Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika

ŠTÁTNY PEDAGOGICKÝ ÚSTAV
NATIONAL INSTITUTE FOR EDUCATION

Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika

Počítačové siete

Predmet: Počítačové systémy

Línia: Vlastný odborový kontext informatiky a informatickej výchovy

Operačný program VZDELÁVANIE

EURÓPSKA ÚNIA
Európsky sociálny fond

Európska únia
Európsky sociálny fond

Moderné vzdelávanie pre vedomostnú spoločnosť/Projekt je spolufinancovaný zo zdrojov ES

Počítačové siete

Identifikácia modulu

Aktivita projektu:	1.2 Vzdelávanie nekvalifikovaných učiteľov informatiky na 2. stupni ZŠ a na SŠ
Línia aktivity:	Vlastný odborový kontext informatiky a informatickej výchovy
Predmet:	Počítačové systémy

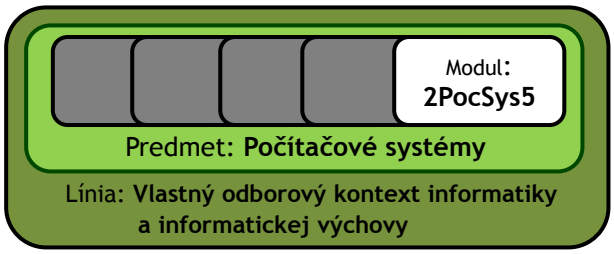
Garant predmetu:

RNDr. Peter Gurský, PhD.
ÚINF PF UPJŠ, Košice
peter.gursky@upjs.sk

Autor:

RNDr. Peter Gurský, PhD.
ÚINF PF UPJŠ, Košice

Zaradenie modulu



Modul tvorí poslednú piatu časť predmetu Počítačové systémy. Účastníci ho absolvujú v štvrtom semestri vzdelávania. Tento modul čiastočne nadväzuje na prvý modul predmetu Počítačové systémy a na predmet „Internet: princípy a tvorba webu“.

Abstrakt modulu

Modul predstavuje princípy počítačových sietí a sieťových protokolov. V úvodnej časti je predstavený vrstvový referenčný model ISO/OSI a vysvetlené úlohy jednotlivých vrstiev. Hlavná časť modulu je rozdelená podľa jednotlivých vrstiev modelu TCP/IP, ktorý tvorí model dnešného internetu. Najvyššia, aplikačná vrstva, bola z časti predstavená v module „Internet: princípy a tvorba webu“. Aplikačná vrstva je preto iba v krátkosti zopakovaná a hlavný dôraz sa kladie na nižšie vrstvy: transportnú, sieťovú, spojovú a fyzickú. V rámci transportnej vrstvy sú účastníkom predstavené protokoly UDP a TCP. Na tejto vrstve sa zaoberáme hlavne významom a použitím čísiel portov na spoluprácu s aplikačnou vrstvou, potvrdzovaním segmentov, nadväzovaním a ukončením TCP spojenia. Na sieťovej vrstve je predstavený protokol IPv4, spôsob adresácie pomocou tried IP adries a delenie sietí pomocou masiek (CIDR). Dôraz sa kladie na pochopenie použitia smerovačov a smerovacích tabuliek. Bližšie sa pozrieme na protokol DHCP, ICMP a fungovanie NAT smerovačov. Na vrstve sieťového rozhrania sa zameriame na pochopenie adresácie cez MAC adresy a ARP protokolu a ARP tabuliek. Zaoberať sa budeme aj rozdielom medzi prepínačom a rozbočovačom. Popíšeme dve najznámejšie prenosové technológie Ethernet a WiFi.

Obsah

Úvod

V tomto module sa účastníci dozvedia o mnohých protokoloch, ktorými sa riadi súčasný internet. Zámerom modulu je pochopenie ich významu a poukázanie na to, kde sa s nimi môžu stretnúť - domáci WiFi smerovač, používanie súkromných IP adries, prideľovanie IP adries DHCP serverom, vzťah otvorených portov a procesov ktoré na nich „počúvajú“, multicastové (napr. televízne) vysielanie, testovanie dostupnosti siete a podobne.

Na mnohé demonštrácie v tomto module postačí príkazový riadok systému Windows alebo BSD/Linux. Okrem toho sa odporúča použiť vopred nainštalovaný program Wireshark. Na jeho fungovanie je potrebné mať práva administrátora. Lektor by mal mať k dispozícii nejaký WiFi smerovač (router).

Vstupné vedomosti

Požadované prerekvizity

Prvý modul predmetu „Počítačové systémy“ a prvý modul predmetu „Internet: princípy a tvorba webu“.

