem.

### Příklad

Požadavek „čti dalších X bytů souboru“ není idempotentní, protože po prvním přečtení X bytů se podruhé přečtou jiná data.

Naproti tomu požadavek „čti X bytů souboru počínaje bytem Y“ je idempotentní. Ověřte si to na příkladu přenosu dat, který si sami navrhnete!

---

Požadavek na idempotentní operace znamená mimo jiné, že není možno samostatně otevřít nebo zavřít soubor. Namísto toho musí být při každém přístupu k souboru soubor otevřen, přečtena nebo zapsána data a soubor opět uzavřen.

Zajištění platformní nezávislosti při přístupu k souborům vyžaduje, aby se při komunikaci nepoužívaly systémově závislé specifikace souboru, tedy např. tzv. cesty popisující umístění souboru v adresářové hierarchii, neboť jejich struktura a reprezentace je v různých operačních systémech různá. Namísto toho se používají jednorozměrné systémové identifikace souboru (tzv. file handle).

I v rámci protokolu NFS však existuje nejméně jedna operace, která principiálně bezstavová být nemůže. Touto operací je přihlášení k serveru. Proto byla tato operace vyčleněna ze sady příkazů NFS a přihlášení je realizováno prostřednictvím tzv. mount serveru, který také vydá klientovi první file handle, jehož pomocí si klient již potom pomocí bezstavové komunikace s NFS serverem může vyžádat další file handle pro práci s dalšími soubory a adresáři.

Implementace protokolu NFS má ještě jednu zvláštnost. Ta spočívá v tom, že obsahuje dva samostatně použitelné protokoly RPC (Remote Procedure Call) a XDR (eXternal Data Representation), které lze použít samostatně i mimo protokol NFS. Někdy se uvádí, že tyto dva protokoly představují jakýsi zárodek relační a prezentační vrstvy v TCP/IP. Ve srovnání s povinně používanými vrstvami referenčního modelu ISO/OSI se však jedná pouze o volitelně použitelné komponenty.

### Kontrolní otázky:

1. Obsahuje protokol FTP nějaký prostředek pro zabezpečení proti odposlechu hesla?
  2. K čemu slouží protokol NFS?
  3. Jaký je rozdíl mezi sdílením souborů a přenosem souborů?
- 

## 14.4 Elektronická pošta

Elektronická pošta je služba přenosu zpráv (původně krátkých a pouze textových, to ale dnes již dávno neplatí), která je implementována různými způsoby v mnoha různých prostředích (známé jsou např. systémy X.400, Lotus ccMail a jiné). Je nutno poznamenat, že různé systémy elektronické pošty jsou vzájemně nekompatibilní a k tomu, aby bylo možné mezi různými systémy komunikovat, jsou nutné převodní brány.

Elektronická pošta v rámci protokolové sady TCP/IP je jednou z implementací služeb elektronické pošty, avšak díky rozšíření používání TCP/IP je ze všech systémů elektronické pošty dnes nejpoužívanější. Je založena na protokolu SMTP a na standardu RFC 822. V dalším textu budeme elektronické poště v rámci protokolové sady TCP/IP hovořit pouze jako o „elektronické poště“, případně „elektronické poště SMTP“ pokud nebude explicitně zmíněn jiný význam tohoto výrazu.

Úspěch elektronické pošty obecně a elektronické pošty v Internetu zvláště spočívá především v jejích vlastnostech. Mezi nejvýznamnější z nich patří rychlost, nízké náklady, efektivita a zejména možnost komunikace „off-line“, tedy skutečnost, že odesílatel může zprávy odesílat bez ohledu na to, zda je příjemce právě připojen, a příjemce může zprávy zpracovat až tehdy, když se mu to hodí.

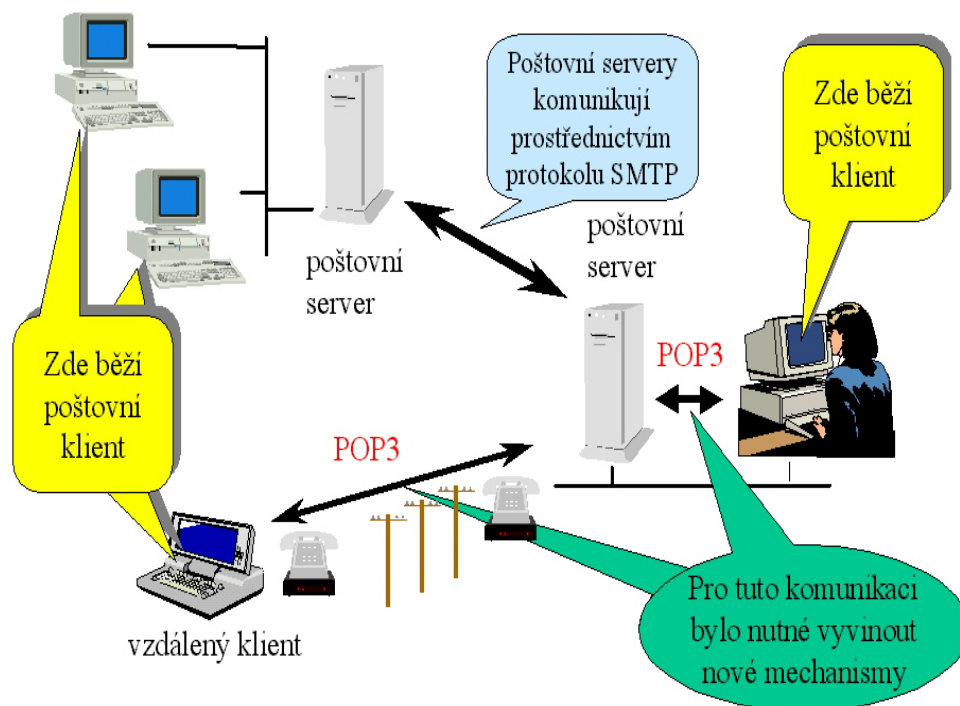
Původní podoba elektronické pošty byla velmi prostá: umožňovala pouze předávání krátkých textových sdělení elektronickou formou. Předávaný text mohl obsahovat pouze znaky základní ASCII sady.

