Otázka 27 - Y36PSI

Zadání

Počítačová komunikace. Algoritmy a mechanismy směrování v sítích. Řízení toku v uzlech sítě a koncových zařízeních. Protokoly Internetu, adresace, programové rozhraní. Propojování sítí a funkce propojovacích prvků. (Y36PSI)

Slovníček pojmů

- **ISO/OSI model - vypracovala organizace ISO jako hlavní část snahy o standardizaci počítačových sítí nazvané OSI**
- **směrovací metody - slouží pro směrování paketů v síti**
- **RIP - Routing Information Protocol (algoritmus směrování)**
- **OSPF - Open Shortest PAth First (algoritmus směrování)**
- **QoS - Quality of service (kvalita služeb)**
- **RTT - Round Trip Time - Doba, která je vyžadována pro přenos informace od zdroje k cíli a zpět**
- **FIFO, WFQ, leaky bucket,.. - plánovací mechanismy pro řízení toku v uzlech sítě**
- **SLIP - Serial Line IP (protokol internetu)**
- **PPP - Point to Point Protocol**
- **IP - Internet protocol**
- **ICMP -Internet Control Message Protocol**
- **TCP - Transmission Control Protocol - garantuje spolehlivé doručování a doručování ve správném pořadí**
- **UDP - User Datagram Protocol - nepotvrzovaná služba bez spojení**
- **RARP - Reverse Address Resolution Protocol - k získání vlastní IP adresy počítače při znalosti MAC adresy**
- **DHCP - Dynamic Host Configuration Protocol - pro automatické přidělování IP adres jednotlivým osobním počítačům v počítačových sítích**
- **IPv6 - Internet Protocol verze 6**
- **IP adresa - identifikuje síťové rozhraní v počítačové síti, která používá IP (internetový protokol)**
- **broadcast - všesměrová zpráva, kterou v počítačové síti přijmou všechna připojená síťová rozhraní**
- **MAC adresa - Media Access Control - jedinečný identifikátor síťového zařízení, který používají různé protokoly druhé (spojové) vrstvy OSI**
- **maska podsítě - popisuje rozdělení počítačové sítě do podsítí (subnetů).**
- **hub - rozbočovač, aktivní prvek počítačové sítě, který umožňuje její větvení a je základem sítí s hvězdicovou topologií**
- **bridge - spojuje dvě části sítě na druhé (linkové) vrstvě referenčního modelu ISO/OSI**
- **switch - aktivní síťový prvek, propojující jednotlivé segmenty sítě**
- **router - (směrovač) je aktivní síťové zařízení, které procesem zvaným routování přeposílá datagramy směrem k jejich cíli**

Počítačová komunikace

ISO model

- **Aplikační**
- **Prezentační**
- **Relační**
- **Transportní**
- **Síťová**
- **Linková (Spojová)**
- **Fyzická**

TCP/IP model

- **Application (HTTP, FTP, SMTP, DNS, TFTP,...)**
- **Transport (TCP, UDP)**
- **Internet (IP)**
- **Network Access**

Jaké má vlastnosti/Co zajišťuje (fyzická, linková, první, druhá, ...) vrstva OSI modelu?

Fyzická **- specifikuje fyzickou komunikaci, předepisuje vlastnost média**

- **Hlavní funkce poskytované fyzickou vrstvou jsou:**
  - **1 zajistit přenos jednotlivých bitů mezi příjemcem a odesilatelem**
  - **2 Navazování a ukončování spojení s komunikačním médiem**
  - **3 Podílejí se na procesu aby všechny zdroje byly efektivně rozloženy mezi všechny uživatele**
  - **4 Modulace neboli konverze digitálních dat na signály používané přenosovým médiem (a #zpět) (A/D, D/A převodníky)**

Linková (spojová) **- Poskytuje spojení mezi dvěma sousedními systémy**

- **Seřazuje přenášené rámce**
- **stará se o nastavení parametrů přenosu linky**
- **oznamuje neopravitelné chyby**
- **Formátuje fyzické rámce, opatřuje je fyzickou adresou**
- **Na této vrstvě pracují veškeré mosty a přepínače**
- **Poskytuje propojení pouze mezi místně připojenými zařízeními a tak vytváří doménu na druhé vrstvě pro směrové a všesměrové vysílání**

Síťová vrstva **- stará se o směrování v síti a síťové adresování**

- **Poskytuje spojení mezi systémy, které spolu přímo nesousedí**
- **Obsahuje funkce, které umožňují překlenout rozdílné vlastnosti technologií v přenosových sítích**
- **Síťová vrstva poskytuje směrovací funkce a také reportuje o problémech při doručování dat**
- **Veškeré směrovače pracují na této vrstvě a posílají data do jiných sítí**
- **Zde se již pracuje s hierarchickou strukturou adres**
- **Nejznámější protokol pracující na 3. vrstvě bude Internetový Protokol (IP). Poskytuje adresaci a směrování dat přes mezilehlé prvky, jednoznačnou adresu v rámci sítě (síťovou adresu), síťovou službu se spojením, síťovou službu bez spojení**
- **stará se o to, aby se jednotlivé pakety dostaly od odesilatele až ke svému skutečnému**

příjemci, přes případné směrovače resp. brány. Vzhledem k nespojovanému charakteru přenosů v TCP/IP je na úrovni této vrstvy zajišťována jednoduchá (tj. nespolehlivá) datagramová služba. Datagramová služba znamená posílání paketů do sítě bez jistoty toho, že se po cestě nějaký paket ztratí, či dojde později. Pakety tak nejdou jednou cestou, ale tak, jak je jednotlivé uzly posílají s ohledem na co nejkratší nebo nejrychlejší aktuální cestu. Pakety tak nemusí dojít ve stejném pořadí, jako jsme je odeslali.