Predpokladané vstupné vedomosti, skúsenosti a zručnosti

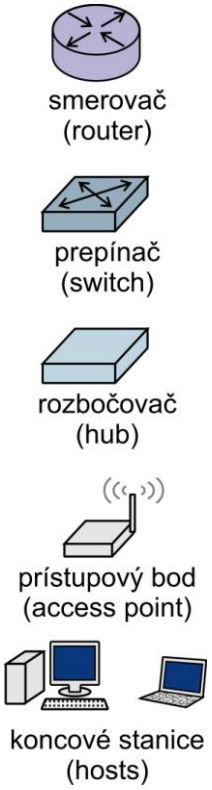
Účastník vzdelávania pozná binárny, dekadický a hexadecimálny zápis čísla, prevod medzi týmito zápismi a vie na čísla aplikovať logické funkcie AND a OR. Pozná použitie aplikačných protokolov HTTP a SMTP. Pozná koncepciu DNS.

Preverenie vstupných vedomostí

Preveďte nejaké číslo z desiatkovej do dvojkovej sústavy.

Vypočítajte logický súčet a súčin dvoch čísiel v binárnom zápise, napríklad:

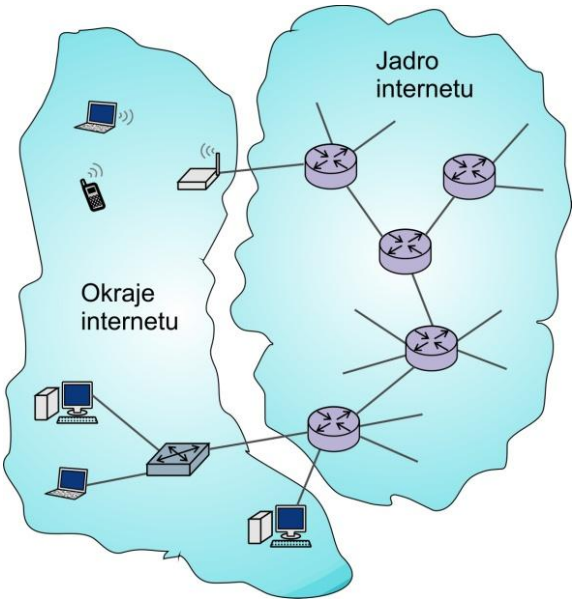
```
    010010011             010010011
AND 110101010         OR 110101010
-------------         ------------
```



Označenia sieťových zariadení

1. Referenčné modely ISO/OSI a TCP/IP

Internet je sieťou počítačových sietí. Je zložený z koncových zariadení a na nich bežiacich sieťových aplikácií a z jadra internetu tvoreného smerovačmi a spojeniami medzi nimi.



Obrázok 1. Jadro a okraje internetu

Internet poskytuje množstvo služieb, ktoré sú riadené **protokolmi**. Protokoly sú určujú spôsob komunikácie zariadení a programov. Majú rôzne formy a štruktúry, ktoré sú podriadené konkrétnemu účelu, napr. "chceme preniesť všetky dáta a v rovnakom poradí", alebo "chceme zistiť, ktorý počítač má danú IP adresu". Protokoly sú to, čo riadi celý internet. Protokoly určujú, kto má čo a kedy právo odoslať a aký tvar to má mať. Presnejšie, protokoly definujú formu, poradie odoslaných a prijatých dát medzi sieťovými prvkami a akcie pri posielaní, prenášaní a prijímaní dát.

Internet je riadený zložitým súborom protokolov. Aby sa zjednodušil návrh komunikačných sietí a vyzdvihli hlavné úlohy protokolov, ale aj úlohy jednotlivých zariadení v sieti, bol navrhnutý už v sedemdesiatych rokoch minulého storočia organizáciou ISO (International Organization for Standardization) referenčný model počítačových sietí nazvaný OSI (Open System Interconnection). Tento model prináša delenie počítačových sietí na vrstvy. Nepopisuje však konkrétne protokoly, ale iba úlohy, ktoré majú jednotlivé vrstvy spĺňať.

Protokoly vyšších vrstiev využívajú služby protokolov nižších vrstiev a dodávajú tak ďalšiu funkcionalitu. Vrstva protokolov môže byť riadená softvérom, hardvérom alebo ich kombináciou. Kým protokoly aplikácií sú určené pre softvér, protokoly fyzickej a spojovej vrstvy sú typicky určené pre hardvér (napr. v sieťových kartách).

1.1 Referenčný model ISO/OSI

Referenčný model ISO/OSI má 7 vrstiev. Každá vrstva má inú úlohu.