Elektronická pošta vychází podobně jako většina ostatních aplikací v TCP/IP z výpočetního modelu klient/server. Serverem je tzv. poštovní server, který zabezpečuje přenos zpráv na cílový server a shromažďuje přijaté zprávy pro ty uživatele, kteří nejsou momentálně připojeni. Klientem je program, který umožňuje zprávy přijímat a číst, psát a odesílat a provádět s nimi další úkony.

Na fungování elektronické pošty SMTP se podílí několik protokolů. Přenos zpráv zajišťuje protokol SMTP, práci s doručenými zprávami ve schránce uživatele zajišťují protokoly POP3 a IMAP. Formát zpráv a formát adres je definován doporučením RFC 822. Pozdější rozšíření, především v oblasti formátu zprávy, jsou definována standardem MIME.

Elektronická pošta SMTP prošla poměrně dlouhým vývojem. Původní koncepce počítala s tím, že uživatelé jsou připojeni k poštovnímu serveru prostřednictvím terminálové sítě, čili de facto přímo, takže poštovní server a klient běží na stejném počítači. Proto služba elektronické pošty používala pouze s přenosem zpráv z poštovního serveru odesílatele (z adresářů určených k odeslání) na jiný poštovní server (případě tentýž) do příslušného adresáře adresáta zprávy. Později se však ukázalo, že častější bude případ, kdy uživatel bude k poštovnímu serveru připojen z jiné stanice prostřednictvím počítačové sítě (zpravidla lokální sítě nebo pomocí dočasného připojení), tedy že klient poběží na jiném uzlu než server. Proto bylo nutné dodatečně vyvinout prostředky pro komunikaci mezi poštovním klientem a serverem. Pro odesílání zpráv na poštovní server bylo možno použít protokolu SMTP, avšak pro přenos zpráv opačným směrem ze schránky uživatele na serveru do schránky uživatele na stanici byl vyvinut protokol POP, dnes se používá ve verzi POP3. Představa fungování elektronické pošty je znázorněna na obrázku 14.3.

## Počítačové sítě



Obr 14.3: Schéma fungování elektronické pošty SMTP

Poštovní schránka je vždy umístěna na poštovním serveru, na kterém má uživatel službu elektronické pošty zřízenou. Zpravidla je však schránka rozdělena na 2 části: přijaté zprávy, které si dosud uživatel nevyzvedl, jsou uloženy ve schránce na serveru, zatímco zprávy, které si uživatel již vyzvedl, jsou při vyzvednutí přeneseny na jeho počítač. V takovém případě je nutné nové poštovní zprávy explicitně přenášet neboli „stahovat“ na počítač uživatele. Ke stahování zpráv se používá zpravidla protokol POP3. Toto uspořádání je univerzální, a proto se s ním můžeme setkat ve všech prostředích.

Druhou možností je umístit celou schránku na poštovní server a vyzvednuté zprávy nikam nepřenášet. Tato varianta není vhodná, pokud je klient připojen pouze dočasně nebo sice trvale, ale pomalým připojením. Naproti tomu v prostředí s trvalým rychlým připojením může přinést některé výhody. Za nejvýznamnější výhodu se považuje skutečnost, že uživatel má v tomto případě k dispozici všechny zprávy včetně již přečtených z každé stanice připojené k jeho poštovnímu serveru. Znamená to, že uživatel si může prohlížet všechny zprávy např. z kteréhokoli počítače v lokální síti. Tato varianta se realizuje zpravidla pomocí protokolu IMAP.

### Formát zprávy

Součástí vývoje elektronické pošty bylo i postupné rozšiřování služeb, především v oblasti možného obsahu zprávy. Z původního omezení na ASCII znaky a striktního omezení velikosti se služba vyvinula do dnešní podoby, kdy je možno do zprávy vkládat znaky národních abeced, formátování, obrázky, přílohy ve formě souborů prakticky libovolného formátu apod. a velikost zprávy je omezena pouze kapacitou poštovní schránky příjemce a případnými omezeními na poštovním serveru odesílatele. O formátu zprávy nyní zmíníme podrobněji.