- **u této vrstvy lze využít i virtuálního kanálu, což je jakýsi tunel mezi koncovými zařízeními. Vytvoření takového kanálu probíhá tak, že počáteční zařízení pošle zaváděcí paket, který obsahuje informace, kam má být jednotlivými uzly sítě směrován a informaci o tom, že uzly si mají toto směrování pamatovat. Každý další paket, který má stejnou identifikační informaci v hlavičce tak poté uzly sítě rozpoznají, přiřadí ho k vytvořenému virtuálnímu kanálu, otevřou port toho kanálu a skrz tento kanál (resp. port, na kterém je kanál vytvořen) ho pošlou. Máme tak jistotu, že pakety dojdou ve stejném pořadí, v jakém jsme je poslali, protože nemají možnost se cestou rozejít, ale jdou vždy tím jedním kanálem, tzn. jednou cestou.**

***Transportní (přenosová) vrstva* - zajišťuje přenos dat mezi koncovými uzly**

- **Hlavními protokoly této vrstvy jsou TCP a UDP**
- **poskytuje rozklad dat na pakety**
- **uspořádání dat podle pořadí**
- **multiplexuje a demultiplexuje data mezi transportními spoji, transportní adresy (adresa, port), koncové řízení toku**

***Relační vrstva (session layer)* - organizuje a synchronizujedialog mezi relačními vrstvami a řídí výměnu dat mezi nimi**

- **Umožňuje vytvoření a ukončení relačního spojení**
- **synchronizace a obnovení spojení**
- **oznamovaní výjimečných stavů**
- **Vytváření rozhraní pro aplikace a synchronizace spojení pomocí transakcí**

Prezentační

- **transformace dat do tvaru použitelného aplikacím**
- **zabývá se strukturou dat, řeší například komprese, kódování (ASCII/EBCDIC), šifrování, big a little endian**

Aplikační

- **Poskytuje aplikacím přístup k nižším vrstvám**

Algoritmy a mechanismy směrování

Směrovací metody

Popisují chování paketu v určitém bodě sítě (kam půjde paket dál).

1. Záplavové - každý uzel kromě příjemce vyšle přijatý paket do všech směrů, krom směru ze kterého ho dostal

- **+ nejkratší cesta**

- **+ spolehlivost**
- **- zahlcení sítě**
- **- potřeba likvidovat nadbytečné pakety**

2. Náhodné - odesílání paketu na náhodný výstup

- **+ odolnost proti změně topologie**
- **+ při částečné znalosti topologie lze její chování upravit, pak už je užitečná**
- **- nezaručuje omezenou dobu doručení**
- **- potřeba dodatečné informace**

3. Izolované - bere v úvahu pouze lokální informace a už ne informace ostatních uzlů sítě

1. ***Horký brambor***

- **odeslání paketu na výstup s nejkratší frontou**
- **nezaručuje omezenou dobu doručení**
- **potřeba dodatečné informace**

1. ***Zpětné učení***

- **využití informací o čase / počtu průchodů v paketu, tabulka obsahuje odesílatele, nejkratší čas, směr odkud paket přišel**
- **nutné kombinovat s jinou metodou**
- **pomalá konvergence při chybě - potřeba zapomínání**

4. Adaptivní - směrovací tabulky se přizpůsobují momentálnímu stavu

- **+ bezpečnost**
- **+ reakce na změnu**
- **možnost kombinace s předchozími metodami**
- **link-state (OSPF)**
- **distance-vector (RIP)**

5. Statické - nastavení tabulek při návrhu sítě, vhodné určit i alternativní směry

6. Hierarchické - rozdělení adresy: prefix oblasti + adresa uzlu uvnitř

- **adresování respektuje topologii**

Algoritmy směrování

Distance vector algoritmy

- výměna kompletních směrovacích tabulek

- **vzdálenost měří pouze pomocí předem definovaných hodnot, nereagují na aktuální změny zatížení**
- **routery si vyměnují kompletní směrovací tabulky → větší zátěž sítě**
- **pomalá konvergence - obzvlášť při výpadku**
- **tento typ zastupuje např.: Ford-Fulkersonův algoritmus**
- **příklady: RIP, IGRP, EIGRP, BGP**

(pozn. jakub dvorak: BGP je path vector protokol, EIGRP neposila cele tabulky, je hybridni mezi link state a distance vector. Distance vector pouziva spis Bellman-forduv algoritmus,

EIGRP ma tzv. DUAL algoritmus.)

Link state algoritmy

- výměna změn sítě → každý uzel má informace o síti

- **rychlá konvergence**
- **nízké zatížení sítě**
- **příklad: OSPF**

RIP - Routing Information Protocol

- **podmínkou pro správné fungování protokolu IP, který zajišťuje přenos na úrovni síťové vrstvy (a hlavně tzv. směrování, neboli volbu směru dalšího přenosu), je existence aktuálních a korektních informací o momentální topologii celé soustavy sítí - má-li se protokol IP rozhodnout, kudy poslat dál nějaký datový paket, musí mít informaci o tom, kudy vede jaká cesta. Skutečná topologie každé soustavy vzájemně propojených sítí (a samozřejmě i celého Internetu) se však může dynamicky měnit, a tak je vhodné, aby se i protokol IP průběžně „dozvídal" o všech změnách. Právě k tomuto účelu slouží protokol RIP, který průběžně rozesílá mezi všechny uzly sítě informace o tom, jaká je momentální topologie celé soustavy vzájemně propojených sítí. Právě tyto průběžně aktualizované informace pak využívá protokol IP ke svým konkrétním rozhodnutím v rámci směrování.**
- **techniky zrychlení konvergence (tzn. adaptace na vyřazení nějakého prvku ze sítě)**
  - **split horizon - uzel nepředává nové informace zpět uzlu, od kterého je získal**
  - **poison reverse - uzel nepředává nové informace zpět uzlu, od kterého je získal a místo toho mu podstrčí hodnotu nekonečna (16)(Pozn. Vojtech Kral: podle mě je to spíše, že když routeru spadne cesta, tak ostatním rozdá 16)**
  - **hold down - předchází obnovám vadných směrovacích cest v pravidelných updatech. Pokud směrovač „spadne", sousedícím směrovačům chybějí pravidelné updaty. Tyto směrovače pak vypočítají novou cestu a informují „sousedy" o změnách v síti. O nových updatech není okamžitě informován každý síťový prvek, tak je možné pro zařízení, které bylo právě informováno o poruše, poslat klasickou update message, která propaguje problematickou cestu jako platnou, tomu zařízení, které bylo právě informováno o změnách.**

OSPF - Open Shortest Path First

- **OSPF je typickým představitelem směrovacího protokolu typu Link State. Vytváří tedy v paměti směrovače kompletní mapu celé sítě, označovanou jako topologická databáze (někdy se jí říká Link State Database). Nad touto databází potom pomocí algoritmu označovaného jako Shortest Path First (SPF) provádí výpočty potřebné k nalezení nejvýhodnější cesty do jednotlivých sítí.**
- **Uplatňuje se rozdělení na oblasti. 0 je páteřní oblast. Komunikace mezi dvěma autonomními systémy musí vždy jít přes páteř**
- **Protože protokol RIP je relativně staršího data a vzniknul s představou, že bude průběžně distribuovat informace o topologii takové soustavy vzájemně propojených sítí, která není příliš velká - určitě ne v takovém rozsahu, jaký se týká většiny částí dnešního Internetu(rozsah informací, které protokol RIP „roznáší" o každém jednotlivém spoji je relativně velký, a hlavně s růstem celé soustavy sítí velmi rychle roste), je tento protokol dnes stále častěji nahrazován protokolem OSPF, který slouží**

přesně stejnému účelu. Průběžně distribuuje směrovací informace, ale má menší režii a je lépe škálovatelný, tj. tato jeho režie s růstem celé soustavy sítí neroste tak rychle, jak by rostla v případě protokolu RIP.