Aplikačná vrstva

Aplikačná vrstva zabezpečuje komunikáciu medzi programami konkrétneho typu. Napríklad HTTP protokol slúži na komunikáciu medzi prehliadačom webových stránok a webovým serverom, FTP protokol na posielanie súborov, DNS protokol na preklad doménových mien na IP adresy a naopak, SMTP, POP3 a IMAP protokoly na posielanie mailov a podobne. Do tejto skupiny patrí aj množstvo protokolov na jedno použitie, ktoré si môže navrhnúť pre svoj sieťový program každý programátor.

Základná jednotka informácie, ktorú si medzi sebou posielajú sieťové programy je **správa** (message)

Prezentačná vrstva

Prezentačná vrstva sa podľa referenčného modelu ISO/OSI modelu stará o interpretáciu dát: zabezpečuje šifrovanie, kompresiu a definuje kódovanie znakov.

Relačná vrstva

Relačná vrstva môže určovať kontrolné body relácie, ku ktorým sa dá v prípade potreby vrátiť, ponúka obnovenie relácie (napr. po výpadku siete) a synchronizáciu. Zabezpečuje riadenie komunikačného vzťahu, t.j. relácie.

Transportná vrstva

Hlavnou úlohou transportnej vrstvy je zabezpečiť komunikáciu dvoch procesov (spustených programov) na rôznych počítačoch. Správy, ktoré si posielajú programy vo vyšších vrstvách môžu byť rôzne veľké – od pár bajtov až po veľké, pokojne aj niekoľko gigabajtové súbory. V prípade väčších správ ich transportná vrstva rozdeľuje na menšie časti, z ktorých každá je prenášaná samostatne. Ku každej časti správy sú pridané ďalšie riadiace informácie, čím vznikne takzvaný **segment**. Tieto riadiace informácie slúžia na cieľovej stanici na určenie toho, pre ktorý proces je táto časť správy určená.

Na internete sa najčastejšie používajú z transportných protokolov dva: **TCP** a **UDP**. Zatiaľ čo UDP protokol umožňuje iba jednoduché odosielanie segmentov, TCP protokol spojovaný a potvrdzovaný. Zabezpečuje tak oproti protokolu UDP doručenie všetkých dát v správnom poradí (ak sa nejaký segment na ceste stratí alebo znehodnotí, zabezpečí jeho opätovné zaslanie), rieši kontrolu zahltenia (spomalí odosielanie, ak niektorý smerovač na ceste je zahltený) a kontrolu toku dát (odosiela iba takou rýchlosťou, akou je ich schopné spracovať prijímajúce zariadenie).

Sieťová vrstva

Protokoly transportnej vrstvy predpokladajú, že segment sa na cieľový počítač „nejako dostane“. Prenesenie segmentu, vytvoreného transportnou vrstvou, na cieľový počítač je úlohou sieťovej vrstvy. Sieťová vrstva pridá k segmentu, okrem iného, informáciu na určenie adresy cieľového počítača, čím vznikne takzvaný **datagram**, alebo **paket**. Hlavnou úlohou tejto vrstvy je nájdenie vhodnej cesty pre datagram smerujúci k cieľovému zariadeniu. Základným protokolom sieťovej vrstvy na internete je protokol **IP** verzie 4, alebo najnovšie aj protokol IP verzie 6.

Spojová vrstva

Úlohou spojovej vrstvy je preniesť datagram medzi ľubovoľnými dvoma susednými zariadeniami. Spojová vrstva poskytuje prostriedky pre riadenie prístupu k médiu, adresovanie uzlov na spoločnom médiu a môže poskytovať aj kontrolu správnosti prenosu, spoľahlivosť, spojovanosť a riadenie toku dát. To, ktoré prostriedky sú použité je dané technológiou, ktorá je typicky prispôsobená prenosovému médiu. Iná technológia môže byť vhodná pri komunikácii po kovovom drôte, iná v optickom vlákne a iná pri bezdrôtovom prenose. Datagram doplnený riadiacimi informáciami tejto vrstvy sa nazýva **rámec**.

Fyzická vrstva

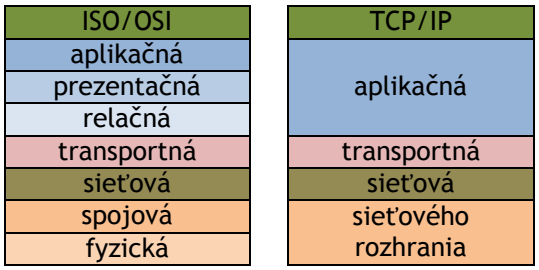
Táto vrstva má za úlohu v danom prenosovom médiu prenášať binárnu informáciu, teda postupnosť jednotiek a núl. Fyzická vrstva prenášaným informáciám nerozumie. Prenosovými médiami sú hlavne kovové a optické drôty a bezdrôtové rádiové spojenie. Fyzická vrstva okrem prenosových médií popisuje aj parametre koncových zariadení a kódovanie a charakter signálu.