Každá zpráva obsahuje 2 základní části: hlavičku zprávy a datovou část

neboli tělo zprávy. Hlavička obsahuje nejdůležitější údaje, jimiž se řídí poštovní server při práci se zprávou. Její struktura včetně přesné syntaxe adres je definována doporučením RFC 822. Toto doporučení nijak nedefinuje obsah těla zprávy, o němž pouze předpokládá, že je tvořeno ASCII textem. Určitou strukturu do těla zprávy zavedl teprve standard MIME, který tím mimo jiné umožnil zasílání příloh ve zprávách.

Hlavička zprávy dle RFC 822 je tvořena jednotlivými položkami, které jsou vždy uvozeny klíčovým slovem končícím dvojtečkou. Každá z položek hlavičky začíná na novém řádku. Pořadí položek v hlavičce není předepsáno, existuje však doporučené pořadí. Několik položek je povinných, většina jich však je nepovinných. Ty řádky v hlavičce, které nezačínají žádným z klíčových slov, jsou ignorovány. Hlavička je od těla zprávy oddělena prázdným řádkem.

Nejdůležitějšími položkami hlavičky zprávy jsou tyto:

- From: (adresa autora zprávy);
- To: (adresa příjemce);
- Sender: (adresa odesílatele, je-li jím někdo jiný než autor zprávy);
- Cc: (adresa pro zaslání kopie zprávy „na vědomí“);
- Bcc: (adresa pro zaslání „slepé“ kopie zprávy - příjemce v této položce se nezobrazí ostatním příjemcům v položkách To: a Cc:);
- Reply-to: (adresa pro zaslání odpovědi, pokud je jiná než adresa From:);
- Return-Path: (adresa pro vrácení zprávy v případě její nedoručitelnosti, pokud je jiná než adresa From:);
- Date: (datum a čas odeslání zprávy)
- Subject: předmět zprávy.

Všechny výše uvedené položky s výjimkou poslední z nich (Subject:) mají pevně stanovenou syntaxi, neboť smějí obsahovat pouze adresu, případně více adres oddělených čárkami. Výjimkou je pochopitelně předposlední položka Date:, která obsahuje datum a čas v předepsaném formátu, do vyplnění této položky však uživatel prakticky nemůže zasáhnout.

Syntaxe adres je vcelku jednoduchá: adresy smějí obsahovat pouze písmena, číslice a některé další znaky ze základní ASCII sady. Adresa má 2 části, které odděluje znak „@” (tzv. zavináč). Ten se pochopitelně na jiném místě adresy vyskytovat nesmí.

Adresa se může používat rovněž ve formě s komentářem (nejčastěji bývá komentářem skutečné jméno adresáta), a to buď nejprve komentář a pak adresa oddělená lomenými závorkami (např. „Tomáš Sochor <tomas.sochor@osu.cz>”), nebo nejprve adresa a pak komentář v kulatých závorkách (např. „tomas.sochor@osu.cz (Tomáš Sochor)”). Je přitom třeba upozornit, že použití českých znaků či jiných znaků národních abeced umožnil až standard MIME, nebylo jej tedy možno používat ve starších poštovních klientech. Pro účely zapsání adresy ve formě odkazu pro použití v HTML dokumentech se adresa uvozuje příznakem „mailto:“. Příkladem takového odkazu je tedy „mailto:tomas.sochor@osu.cz”.



Vzhledem k tomu, že elektronická pošta SMTP vznikala v době, kdy nebylo možné po síti běžně zasílat 8-bitová slova (8. bit se často používal jako tzv. paritní), předpokládá elektronická pošta, že všechny znaky tvořící zprávu budou patřit do základní ASCII sady (kódy 0-127). Pro přenos 8-bitových bytů bylo nutno taková data nejprve upravit do 7-bitové formy.

Pro úpravu příloh do 7-bitové podoby se postupně vyvinulo několik způsobů. Nejstarší z nich (UUEncode pocházející z unixu a BinHex pocházející z počítačů Macintosh) jsou poněkud nesystematické, protože řeší pouze přibalování příloh do zpráv a nezapovídají se dalšími aspekty, např. rozlišením typu přílohy, formátováním obsahu (těla) poštovních zpráv, použití znaků národních abeced v těle zprávy apod.

Systematičtější řešení přinesl až standard MIME (Multipurpose Internet Multimedia Extensions), který umožňuje bezproblémovou práci s přílohami díky tomu, že definuje nejen způsob úpravy přílohy pro odeslání, ale umožňuje přikládání více příloh do zprávy, definuje typy příloh napomáhající příjemci otevření souboru ve správné aplikaci, umožňuje vkládání 8-bitových znaků (např. češtiny, formátovacích znaků apod.) do těla zprávy, předmětu zprávy i komentářové části adres.

MIME zavádí 2 typy kódování (neboli převodu 8-bitových dat na 7-bitová), konkrétně tzv. Quoted Printable a Base64. zavádí dvousložkové MIME typy, které definují typ přílohy a způsob zpracování (např. image/gif, application/msword apod.), a především zavádí do hlavičky nové položky, které se použijí především pro předání informací o přílohách, kódování apod. v hlavičce zprávy.