Řízení toku v uzlech sítě a koncových zařízeních

Řízení toku pomocí QoS - Quality of Service (kvalita služeb)

- **snaží se zaručit koncovému uživateli doručení dat v potřebné kvalitě**
- **uplatňuje se v přenosu multimédií, IP telefonii atd.**
- **zajištěna pomocí specfikace TOS (Type of Service) v hlavičce IP protokolu (obvykle se ale ignoruje)**
- **implementuje se pomocí pozdržení potvrzení, což zvýší RTT (Round Trip Time = doba, za kterou přijde potvrzení odeslaného datového segmentu), nebo pomocí změny velikosti TCP okénka**

Řízení toku v protokolu TCP

- **efektivně lze omezovat pouze odchozí tok dat**
- **pro příchozí tok musíme použít nepřímé metody**
- **můžeme pozdržet potvrzení a tím prodloužit RTT (Round Trip Time) - pozor na timeout, který působí opětovné odeslání dat**
- **můžeme měnit velikost okénka**

Plánovací mechanismy

1. **FIFO - First in, first out (nejjednodušší přístup - pakety jsou zpracovávány v pořadí jejich příchodu)**
2. **prioritní FIFO (pakety jsou rozděleny do dvou prioritních tříd: důležité pakety a méně důležité pakety. Pro každou třídu existuje speciální fronta, přičemž platí, že libovolný paket z fronty důležitých paketů má přednost před libovolným paketem z fronty méně důležitých paketů. Vzájemné mezi pakety z každé skupiny je jednotlivými frontami zachováno. Front (a tím pádem i prioritních tříd) může být i více než 2)**
3. **WFQ - Weighted fair queuing (každý datový tok má svojí frontu. Pokud jeden tok např. posílá rychleji nebo větší pakety, než ostatní, projevuje se to jen na zpracování onoho toku a netrpí tím ostatní)**
4. **(Pozn. Vojtech Kral: Leaky Bucket - metaforicky se dá přirovnat ke kyblíku s dírou ve dně. Packety odtékají rychlostí, podle díry ve dnu a pokud kyblík přeteče, pakety se zahazují. Neumožní narozdíl od Token bucketu špičky v sítovém přenosu, protože díra ve dnu je pořád stejná)**
5. **Token Bucket - v zásobníku na tokeny je určitý počet tokenů. Pokud přijde paket o velikosti N bytů na řadu, je odebráno ze zásobníku N tokenů. Pokud jejich počet nestačí, je s paketem nakládáno dle implementace (zahodí se atd), jinak je odeslán**
6. **Round Robin (funguje jako bariéra v datovém přenosu. Mechanismus Round Robin postupně po dobu krátkého časového kvanta propouští pakety z jednotlivých datových toků, které dorazily k bariéře, přičemž některé toky mohou mít přednost)**

Protokoly internetu

Protokoly na linkové vrstvě

Zaručují přenos dat mezi přímo propojenými systémy.

1. SLIP - Serial Line IP

- **definuje pouze zapouzdření paketů na sériové lince**

2. CSLIP - Compressed SLIP

- **pouze redukce záhlaví TCP a IP**
- **přenáší se změny položek záhlaví**
- **ignorují se změny záhlaví**

3. PPP - Point to Point Protocol

- **protokol umožňující autentizaci, šifrování a kompresi**

4. Etherner II (DIX)

Protokoly na síťové vrstvě

1. IP - Internet Protocol

k čemu se používá

- **základní protokol síťové vrstvy a celého Internetu**
- **vysílání datagramů na základě síťových IP adres obsažených v jejich záhlaví**
  - **každý datagram je samostatná datová jednotka, která obsahuje všechny potřebné údaje o adresátovi i odesilateli a pořadovém čísle datagramu ve zprávě**
  - **datagramy putují sítí nezávisle na sobě a pořadí jejich doručení nemusí odpovídat pořadí ve zprávě**
  - **doručení datagramu není zaručeno, spolehlivost musí zajistit vyšší vrstvy (TCP, aplikace)**
- **poskytuje vyšším vrstvám síťovou službu bez spojení**

funkce a činnosti vykonávané protokolem IP

- **adresace koncových uzlů a sítí v IP intersíti**
- **vytváření IP paketů z paketů protokolů vyšší vrstvy**
- **směrování IP paketů přes IP intersíť**
- **fragmentace IP paketů**
- **stará se o segmentaci a znovusestavení datagramů do a z rámců podle protokolu nižší vrstvy (např. Ethernet)**
- **protokol IP je základním přenosovým prostředkem pro protokoly TCP/IP**

fragmentace

- **umožňuje vložení IP paketu do kratších rámců nižší vrstvy (MTU - Maximum Transmission Unit)**
- **fragmentaci provádí libovolný směrovač**

- **defragmentaci provádí koncový uzel**
- **možnost zakázání fragmentace**

2. ICMP - Internet Control Message Protocol

- **slouží k přenosu řídících hlášení, která se týkají chybových stavů a zvláštních okolností při přenosu(př. ping, traceroute)**
- **přenos chybových i řídících informací**
  - **testování dostupnosti (Echo Req/Rep)**
  - **řízení zahlcení a toku**
  - **změna směrovací tabulky**
  - **informace o masce**
  - **časová synchronizace**
- **omezená implementace**
- **zahazování z bezpečnostních důvodů**

některé služby, např. echo, redirect...