1.2. Referenčný model TCP/IP

Referenčný model ISO/OSI je abstraktným popisom. Na internete sa v súčasnosti využíva rodina protokolov TCP/IP, ktorá má svoj vlastný referenčný model. V tomto modeli sú prezentačná a relačná vrstva súčasťou aplikačnej vrstvy. V praxi sa totiž ukázalo, že mnohé aplikácie služby prezentačnej a relačnej vrstvy nevyužívajú.

Referenčný model TCP/IP tiež spája úlohy spojovej a fyzickej vrstvy v spoločnej vrstve sieťového rozhrania. Typickým zástupcom tejto vrstvy je rodina technológií pre lokálne (LAN) siete nazývaná Ethernet.

Okrem rodiny protokolov TCP/IP existuje ešte celý rad ďalších modelov počítačových sietí. V tomto učebnom texte sa budeme venovať iba popisu funkcií referenčného modelu TCP/IP, keďže ide o model, ktorý je v dnešnom internete dominantný.



Obrázok 2. Vrstvy referenčných modelov ISO/OSI a TCP/IP

2. Aplikačná vrstva modelu TCP/IP

Je bežné, že dve rôzne sieťové aplikácie od rôznych výrobcov, a pokojne aj na rôznych operačných systémoch, spolu na internete normálne komunikujú. Komunikácia medzi ľubovoľnými dvoma spustenými programami je tvorená správami, ktoré si navzájom vymieňajú. To, že si tieto programy rozumejú, umožňujú aplikačné protokoly, ktoré určujú, aké správy si majú tieto aplikácie posielať. Stačí, že týmto správam rozumejú iba tieto dva programy. Ostatné sieťové zariadenia na ceste nijako tieto správy nespracúvajú ani neanalyzujú. Vývoj nových aplikačných protokolov je teda úplne nezávislý od fungovania siete a prenosu správ v nej.

Každý aplikačný sieťový protokol definuje:

- typy správ, ktoré si programy posielajú (požiadavky, odpovede, informácie, rôzne typy dát)
- tvar správ určujúci, z čoho budú správy zložené a aké hodnoty môžu jednotlivé časti správy obsahovať
- význam správ a informácií v nich
- okolnosti, za ktorých sa jednotlivé správy posielajú

RFC špecifikácie nájdete na stránke: http://www.ietf.org/rfc.html

Definícia verejných aplikačných protokolov je daná v ich voľne prístupných RFC špecifikáciách. Súkromné aplikačné protokoly umožňujú vývoj nových sieťových programov, kde si špecifikáciu musí vytvoriť jeden programátor (alebo skupina programátorov, ktorá sa na danom protokole dohodla).

Pred samotným písaním nového protokolu je potrebné si uvedomiť, akú rolu budú hrať jednotlivé časti sieťovej aplikácie. Sieťové aplikácie sa z tohto pohľadu dajú rozdeliť na tri skupiny:

- **Klient/server architektúra** predpokladá , že na internete je jeden, alebo viac, stále zapnutých počítačov s pevnou IP adresou, nazývaných servery. Na týchto počítačoch beží serverová časť aplikácie. Klienti spúšťajú a vypínajú klientskú časť aplikácie podľa potreby. Môžu mať dynamickú aj privátnu IP adresu. Klienti komunikujú vždy iba so serverom, nikdy nie s ostatnými klientmi. Server musí byť často výkonný počítač alebo dokonca klaster

niekoľkých serverov, aby zvládol nápor množstva klientov. Klient/server architektúru realizujú napríklad protokoly HTTP, FTP či DNS.

- V **peer-to-peer architektúre** žiaden z počítačov nemusí byť stále zapnutý. Programy v tejto architektúre sú niekedy klientmi, niekedy servermi a často oboma naraz. Môžu sa odpájať a pripájať podľa ľubovôle. Záťaž je rozložená oveľa rovnomernejšie ako v prípade klient/server architektúry. Manažovanie siete je dosť ťažké. Príkladom takejto architektúry sú Gnutella alebo Freenet.
- Posledný typ je **hybrid klient/server a peer-to-peer architektúry**. Stále zapnutý server sa používa väčšinou na registráciu a vyhľadávanie napojených účastníkov, prípadne indexovanie ich dát. Samotné nosné dátové prenosy sa dejú už medzi jednotlivými účastníkmi. Príkladmi tejto architektúry sú protokoly XMPP (známy aj ako Jabber), Skype alebo BitTorrent.