Zavedení nových položek do hlavičky je umožněno již zmíněnou vlastností definice hlavičky dle RFC 822, že totiž položky začínající jiným než klíčovými slovy se ignorují. Z důležitých nových položek si uvedeme alespoň položku Content-type určující pomocí MIME typu typ obsahu zprávy a případně i informaci o kódování národních znaků. Díky zavedení této položky bylo například umožněno dnes poměrně rozšířené zasílání zpráv ve formátu HTML.

MIME typy mají 7 základních typů (text, image, audio, video, application, multipart a message) a velké množství podtypů, které není uzavřeno pro tvorbu nových. Uvádí se vždy ve tvaru typ/podtyp a používají se také mimo elektronickou poštu, např. k určení typu stránek, které WWW server zasílá klientovi.

V případě, že uživatel používá klienta, který dosud standard MIME nepodporuje, bude mít některé části zprávy nečitelné, avšak nezpůsobí to nedoručení zprávy.

### Kontrolní otázky:

1. Který protokol pro práci s elektronickou poštou je nejstarší a proč?
2. K čemu slouží protokoly POP3 a IMAP a čím se liší?
3. K čemu slouží položka Bcc: v hlavičce zprávy?

## 14.5 World Wide Web

WWW neboli World Wide Web je dnes nejrozšířenější službou Internetu. Tato

služba se však používá k prezentaci dat i mimo Internet, v soukromých sítích (Intranetech) apod.

Služba WWW vznikla v roce 1989 ve středisku CERN v Ženevě, původně jako textová služba. Služba WWW vychází z principu hypertextu. Hypertext je dokument rozdělený na menší stránky, mezi nimiž mohou existovat libovolné vazby pomocí tzv. aktivních odkazů. Hypertext se často používá např. v systémech nápovědy k operačním systémům, zde se však používají pouze odkazy mezi jednotlivými stránkami, případně na jiné dokumenty v rámci daného počítače. Služba WWW obohacuje hypertext především o možnost umísťování odkazů na libovolný soubor kdekoli na síti. Přitom aktivní odkazy mohou být umístěny nejen v samotném textu, ale i na obrázcích nebo jejich částech.

Služba WWW prošla rychlým vývojem od původně textové služby do dnešní podoby, kdy umožňuje začleňování celé řady formátů souborů včetně multimediálních (zvuk, video, animace apod.).

Během vývoje se WWW stal ze služby také platformou pro poskytování jiných služeb. Příkladem je vyhledávání v Internetu, které bylo v počátcích realizováno specializovanými službami (Archie, Veronica, FTP Search apod.), po nástupu služby WWW se začalo přesouvat pod tuto platformu a dnes se vyhledávání realizuje zpravidla prostřednictvím přístupu přes WWW a mnohé specializované vyhledávací služby prakticky zanikly. Další službou, ke které se často přistupuje prostřednictvím WWW, je elektronická pošta.

Služba WWW je také velmi významná tím, že se jedná prakticky o jedinou službu TCP/IP, která se ve velkém měřítku používá ke komerčním účelům. Právě komerční využití této služby je největším hybatelem jejího prudkého vývoje.

Služba WWW vychází z modelu klient/server. WWW server uchovává a spravuje jednotlivé WWW stránky a na žádost klienta jim je poskytuje (zasílá). WWW klient (většinou nazývaný WWW prohlížeč neboli browser) si vyzvedává stránky od WWW server; a zobrazuje je pro uživatele.

Pro fungování služby WWW je nutné mít definovaný jednak způsob přenosu stránek mezi serverem a klientem (to definuje protokol HTTP), a také formát stránek (ten je definován jazykem HTML). Filosofie jazyka HTML vychází z toho, že definuje strukturu dokumentu (např. nadpisy, seznamy, obrázky apod.), nikoli to, jak má dokument vypadat na obrazovce (to definuje WWW klient podle svých grafických schopností). V dnešní době je jazyk HTML poměrně komplikovaný – obsahuje velké množství různých možností, např. vkládání částí programového kódu pomocí skriptů nebo appletů, umožňuje pomocí formulářů získávat data od uživatele apod.

WWW klient zobrazuje obdržené HTML soubory a soubory jiných typů podle svých grafických možností, přičemž někdy používá pro zobrazení speciálních formátů externí programy v rámci operačního systému stanice, na které pracuje. V dnešní době je většina WWW klientů schopna fungovat i jako klienti služby FTP, často je jejich součástí i klient elektronické pošty.

Protokol http je jednoduchý přenosový protokol, který využívá služeb protokolu TCP. Data přenáší v textovém tvaru, server očekává požadavky na portu 80. Protokol http funguje bezstavově, pro každý objekt na stránce se

navazuje samostatné spojení. Komunikace probíhá tak, že klient zašle požadavek a server na něj odpoví a pak spojení ukončí (v případě protokolu http 1.1 definovaného v RFC 2068 se ukončí spojení až po načtení celé stránky, resp. po přenesení všech souborů z daného serveru, které jsou v daném okamžiku požadovány).