- **Echo Request ... požadavek na odpověď, každý prvek v síti pracující na IP vrstvě by na tuto výzvu měl reagovat. Často to z různých důvodů není dodržováno.**
- **Echo Reply ... odpověď na požadavek**
- **Destination Unreachable ... informace o nedostupnosti cíle, obsahuje další upřesňující informaci**
  - **Net Unreachable ... nedostupná cílová síť, reakce směrovače na požadavek komunikovat se sítí, do které nezná cestu**
  - **Host Unreachable ... nedostupný cílový stroj**
  - **Protocol Unreachable ... informace o nemožnosti použít vybraný protokol**
  - **Port Unreachable ... informace o nemožnosti připojit se na vybraný port**
- **Redirect ... přesměrování, používá se především pokud ze sítě vede k cíli lepší cesta než přes defaultní bránu. Stanice většinou nepoužívají směrovací protokoly a proto jsou informovány touto cestou. Funguje tak, že stanice pošle datagram své, většinou defaultní, bráně, ta jej přepošle správným směrem a zároveň informuje stanici o lepší cestě.**
  - **Redirect Datagram for the Network ... informuje o přesměrování datagramů do celé sítě**
  - **Redirect Datagram for the Host ... informuje o přesměrování datagramů pro jediný stroj**
- **Time Exceeded ... vypršel časový limit**
  - **Time to Live exceeded in Transit ... během přenosu došlo ke snížení TTL na 0 aniž byl datagram doručen**
  - **Fragment Reassembly Time Exceeded ... nepodařilo se sestavit jednotlivé fragmenty v časovém limitu(např pokud dojde ke ztrátě části datagramů)**

protokoly sloužící pro přidělení IP adresy

3. ARP - Address Resolution Protocol

- **spolu s IP protokolem se používá k získání ethernetové MAC adresy sousedního stroje z jeho IP adresy**

4. RARP - Reverse Address Resolution Protocol

- **přidělení adresy bezdiskové stanici**
- **nepoužívá se**

5. BOOTP - Bootstrap Protocol

- **starší protokol**
- **statické přidělení parametrů**
- **Protokol BootP slouží k získaní konfigurace. Klient ve svém požadavku (BOOTREQUEST) uvádí svou hardwarovou adresu a volitelně svou IP adresu, jméno serveru a požadovaný boot soubor. Server v odpovědi (BOOTREPLY) pošle požadovanou IP adresu, IP adresu gateway, svou IP adresu a jméno.**

6. DHCP - Dynamic Host Configuration Protocol

- **DHCP protokol umožňuje prostřednictvím jediného DHCP serveru nastavit všem stanicím sadu parametrů nutných pro komunikaci v sítích používajících rodinu protokolů TCP/IP včetně parametrů doplňujících a uživatelsky definovaných. Významným způsobem tak zjednodušuje a centralizuje správu počítačové sítě (například při přidávání nových stanic, hromadné změně parametrů, nebo pro skrytí technických detailů před uživateli). DHCP servery mohou být sdruženy do skupin, aby bylo přidělování adres odolné vůči výpadkům.**

Protokoly na transportní vrstvě

1. UDP - User Datagram Protocol

- **nepotvrzovaná služba bez spojení**

použití (i příklady aplikací)

- **stream - proud - tok dat**
- **vysílání internetových rádií a televizí(nezáleží na tom, zda nějaký packet nepřijde, prostě se jede dál)**

vlastnosti

- **rychlejší jak TCP**
- **neřeší kontrolu došlých packetů a ani pořadí ve kterém přicházejí, může se stát že z řady odeslaných packetů 1,2,3,4,5 vám dojdou ve stavu 4,2,5,3 s tím že 1 vůbec nedošla**

porty

- **UDP používá porty, aby bylo možné rozlišit v počítači jednotlivé aplikace a správně jim doručit data, i když jich komunikuje v počítači více. Port je 16 bitová hodnota, která umožňuje používat porty z rozsahu 0-65535. Port 0 je rezervován, ale je možné ho použít, pokud odesílající proces neočekává žádnou odpověď**
- **Porty 1-1023 jsou tzv. dobře známé (anglicky well known ports) a na Unixech a odvozených operačních systémech jsou potřeba práva uživatele root, aby je bylo možné použít. Porty 1024-49151 jsou registrované porty. (Nemělo by se jich používat pro jiné aplikace než jsou registrované u IANA.) Porty 49152-65535 jsou používány pro komunikaci klienta se serverem**

2. TCP - Transmission Control Protocol

použití (i příklady aplikací)

- **posílání souborů (záleží na aplikaci, třeba u Torrentů funguje jak TCP tak UDP)**
- **HTTP protokol, neboli WWW stránky**

vlastnosti

- **kontrola došlých dat**
- **využívá algoritmu klouzající okénko**

porty

- **Několik příkladů: FTP (port 21 a 20), SMTP (port 25), DNS (port 53) a HTTP (port 80). Registrované porty jsou typicky používané aplikacemi koncových uživatelů při otevírání spojení k serverům jako libovolné čísla zdrojových portů, ale také mohou identifikovat služby. Dynamické/privátní porty mohou být také používány koncovými aplikacemi, ale není to obvyklé**

TCP spojení

- **navázání - třífázový protokol - klient pošle SYN na server, ten pátky pošle SYN + ACK a klient zpátky pošle ACK - v ideálním případě se spojí**
- **komunikace - řazení dat do správného pořadí, znovuposílání ztracených rámců, odstranění duplikací, kontrola přetečení zásobníku pro zpracování, využití klouzavého okénka**
- **ukončení - třífázový protokol - klient pošle FIN na server, ten pátky pošle FIN + ACK a klient zpátky pošle ACK - v ideálním případě spojení skončí (Pozn. Vojtech Kral: řekl bych, že 4 cestné ukončení, protože u TCP může se uzavřít spojení jen v jednom směru a pak je toto nutné uzavřít i v druhém směru)**
- **rámcová znalost řízení toku - změna velikosti okénka, Nagleův algoritmus (pro využití kapacity kanálu při malém zatížení), odesílání potvrzení po větších blocích, přenastavení timeoutu**

rozdíly mezi TCP a UDP

- **TCP**
  - **spojově orientovaný protokol**
  - **Spojení může otevřít klient nebo server a pak mohou už být posílána jakákoliv data oběma směry**
  - **spolehlivost – TCP používá potvrzování o přijetí, opětovné posílání a překročení časového limitu. Pokud se jakákoliv data ztratí po cestě, server si je opětovně vyžádá. U TCP nejsou žádná ztracená data, jen pokud několikrát po sobě vyprší časový limit, tak je celé spojení ukončeno**
  - **zachování pořadí – jestliže se odešlou 2 zprávy, jedna po druhé, první dorazí nejdřív k serveru. Pokud data dorazí ve špatném pořadí, TCP vrstva se postará o to aby některá data pozdržela a finálně je předala správně seřazená**
  - **vyšší režie – TCP protokol potřebuje tři pakety jen pro otevření spojení, což však umožňuje zaručit spolehlivost celého spojení**
- **UDP**
  - **jednodušší protokol založený na odesílání nezávislých zpráv**
  - **bez záruky – protokol už neumožňuje ověřit jestli došla zamýšlenému příjemci. Datagram se může po cestě ztratit. UDP nemá žádné potvrzování, přeposílání**

ani časové limity
- nezachovává pořadí – jestliže odešleme dvě zprávy jednomu příjemci, nemůžeme předvídat v jakém pořadí budou doručeny
- jednoduchost – nižší režie, než u TCP (není zde řazení, žádné sledování spojení atd.)