Pro komunikaci je definováno několik jednoduchých příkazů označovaných jako metody. Nejdůležitějšími metodami jsou GET, pomocí které klient žádá o zaslání stránky, a POST, kterou klient odesílá data ve formuláři serveru. Odpovědi mají podobně jako v případě FTP podobu tříznakových číselných kódů. Odpověď začínající číslicí 1 je informační, číslicí 2 začíná kladná odpověď serveru, číslicí 3 začíná upozornění na očekávanou další aktivitu klienta, číslicí 4 začíná oznámení chyby na straně klienta a číslicí 5 začíná oznámení chyby na straně serveru. Typickým příkladem odpovědi, s níž se může uživatel často setkat, je „404“, jejíž význam je vyjádřen textem „Not Found“, což znamená, že na daném serveru nebyl nalezen požadovaný soubor. Součástí kladné odpovědi pochopitelně je i požadovaná stránka (v případě metody GET).

Vzhledem ke skutečnosti, že bezstavový charakter komunikace neumožňuje, aby si server pamatoval například průběh předchozí komunikace s klientem, omezuje to možnost poskytování některých služeb. Jednou z možností, jak to obejít, je vložit tyto informace do adresy serveru jako parametr, avšak univerzálnější řešení představují tzv. cookies, které byly zavedeny v RFC 2109. Cookies jsou krátké textové údaje, které generuje server a zasílá je klientovi. Ten si je může uložit na disk pro potřeby další komunikace s tímto serverem. Při další komunikaci s tímto serverem mu klient zašle příslušný cookie a server si jej může podle toho identifikovat. Ukládání cookies na straně klienta je pochopitelně volitelné a je možné jej zakázat. Tím se zvýší bezpečnost klienta, ovšem může se omezit možnost pracovat s některými WWW servery.

### Kontrolní otázky:

1. Co je to hypertext?
2. K čemu slouží jazyk HTML?
3. K čemu slouží tzv. metody protokolu HTTP?
4. Co signalizuje odpověď WWW serveru 404?
5. Co je cookie a kde se uchovává?

---

## 14.6 Služba DNS

Při práci se službami TCP/IP si mnozí uživatelé ani neuvědomí, že prakticky všechny služby využívají jednu službu pomocnou, totiž DNS (Domain Name System). Služba DNS slouží k tomu, aby si uživatel nemusel pamatovat IP adresy, které jsou zapamatovatelné obtížně, a místo toho si mohl zapamatovat adresu serveru v textové formě. Jistě se snáze pamatuje např. adresa [www.osu.cz](http://www.osu.cz) než adresa 195.113.106.17. Součástí služby DNS jsou pochopitelně i pravidla pro vytváření doménových jmen, správa domén, ale my se zde zaměříme jen na to nejdůležitější pro uživatele, tedy na mechanismus převodu symbolických jmen na IP adresy.

Systém DNS je založen na hierarchické struktuře tzv. domén, jimž odpovídá systém tzv. DNS serverů, které odpovídají na požadavky klientů.

Požadavek na DNS server vygeneruje každý klient každé aplikační služby TCP/IP při obdržení požadavku, v němž se vyskytuje symbolické jméno místo IP adresy. Protože je těchto požadavků velké množství, je systém zefektivněn ukládáním odpovědí na požadavky do cache na každém DNS serveru. Teprve pokud není odpověď na požadavek nalezena v cache, provede se dotaz do hierarchické struktury DNS serverů.

### Shrnutí

V této kapitole jste se seznámili s vlastnostmi aplikační vrstvy a jejich nejvýznamnějšími službami. K nim patří především WWW, elektronická pošta, telnet, FTP a NFS a služba DNS, s nimiž jsme se seznámili podrobněji. Odpověď z chace se nazývá neautoritativní. Aplikace si však může v případě potřeby vyžádat pouze autoritativní odpověď, tedy odpověď pocházející přímo od DNS serveru obsluhujícího příslušnou doménu.

### Korespondenční úkoly:

1. Napište, který z protokolů pro vzdálené přihlašování a práci se soubory (telnet, FTP, NFS) používáte v praxi a k jakému účelu.
2. Proč není možné bez úprav přiloženého souboru posílat ve zprávě přílohy?
3. Proč je při odesílání zpráv vhodnější držet se standardu MIME místo kódování UUencode nebo BinHex?
4. Popište přesně použití položky Bcc: (slepá kopie) v hlavičce zprávy elektronické pošty!
5. V čem myslíte, že spočívá potenciální nebezpečnost cookies pro klienta?
6. Pokud provedete změnu hesla svého uživatelského účtu prostřednictvím protokolu telnet na dálku, jakému nebezpečí se vystavujete?
7. Jaké základní výkonné příkazy uživatelského jazyka protokolu FTP znáte?

## 15 Moderní přenosové technologie

### 15.1 ATM

ATM neboli Asynchronous Transfer Mode je přenosová technologie, která pochází z prostředí telekomunikačních sítí. Vznikla původně pro potřeby širokopásmových přenosů pomocí služby B-ISDN, ale díky svému poměrně pozitivnímu přijetí v prostředí počítačových sítí se později osamostatnila. Základní ideou při vzniku ATM byla snaha o sjednocení počítačových a telekomunikačních sítí na základě jedné přenosové technologie.

ATM se snaží vyjít vstříc jak požadavkům z oblasti počítačových sítí na nárazové přenosy větších objemů dat, kterým lépe vyhovuje přenos dat ve větších blocích o pružně měnitelné velikosti, tak požadavkům telekomunikačních sítí na pravidelné přenosy předem stanoveného množství dat při dodržení určitých kvalitativních požadavků na přenos, čemuž lépe vyhovuje přenos dat v blocích pevné délky (v průměru výrazně menší než datové bloky počítačových sítí).

Řešení, které ke splnění protichůdných požadavků ATM našlo, spočívá v přenosu dat v blocích pevné délky datové části 48 bytů s 5-bytovou hlavičkou. Tyto bloky se nazývají **buňky**.

Z hlediska vývoje technologií v telekomunikačních sítí lze na ATM pohlížet jako na specifickou formu statistického multiplexu, jehož princip spočívá v tom, že datové toky jsou sdružovány do jednoho datového toku, aniž by bylo dopředu určeno využití jednotlivých časových slotů. V oblasti počítačových sítí navazuje na protokoly X.25 a Frame Relay. Technologie ATM však přesto díky své odlišnosti od ostatních technologií počítačových sítí naráží na mnohé problémy při spolupráci s nimi.

ATM pracuje na spojovaném principu, nejedná se však o přepojování okruhů, ale o tzv. přepojování buněk. Jeho princip spočívá v tom, že každý uzel může síť požádat o vytvoření ne jen jednoho, ale i více současně existujících spojení s různými uzly. Tato spojení se nazývají virtuální okruhy. Virtuální okruhy mohou navíc mít různé parametry, např. různou přenosovou rychlost, různou míru spolehlivosti apod. Uzel pak předává do sítě buňky, které jsou podle jejich identifikátorů přiřazeny do konkrétních virtuálních okruhů.

Virtuální okruhy jsou jednosměrné, v případě potřeby je však lze vytvářet v protisměrných párech. Virtuální okruhy nepoužívají potvrzení, protože se předpokládá, že přenosy se budou odehrávat po optických vláknech, která jsou velmi spolehlivá. Spolehlivost je v případě potřeby zajišťována na vyšších vrstvách technologie ATM.

Virtuální okruhy se pro účely jednoduššího přenosu ATM sítí zpravidla sdružují do virtuálních cest. Vnitřní ATM přepínače (zvané též ATM ústředny) se totiž na rozdíl od vnějších (tedy těch, k nimž jsou přímo připojeny koncové ATM uzly) provádějí rozhodování o cestě jen na základě identifikátoru virtuální cesty a identifikátor virtuálního kanálu nebere v úvahu. Ten se použije až při doručování koncovým uzlům, tedy na vnějších ATM přepínačích.

ATM definuje několik tříd služeb, které si mohou aplikace zvolit podle potřeby. Jejich přehled uvádí následující tabulka.

## Počítačové sítě

	CBR	RT - VBR	NRT-VBR	ABR	UBR
Garance přenosové kapacity	Ano	Ano	Ano	Částečně	Ne
Vhodnost pro přenosy v reálném čase	Ano	Ano	Ne	Ne	Ne
Vhodnost pro nárazový provoz	Ne	Ne	Ano	Ano	Ano
Informuje o zahlcení	Ne	Ne	Ne	Ano	Ne

Tabulka 15.1: Typy služeb ATM

Praktický význam ATM je dodnes omezený. Jejich využití je velmi nákladné, což je dáno především vysokými pořizovacími náklady ATM zařízení. Proto se ATM využívá převážně v páteřních sítích, případně pro speciální účely, kde možnost garance kvality služeb převáží nevýhody v problematické spolupráci s ostatními síťovými technologiemi. Použitím ATM v páteřních sítích se však mnohdy ztrácí výhoda ATM spočívající ve schopnosti garantovat třídy služeb podle požadavku aplikací.

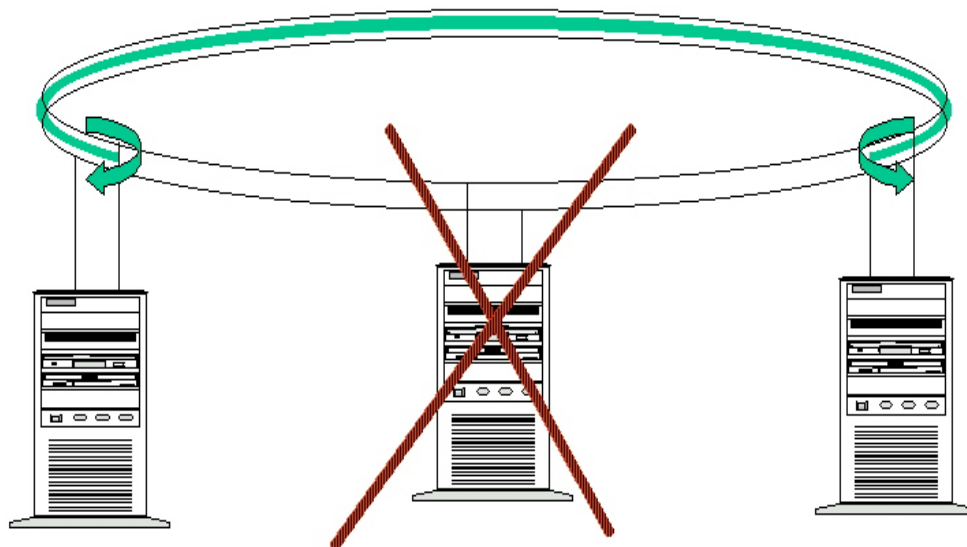
### 15.2 FDDI

FDDI (Fiber Distributed Data Interface) je přenosová technologie, která byla vyvinuta s cílem využít možnosti optických vláken pro přenos dat. Vznikla v polovině 80. let 20. století, standardizována byla institutem ANSI.