IPv6

formát datagramu:

datagram v IPv6 odráží snahu tvůrců vytvořit co nejmenší záhlaví a to pouze se základním obsahem. Tvůrci přesunuli málokdy používané části datagramu z IPv4 do tzv. „rozšiřujících hlaviček" v IPv6. Jde zejména o položky potřebné k fragmenaci datagramu. které byly v každém IPv4 datagramu. Pro základní záhlaví byla stanovena nová pevná velikost, takže položka „velikost záhlaví" byla vynechána. Při dnešní nízké chybovosti a vysoké propustnosti linek, ztratilo opodstatnění pole „kontrolní suma". Principy použité v IPv4 (Fragmentace, Směrování) prošli jen nepatrnými změnami

řetězení hlaviček:

k základnímu záhlaví v IPv6 je možné připojit i rozšiřující hlavičky a to prostřednictvím položky „další hlavička" v základním záhlaví. Tuto položku obsahuje i každá rozšiřující hlavička, přičemž je možné tímto způsobem připojit několik doplňujících záhlaví. Hodnota položky „další hlavička" taktéž zastupuje položku „protokol" z IPv4, kterou se určuje typ údajů nesený za celým záhlavím

adresace

- **délka adresy – 128b**
- **druhy adres**
  - **Individuální (unicast)-označují jedno rozhraní připojeného počítače či zařízení**
  - **Skupinové (multicast)-představují adresu skupiny síťových rozhraní. Paket se skupinovou cílovou adresou bude dopraven všem členům skupiny. Tyto adresy se používají nejčastěji pro šíření zvukového či obrazového signálu, videokonference a podobně**
  - **Výběrové (anycast)-také označují skupinu síťových rozhraní, ale datagram bude dopraven jen na jedno z nich (zpravidla to nejbližší). Výběrové adresy umožňují například realizovat některé speciální služby - klient odešle datagram s obecnou adresou a některý z dostupných serverů se jej ujme**
- **broadcast adresy nejsou podporovány**
- **zápis adres**
  - **FEDC:1234:0000:ABCD:0F12:0000:0000:4567**
- **zkracování**
  - **FEDC:1234::ABCD:F12:0:0:4567**
  - **FEDC:1234:0:ABCD:F12::4567**
- **prefixy**
  - **FEDC:1234:0000:ABC0:0000:0000:0000:0000/60**
  - **FEDC:1234:0:ABC0::/60**

fragmentace:

- **MTU (Maximum transfer unit) je hodnota udávající maximální velikost přenosové jednotky na úrovni vrstvy síťového rozhraní (nejnižší vrstvy) komunikačního modelu TCP/IP.**

- **Protokol IP umožňuje rozdělení velkého IP paketu na menší celky (fragmenty). Každý fragment je samostatný IP paket - má svou (novou) IP hlavičku. Každý fragment potom putuje k cíli samostatně, nezávisle na ostatních. U příjemce jsou jednotlivé fragmenty poskládány a je z nich sestaven původní celek - IP paket. Každý fragment je samostatný IP paket, z čehož vyplývá, že může být podle potřeby opět rozdělen na další fragmenty**
- **provádí se pouze u odesilatele**
- **informace o fragmentaci v rozšiřující hlavičce**
- **hlavičky až k fragmentační – nefragmentovatelné**
- **minimální MTU (Maximum Transmission Unit) pro IPv6 je 1280B (ale očekává se, že klient provede MTU Path discovery)**
- **vyhledávání MTU cesty (MTU Path discovery)**
  - **ICMPv6**
  - **pravidelné opakování (cca 10min)**
  - **nemusí se implementovat**

automatická konfigurace:

- **dva typy automatické konfigurace**
- ***stavová konfigurace* - charakterizuje jí v IPv4 používaný systém DHCP, který je upravený pro potřeby IPv6 a nese název DHCPv6**
  - **nevyužívá broadcast**
  - **1 DHCP Unique Identifier (DUID)**
    - **jednoznačně identifikuje uzel**
    - **linková adresa a čas**
    - **přiděleno výrobcem**
  - **2 Identity Association (IA) - jednoznačně identifikuje rozhraní**
  - **3 vyhledání všech serverů (FF02::1:2 – adresa agenta) –> solicit - advertise**
  - **4 oslovení zvoleného serveru podle DUID (FF02::1:2)! –> request – reply**
  - **5 obnovení - renew, rebind, release, confirm**
  - **6 rekonfigurace vyvolaná serverem - reconfigure**

**** bezstavová konfigurace***

- **Kreativní novinkou je konfigurace bezstavová, která nevyžaduje žádné servery. Jejím základním pilířem je tak zvané objevování sousedů (neighbor discovery). Základní myšlenka je celkem prostá: každý směrovač v určitých intervalech rozesílá do sítí, k nimž je připojen, tak zvané ohlášení směrovače. V něm jsou obsaženy základní informace - především prefixy adres dané sítě a zda on sám může sloužit pro předávání paketů ven (jako implicitní směrovač, default gateway)**
- **Z ohlášení směrovačů (o které může při startu aktivně požádat pomocí výzvy směrovači) se počítač dozví, jaké adresy používá zdejší síť. K nim si doplní zbývající část (typicky 64 bitů), která se jednoznačně generuje z jeho Ethernetové adresy. Tak získá platné IPv6 adresy pro své rozhraní. Také si vytvoří seznam implicitních směrovačů, kterým bude předávat pakety směřující mimo síť. Pokud je jich víc, zpočátku je prostě střídá a směrovací tabulku si postupně vylepšuje na základě jejich upozornění (ICMP přesměrování), pokud paket k určitému cíli poslal nevhodným směrovačem.**
- **1 ohlášení směrovače**
- **2 určení adresy**

- **3 konfigurace směrování**

mobilita:

- **Základní charakteristikou mobilních zařízení je, že za provozu přecházejí z jedné sítě do druhé (např. když komunikujete ze svého notebooku během cesty z Prahy do Vídně) a tudíž mění svou IP adresu. To nepředstavuje problém, pokud spojení navazuje mobilní počítač - prostě použije svou aktuální adresu**
- **Potíže nastanou, pokud se někdo chce spojit s cestujícím počítačem. Řešení je poměrně prosté. Každý počítač má svou domácí síť a jí odpovídající domácí IP adresu. Tato adresa je zanesena v DNS a tu také použije externí stroj při pokusu o spojení**
- **Pokud je počítač právě na cestách, zastupuje jej domácí agent. Tuto roli hraje jeden ze směrovačů v domácí síti mobilního stroje (podpora mobility zahrnuje i způsob jeho automatického výběru, aby nemusel být konfigurován staticky). Domácí agent na sebe přesměruje data určená mobilnímu stroji (při objevování sousedů odpovídá místo něj). Mobilní uzel průběžně informuje svého domácího agenta o aktuální IP adrese**
- **Navázání spojení tudíž probíhá následovně: externí počítač zašle paket s žádostí o navázání spojení na domácí (trvalou) adresu mobilního uzlu. Jeho domácí agent ji zachytí a předá tunelem mobilnímu uzlu. Ten odpoví žadateli a zároveň zahájí proces nazvaný optimalizace cesty. Jeho cílem je informovat protějšek o aktuální adrese mobilního uzlu. Situaci komplikují bezpečnostní mechanismy, aby se kdokoli nemohl prohlásit za jiný počítač na cestách. Jakmile se protějšek dozví aktuální adresu mobilního uzlu, probíhá další komunikace přímo mezi nimi**

rozdíly oproti IPv4

zjednodušení hlavičky

- **zjednodušuje původní hlavičku odstraněním přebytečných polí a zřetězuje volitelné podhlavičky se základní IP hlavičkou**
- **mimo základní hlavičky se používají volitelné podhlavičky pro směrování, fragmentaci a ověřování přístupu - např. podhlavičky s názvem Hop-by-Hop Options, Routing, Fragment, Destination Options, Authentication, Encalupsulating Security Payload**
- **tyto se používají jen v případě požadavku na danou funkci**

rozšíření adresního prostoru IP adres

- **IPv6 rozšiřuje adresní prostor z původních 32 bitů na 128 bitů. (5.1028 IP adres na jednoho současného člověka)**

automatická konfigurace uzlů

- **protokol má zapracované mechanismy automatického předělování konfiguračních údajů sítě jeho jednotlivým uzlům**
- **odpadá tak manuální předělování IP adres na lokálních uzlech, respektive směrovačích sítě**

bezpečnostní procedury

- **nový protokol má přímo zabudované bezpečnostní procedury ověřování přístupu (Authentication) a kodování (Encryption) na úrovni IP komunikace**
- **IPv6 umožňuje volitelný výběr bezpečnostních metod a parametrů, implicitně se však používají metody autorizace přístupu MD5 a kódování podle DES (Data Encryption**

Standard)

- **IP má bezpečnostní vlastnosti implementované prostřednictvím volitelných podhlaviček AH (Authentication Head) a ESP (Encapsulating Security Payload)**

podpora multimediálních aplikací

- **nárust aplikací požadujících komunikaci v reálném čase (real-time) v síti Internet si vyžádal i zvýšení podpory tohoto druhu komunikace na straně IP protokolu - IP za tímto účelem používá motodu označení toku návěští (Flow label)**
- **tokem rozumíme IP pakety přenášené mezi zdrojem a cílem v IP síti se speciálními požadavky na přenos**
- **přiřazením číselného návěští danému toku prostřednictvím protokolu RSVP (Resource Reservation Protocol), může IP směrovač obsluhovat přenos paketů různých toků odlišným způsobem**

Adresace

Protokol TCP/IP používá pro adresování IP adresy a masky podsítě, pomocí kterých se síť rozděluje do logických bloků - subnetů.

IP adresa

je logická adresa zařízení v síti IP. Skládá se ze 4 částí zvaných octety, každá část je veliká 8 bitů, a zapisuje se oddělená tečkou. Adresa se většinou zapisuje v dekadické formě, ale pro výpočet je jasnější binární zápis. Teoreticky je tedy adresní rozsah od 0.0.0.0 do 255.255.255.255. Příkladem IP adresy je třeba 68.12.5.10.

Broadcast IP adresa

je adresa, na kterou se posílá komunikace v případě brodcastového vysílání. Jedná se o adresu, která má binárně samé jedničky, je to tedy IP adresa 255.255.255.255. Tato adresa určuje všechny klienty v síti.

Maska podsítě (Subnet mask)

nám pomáhá určit rozdělení sítě na podsítě. Určuje, která část IP adresy je síťová, a která pro hosty. Zápis je stejný jako u IP adresy, ale platné hodnoty jsou pouze ty, které mají v binárním tvaru zleva jedničky a zprava nuly (pokud se zleva na některé pozici objeví nula, dále již musí následovat pouze nuly). Jedničky v masce jsou tzv. network ID a je to část, která je pro daný subnet stále stejná. Nuly jsou tzv. host ID a tedy část, která je proměnná a určuje adresu hosta v daném subnetu. Příkladem jednoduché masky je 255.255.255.0, ta určuje, že prvních 24 bitů adresy je network ID a posledních 8 bitů je hostovská část.

Subnet mask se může zapisovat také ve zkrácené formě, které se říká CIDR notace. Ta se zapisuje jako IP adresa následovaná lomítkem (/) a číslem, které reprezentuje počet jedničkových bitů v masce podsítě v binární formě. Protože celkový počet bitů v masce je 32, tak počet nul je 32 - počet jedniček. Příklad CIDR notace je 10.0.5.2/20 a tedy maska je 255.255.240.0.

Adresní prostor

byl rozdělen do základních pěti tříd

třída	určující bity	rozsah adres	maska
class A	0	0 - 127.x.x.x	255.0.0.0

class B	10	128 - 191.x.x.x	255.255.0.0
class C	110	192 - 223.x.x.x	255.255.255.0
class D	1110	224 - 239.x.x.x	255.255.255.255
class E	1111	240 - 255.x.x.x	

Příklad na adresaci sítě

Máme adresu 10.0.5.2/20

maska binárně 11111111 11111111 11110000 00000000

maska dekadicky 255 255 240 0

Použijeme třetí octet, kde jsou 4 jedničky a 4 nuly (plus 8 nul ve čtvrtém octetu). Hostů v každém subnetu může být 212 - 2 = 4094. (proč -2? Jedna adresa je broadcast a druhá adresa sítě) V rámci třetího octetu můžeme vytvořit 24 = 16 podsítí.