FDDI předpokládá fyzickou kruhovou topologii tvořenou optickými vlákny. Používá deterministickou přístupovou metodu založenou na principu předávání pověření (token passing). Umožňuje velký rozsah sítě, neboť obvod kruhu může činit až 200 km při maximální vzdálenosti dvou stanic 2 km.

FDDI je dobře přizpůsobeno pro použití v páteřních spojích, protože má zabudováno mechanismus automatické rekonfigurace, který umožňuje udržet síť v provozu i při přerušení kruhu na jednom místě.

Proto, aby bylo možné docílit udržení provozu při přerušení kruhu, je kruh FDDI dvojitý. Hlavní kruh se běžně používá pro provoz, vedlejší kruh je v záloze. V případě přerušení kruhu se provede tzv. rekonfigurace. Na uzlech, které sousedí s přerušeným úsekem, se propojí hlavní kruh se záložním a záložní kruh se použije pro protisměrný provoz. Tím se kruh prodlouží, ovšem provoz v kruhu je zachován. Schéma rekonfigurace kruhu FDDI je uvedeno na obrázku 15.1.



Obr 15.1: Rekonfigurace FDDI kruhu

Praktický význam FDDI spočívá především v jeho využití v páteřních sítích. Vzhledem ke shodnému adresnímu prostoru s Ethernetem je poměrně snadná jejich koexistence a spolupráce.

### 15.3 Gigabitový Ethernet

Ethernet je bezpochyby nejrozšířenější současnou síťovou technologií. Jeho nejnovější verze dosahuje přenosové rychlosti 1000 Mb/s (1 Gb/s), což je stokrát více, než šířka pásma původní verze Ethernetu. Přitom i tento nový gigabitový Ethernet zůstává kompatibilní s existujícími „starými“ verzemi Ethernetu, používá stejnou přístupovou metodu CSMA/CD a MAC adresy.

Historie Ethernetu začíná již v 70 letech a od té doby si s převahou udržuje výsostné postavení na trhu síťových technologií (podle různých odhadů je více než 80 % v současnosti prodávaných síťových rozhraní typu Ethernet, stejný odhad platí pro procento celkově již instalovaných uzlů).

V roce 1995 došlo přijetím standardu Fast Ethernet k podstatnému nárůstu přenosového pásma (100 Mb/s) a zdálo se, že možnosti dalšího zvyšování jsou již plně vyčerpány. Opak je ale pravdou - stalo se něco, co jsme si nikdo nedokázali představit ani v té nejbujnější fantazii.

První návrh gigabitového Ethernetu spatřil světlo světa v červenci 1997. Nový gigabitový standard je plně kompatibilní s existujícími instalacemi Ethernetu. Jako přístupovou metodu zachovává CSMA/CD. Použití gigabitového Ethernetu je nejčastější pro páteřní spoje lokálních sítí - propojení serverů a prepínačů Fast Ethernet.

Na fyzické vrstvě se používá (vzhledem k převažujícímu použití optických vláken) specifikace Fibre Channel. Existují celkem 4 specifikace gigabitového Ethernetu.

Specifikace 1000Base-SX je určena pro levná mnohavidová vlákna pro kratší horizontální vedení nebo páteřní aplikace. Pro překlenutí větších vzdáleností jednovidovými vlákny je pak určena specifikace 1000Base-LX.

Na metalickou kabeláž jsou zaměřeny dvě specifikace. První z nich,

1000Base-CX je určena pro krátká propojení (do 25 m) stíněným kabelem typu twinax, např. propojení serverů a přepínačů v serverových farmách. Druhá specifikace pro metalickou kabeláž, 1000Base-T, využívá UTP kabeláž kategorie 5 standardních horizontálních rozvodů budov (do 100 m).

Minimální velikost rámce Ethernetu je 64 bajtů. Ta je právě dána standardem 802.3 pro zajištění toho, aby stanice neskončila svoje vysílání dříve, než první bit rámce dosáhne vzdáleného konce kabelu, kde může nastat kolize s jiným rámcem, a případný interferenční signál kolize se nevrátí zpět k vysílající (a zároveň poslouchající) stanici.

Tato minimální velikost rámce se nazývá slot size a pro Ethernet je uvedených 64 bajtů. Odvozenou hodnotou je tzv. kolizní slot (slot time), minimální čas, po který stanice musí vysílat.

Max. vzdálenost mezi dvěma uzly standardního Ethernetu je v případě žlutého koaxiálního kabelu 2,5 km (při max. počtu čtyř opakovačů). Zvýšení přenosové rychlosti musí být vykoupeno:

- buď zachováním slot time (tj. zachováním min. velikosti rámce) a zmenšením rámce, nebo
- zvětšením slot time (tj. zvětšením min. velikosti rámce) při nezměněné velikosti rámce, nebo
- kombinací obou způsobů.