Výpočet základní adresy sítě

Pokud máme adresu hosta a masku podsítě, můžeme jednoduše spočítat základní adresu podsítě. Pokud vezmeme binárně adresu hosta a masku sítě a provedeme bitový součin (AND), dostaneme adresu sítě. Příklad pro 10.217.123.7/20

***IP adresa* dekadicky 10 217 123 7 binárně 00001010 11011001 01111011 00000111**

***Maska podsítě* dekadicky 255 255 240 0 binárně 11111111 11111111 11110000 00000000**

Provedeme bitový AND a dostaneme adresu sítě binárně 00001010 11011001 01110000 00000000 dekadicky 10 217 112 0

Výpočet broadcast adresy subnetu

Broadcast adresu subnetu, v kterém se nachází klient, spočítáme jednoduše. V IP adrese hosta změníme bity v hostovské části na 1. Matematicky řečeno vezmeme IP adresu a provedeme bitový součet (OR) s negovanou (NOT) maskou podsítě. Pro předchozí příklad (síť 10.217.123.7/20), kde prvních 20 bitů je network ID a zbylých 12 je hostovská část, to tedy je:

IP adresa

dekadicky 10 217 123 7

binárně 00001010 11011001 01111011 00000111

Invertovaná (bitová negace) maska

binárně 00000000 00000000 00001111 11111111

dekadicky 0 0 15 255

Provedeme bitové OR a máme broadcast adresu subnetu

binárně 00001010 11011001 01111111 11111111

dekadicky 10 217 127 255

Vyřešená adresace ze zkoušky

Nejprve si zjistím počet zařízení v každé síti. Poté si určím masku podsítě, tj, mám např. 7 PC + adresa podsítě + broadcast, potřebuji tedy min. 9 adres, to znamená masku /28 (pro 16 zařízení, menší mít nemohu). Poté již čísluju jednotlivá zařízení a takto postupuju postupně vzůru po síti až k internetu. Tam mám pouze 2 zařízení (router a internet + broadcast + síť),

potřebuji tedy masku /30 pro 4 zařízení. 14psi_reseni.pdf

Druhá řešená písemka

13psi_reseni.pdf

Velice dobře vysvětleno také v následujících odkazech

http://www.security-portal.cz/clanky/rozd%C4%9Blov%C3%A1n%C3%AD-ip-s%C3%ADt%C3%AD [http://www.security-portal.cz/clanky/rozd%C4%9Blov%C3%A1n%C3%AD-ip-s%C3%ADt%C3%AD] http://www.abclinuxu.cz/clanky/site/jemny-uvod-do-adresace-v-protokolu-ip-verze-4 [http://www.abclinuxu.cz/clanky/site/jemny-uvod-do-adresace-v-protokolu-ip-verze-4]

Programové rozhraní

Připojení počítače k místní síti je možné pouze tehdy, jsou-li k dispozici následující technické (HW) a programové (SW) součásti:

- **přenosové médium (kabel)**
- **síťová karta (tzv. adaptér)**
- **obslužný program pro adaptér (tzv. driver)**
- **síťový operační systém**

Většina současných výrobců sítí má snahu o to, aby jejich síť mohla spolupracovat s co největším množstvím ostatních (konkurenčních) sítí. Z těchto důvodů dochází ke snaze vytvořit síť (v rámci možností) modulární, aby byla možná co největší zaměnitelnost jednotlivých komponent od více výrobců. V praxi se tento trend dostal až tak daleko, že ve více sítích je možné použití síťových karet různých výrobců – příkladem je např. NOVELL. Výrobci sítí proto mají snahu své výrobky navrhovat tak, aby vyhovovaly tzv. VRSTVOVÉ STRUKTUŘE KOMUNIKACE V SÍTI podle doporučení organizace ISO.

Programové vybavení

Každé komunikující zařízení musí být vystrojeno příslušným programovým vybavením (rozprostřeným přes síť) To - na nižší úrovni - řídí tok dat mezi koncovými zařízeními prostřednictvím pasivních prostředků sítě , při spolupůsobení aktivních prostředků sítě. Na vyšší úrovni zajišťuje větší zabezpečenost přenosu (protokoly) a umožňuje vstup do síťových aplikací..

Programové vybavení na nižší úrovni

Programové vybavení na nižší úrovni: zajišťuje spolupráci koncových zařízeních s komunikačními zařízeními, prostřednictvím "protokolů" zajišťuje výměnu dat mezi koncovými zařízeními a aktivními prostředky sítě. Je většinou závislé na typu komunikačního zařízení a počítačové sítě.

- **Příklady: Ovladače síťových karet, ARP protokol, CSMA/CD, PPP.**

Programové vybavení na vyšší úrovni

Programové vybavení na vyšší úrovní je tvořeno řadou protokolů vloženou na programové vybavení koncového zařízení. Slouží k zajištění cílové služby nebo k zpřístupnění síťového prostředku. Obvykle je tato část programového vybavení nezávislá na vybavení nižší úrovně, typu komunikačního zařízení nebo počítačové sítě.

- **Příklady: protokoly TCP/IP, UDP, SNMP, TELNET, RSA, DCE.**

Propojování sítí a funkce propojovacích prvků

Propojujeme sítě s různými topologiemi a operačními systémy. K propojení sítí můžeme použít:

```
- opakovač (repeater, hub)
- most (bridge)
- přepínač (switch)
- směrovač (router)
- přenosovou bránu (gateway)
```

Některé tyto prvky použijeme i pro zlepšení spojení v rámci sítě LAN.

Opakovač (repeater, hub)

- **elektronický pasivní síťový prvek**
- **přijímá zkreslený, zašuměný nebo jinak poškozený signál - vysílá ho dále opravený, zesílený a správně časovaný**
- **vede ke zvýšení dosahu média bez ztráty kvality a obsahu signálu**
- **patří do první (fyzické) vrstvy referenčního modelu OSI (pracují přímo s elektrickým signálem)**
- **nemá žádnou vyrovnávací paměť**
- **může propojovat libovolný počet segmentů sítě, ale pouze segmenty se stejnou přenosovou rychlostí**
- **opakovač šíří kolize**
- **v Ethernetu jich nemůže být libovolně mnoho kvůli době šíření kolize (CSMA/CD) - mezi každými 2 body maximálně 2 opakovače**

Most (bridge)

- **řídí provoz mezi sítěmi přepojováním paketů z jedné sítě do druhé**
- **nešíří tedy každý paket do všech připojených segmentů sítě**
- **zlepšuje spolehlivost, výkon a bezpečnost sítí**

Přepínač (switch)

- **aktivní síťový prvek**
- **propojuje jednotlivé segmenty sítě**
- **obsahuje větší či menší množství portů (až několik stovek), na něž se připojují síťová zařízení nebo části sítě**
- **analyzuje procházející pakety a podle informací v nich obsažených (adres, identifikátorů apod.) rozhoduje, kam paket předat dál**
- **nepropaguje kolize (má paměť)**
- **všesměrově šíří pouze broadcasty**
- **často se používá jako náhrada opakovače**
- **přes jeden přepínač může probíhat více přenosů až do maximální kapacity přepínače**
- **používá režimy provozu Cut-through a Store&Forward**
    - **Cut-through spočívá v tom, že přepojovací uzel (switch) nečeká na načtení celého přenášeného bloku dat, a snaží se jej zpracovat co možná nejrychleji. V konkrétním případě (předpokládejme Ethernetové rámce) si počká jen na načtení hlavičky Ethernetového rámce, ze které již dokáže poznat co má s**

celým blokem dat udělat, a okamžitě také své rozhodnutí naplní. To pak v praxi znamená, že příslušný rámec začne průběžně odesílat v příslušném výstupním směru ještě v době, kdy jej z jiného směru sám teprve přijímá!