Standard Fast Ethernet vyřešil tento problém prvním z uvedených způsobů, tj. redukcí délky rámců. Maximální velikost kolizní domény se zmenšila v případě UTP kabelů na 200 m, resp. 210 m.

V případě gigabitového Ethernetu se tvůrci specifikace nutně dostali ke stejnému rozhodování. Gigabitový Ethernet je opět desetkrát rychlejší (než Fast Ethernet). Při zachování stejné slot size (min. velikosti rámce) by došlo k redukci segmentů na pouhých 10 m. To je však již příliš málo, aby šlo o použitelné řešení.

Autoři specifikace přesto zachovali jak minimální a maximální velikost rámců standardního Ethernetu, tak rozumnou délku segmentů. Jak toho dosáhli? Zvláštním procesem, zvaným Carrier Extension. Gigabitový Ethernet používá sice stejný minimální rámec o velikosti 64 bajtů, ale zvětšenou hodnotu slot size na 512 bajtů. Že by tyto dvě hodnoty měly být stejné, jak jsme si uvedli o pár odstavců výše? Nemusí, uvědomíme-li si, že slot time je doba vysílání paketů minimální délky, potřebná k zajištění detekce kolizí všemi zúčastněnými uzly. Potřebujeme-li zachovat zpětnou kompatibilitu, tedy stejnou velikost min. rámce, musíme tento rámec vysílat delší dobu. Jak? Jednoduchým doplněním o neplatná data na požadovanou velikost.

V praxi uvedené řešení funguje tedy tak, že je-li rámec menší než 512 bajtů, je doplněn na velikost 512 bajtů neplatnými speciálními symboly, tzv. Carrier Extension. Každý vysílaný rámec tak má min. velikost 512 bajtů a je splněna podmínka dostatečného slot time, doby pro vysílání a detekci kolizí i těch nejmenších rámců.

Doplnění rámce do dostatečné délky zvláštními neplatnými znaky je sice jednoduché řešení, bystřejšího čtenáře již jistě ale napadlo, že je to také značné



plýtvání šířkou pásma. Při nejmenším paketu je doplněno „zbytečných“ 448 doplňujících bajtů.

Dalo by se sice namítnout, že při rychlosti 1 Gb/s nám na pár bajtech nemusí až tak záležet, není to však úplně pravda. Při větším množství malých paketů by přeci jen mohlo docházet k významnému poklesu výkonu sítě (při vysílání pouze 64 bajtových rámců by klesla efektivní přenosová rychlost na pouhých 120 Mb/s!). Samozřejmě jde o extrémní případ, běžný průměr rámců je někde mezi 200 až 500 bajty, i tak by to ale znamenalo datovou propustnost „pouze“ 300 až 400 Mb/s.

Proto bylo řešení s rozšířením malých rámců doplněno o tzv. Packet Bursting, čili posílání rámců ve shlucích. Chce-li stanice poslat více rámců, první rámec je (je-li to nutné) doplněn na potřebnou velikost užitím Carrier Extension. Následující rámce jsou ale vysílány jeden po druhém hned za sebou, s minimální odstupem IPG (Inter-packet gap, mezera mezi jednotlivými vysílanými rámci). Tak jsou ve shluku odvíšlány i malé rámce bez nutnosti jejich doplňování na minimálních 512 bajtů, což by bylo nutné při jejich samostatném odvíšlání. Vysílání shluku rámců pokračuje až do vyčerpání času (burst timer) potřebného pro odvíšlání plného rámce - 1500 bajtů. Tímto způsobem je velice efektivně sníženo ono „plýtvání“ přenosovým pásmem.

### Shrnutí

V této kapitole jste se seznámili s vlastnostmi ATM, FDDI a gigabitového Ethernetu jakožto nejvýznamnějších moderních přenosových technologií používaných v počítačových sítích.

### Kontrolní otázky:

1. Jaký mechanismus používá FDDI pro zvýšení spolehlivosti sítě?
2. Čím se liší gigabitový Ethernet od Fast Ethernetu pracujícího s rychlostí 100 Mb/s?
3. Jaké faktory brání širšímu využití ATM v počítačových sítích?

### Literatura

1. Feibel, W: Encyklopedie počítačových sítí. Computer press 1996
2. Přichystal, O.: Novell NetWare 5 - Podrobná příručka. Computer Press 1999
3. Janeček, Bílý: Lokální sítě. Vydavatelství ČVUT 1997
4. Hunt, C.: Konfigurace a správa sítí TCP/IP. Computer Press, Praha, 1997
5. Peterka, J.: Archiva on-line článků na adrese <http://archiv.czech.net>
6. Dobda, L.: Ochrana dat v informačních systémech. Grada, Praha 1998
7. Garfinkel, S., Spafford, G.: Bezpečnost v UNIXu a Internetu v praxi. Computer Press, Praha, 1998
8. Dostálek, L., Kabelová, A.: Velký průvodce protokoly TCP/IP a systémem DNS. Computer Press, Praha, 1999
9. Dilip C. Naik: INTERNET – standardy a protokoly. Computer Press 1999