- Store&Forward - každý jednotlivý blok dat je v každém přestupním uzlu nejprve celý načten, a teprve pak zpracován. V případě switchů, kterým dost záleží na rychlosti, to ale nemusí být zrovna nejvhodnější. Proto některé switche používají ke svému fungování poněkud jiný princip, kterému se říká cut-through (zatímco jiné zůstávají u původního principu store&forward).

Brána (gateway) - 3 typy

- směrovač
- aplikační brána
- brána pro překlad protokolů z jedné množiny protokolů do jiné
- příklady aplikační brány:
- software pro poštovní aplikace, proxy WWW server apod.

Router (směrovač)

- pracuje na třetí vrstvě OSI modelu (Layer 3) - rozhoduje podle IP adresy
- hraniční router se občas označuje jako gateway (brána)
- slouží pro spojování sítí
- nabízí služby uvnitř LAN (směrování ze zdroje do cíle, segmentování sítě, ARP) a propojení do WAN (přes serial, ISDN, DSL, opticu)
- broadcasty se standardně nepřeposílají - snižuje velikost broadcast domény
- je pomalejší než switch, často je dnes nahrazován Layer 3 switchem
- základní datovou strukturou pro směrování je směrovací tabulka (routing table). Představuje vlastně onu sadu ukazatelů, podle kterých se rozhoduje, co udělat s kterým paketem. Směrovací tabulka je složena ze záznamů obsahujících cílovou adresu a akci, která se má s datagramy provést.
- jak vznikne a jak je udržována směrovací tabulka. Tento proces mají obecně na starosti směrovací algoritmy. Když jsou pak pro určitý algoritmus definována přesná pravidla komunikace a formáty zpráv nesoucích směrovací informace, vznikne směrovací protokol (routing protocol). Směrovací algoritmy můžeme rozdělit do dvou základních skupin: na statické a dynamické. Často se také mluví o statickém a dynamickém směrování, které je důsledkem činnosti příslušných protokolů.
  - Při statickém (též neadaptivním) směrování se směrovací tabulka nijak nemění. Je dána konfigurací počítače a případné změny je třeba v ní provést ručně.
  - Dynamické (adaptivní) směrování průběžně reaguje na změny v síťové topologii a přizpůsobuje jim směrovací tabulky. Do této skupiny patří již zmíněný RIP či OSPF.

Propojení pomocí routeru

Router propojuje stanice, které se nacházejí v jiných subnetech, takže musí dojít k nepřímému doručení. Router již není transparentní síťový prvek, ale ostatní síťová zařízení jej musí adresovat. Obecný způsob komunikace:

1. na síťové vrstvě se vytvoří hlavička IP paketu, která obsahuje IP adresu zdrojové a cílové stanice

2. **zdrojová stanice testuje, zda je cílová IP adresa ve stejném subnetu, tedy zda je pro ni lokální**
3. **dále pokračuje vrstva síťového rozhranní vytvářením hlavičky ethernetového rámce, vloží svoji zdrojovou MAC adresu, cílová MAC adresa se přiřadí k IP adrese z ARP keše (dotazu) a určí se podle:**

1. **při lokální komunikaci adresy cílové stanice**
2. **pokud není lokální, tak se podívá do routovací tabulky a použije adresu patřičného routeru (další hop, často se použije gateway)**
3. **rámec se odešle do sítě**
4. **data přijdou na router, ten podle MAC adresy pozná, že jsou určena jemu**
5. **zkontroluje paket a sníží TTL (doba života v síti - počet hopů)**
6. **znovu se provádí kontrola, zda je IP adresa lokální na jednom z interfaců a buď se odešle dalšímu routeru nebo přímo cílové stanici**
7. **cílová stanice přijme rámec podle MAC adresy**
8. **ověří jej, zkontroluje IP adresu a postoupí jej skrze vrstvy nahoru**

Princip funkce

- **v paměti si sestavuje routovací tabulku podle sítí, kam má přímo připojené interfacy, podle statických hodnot a podle informací od ostatních routerů (záleží na použitém protokolu)**
- **u příchozích paketů se dívá na cílovou IP adresu a podle routovací tabulky určuje cestu k cíli (odesílá data na daný port)**
- **při odesílání dat modifikuje hlavičku rámce, jako zdrojovou MAC adresu vkládá svojí a jako cílovou buď další router nebo stanici**
- **pokud cílová IP adresa patří do některého přímo připojeného subnetu, tak odesílá přímo této stanici, přitom se koukne do ARP tabulky, zda má pro danou IP adresu MAC adresu, pokud ne, tak odešle ARP dotaz (kdo má tuto IP?), pokud nedostane odpověď, tak rámec zahodí, pokud ano, tak doplní ARP tabulku a rámec odešle**

Zdroje

- **Přednášky a semináře Y36PSI**
- **Počítačové sítě http://www.samuraj-cz.com/clanek/pocitacove-site-computer-networks/ [http://www.samuraj-cz.com/clanek/pocitacove-site-computer-networks/]**
- **Online IP Calculator http://www.subnet-calculator.com/subnet.php?net_class=A [http://www.subnet-calculator.com/subnet.php?net_class=A]**
- **Velký průvodce protokoly TCP/IP a systémem DNS**
- **ABC Linuxu http://www.abclinuxu.cz/clanky/site/jemny-uvod-do-adresace-v-protokolu-ip-verze-4 [http://www.abclinuxu.cz/clanky/site/jemny-uvod-do-adresace-v-protokolu-ip-verze-4]**
- **Svět sítí http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&clanekID=215 [http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&clanekID=215]**
- **eArchiv http://www.earchiv.cz [http://www.earchiv.cz]**

spolecne/spol27.txt · Poslední úprava: 2010/06/09 14:22 autor: Strider