Projekt Freenet (http://freenetproject.org) je peer-to-peer sieť v ktorej každý účastník aj zdieľaný obsah majú zabezpečenú úplnú anonymitu

2.1. Komunikácia aplikačnej vrstvy s nižšími vrstvami

Každý sieťový program sa spustí ako proces prípadne množina procesov (záleží na programátorovi). Pokiaľ chcú dva procesy komunikovať v rámci jedného počítača môžu využiť medziprocesovú komunikáciu, ktorú poskytuje operačný systém. Procesy na rôznych počítačoch komunikujú prostredníctvom správ aplikačným protokolom.

Bez ohľadu na typ architektúry aplikácie, je pri sieťovej komunikácii medzi dvoma procesmi vždy jeden z procesov klientský a jeden serverový. Klientský proces je ten, ktorý inicializuje komunikáciu a serverový proces je ten, ktorý čaká, že sa naňho niekto napojí.

Na to, aby dva procesy na rôznych staniciach mohli komunikovať, musí klientský proces poznať identifikátor, presne určujúci, kde sa nachádza serverový proces, aby s ním mohol inicializovať komunikáciu. Identifikátor zahŕňa **IP adresu** stanice, na ktorej beží serverový proces a **číslo portu**, ktorý jednoznačne identifikuje proces v rámci tejto stanice. Netreba zabúdať, že na jednom počítači môže byť spustených niekoľko serverových procesov a musia sa preto ich identifikátory od seba líšiť.

Jednotlivé webové aplikácie majú rôzne nároky na transportnú vrstvu a sieťové pripojenie. Niektoré nie sú citlivé na stratu časti dát (napr. VoIP, streamované rádia a televízie), iné vyžadujú stopercentne spoľahlivé dátové prenosy (napr. kopírovanie súborov). Na druhej strane, rôzne multimediálne prenosy obrazu a zvuku vyžadujú nejakú minimálnu prenosovú rýchlosť, aby ich kvalita bola dostatočná. Iné aplikácie nie sú až také citlivé na prenosovú rýchlosť a dokážu sa prispôsobiť aktuálnym možnostiam. Aplikácie pracujúce v reálnom čase vyžadujú zasa malé zdržanie správ. Charakter internetu však neumožňuje žiadnemu transportnému protokolu zabezpečiť minimálnu šírku pásma a ani nezaručí malé zdržanie správ. Jediné, čo vieme urobiť, je, zabezpečiť si takého poskytovateľa internetového pripojenia, ktorý nám poskytne „dostatočne rýchle a priepustné pripojenie“.

Všetky aplikačné protokoly, ktoré očakávajú spoľahlivý prenos všetkých správ v správnom poradí využívajú transportný protokol TCP. Protokol UDP je odľahčený protokol nepodporujúci spoľahlivý prenos dát (navyše ani v správnom poradí). Ak sa odosielajúca stanica nemusí starať o to, či všetky dáta spoľahlivo došli ku klientom, je vhodnejšie použiť UDP protokol, ktorý odľahčí odosielajúcu stanicu aj sieťovú komunikáciu. Typicky sú to práve spomínané streamované rádia a televízia či telefonovanie cez internet.

2.2 Aplikačné protokoly

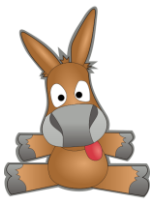
V module „Internet: princípy a tvorba webu“ ste sa už zoznámili s viacerými hlavnými aplikačnými protokolmi. Poznáte už aplikačný protokol HTTP na komunikáciu webového servera a webového prehliadača. Okrem toho poznáte hlavné črty protokolu SMTP, ktorý definuje komunikáciu pri posielaní e-mailov a formát e-mailových správ.

Aktivita	Opakovanie: Zobrazte si požiadavku a odpoveď v HTTP protokole cez vhodné rozšírenie webového prehliadača. Napríklad rozšírenie Firebug alebo Live HTTP Headers v prehliadači Firefox.
Aktivita	Otvorte si vášho e-mailového klienta a zobrazte si nejaký e-mail s prílohami. Identifikujte hlavičku a telo e-mailu. V hlavičke nájdite e-mail odosielateľa a príjemcu. Zhoduje sa e-mail odosielateľa s e-mailom v riadku začínajúcom s „Reply-to:“?

Odosielateľ e-mailu je v riadku začínajúcom s „From:“. Pokiaľ je uvedená iná e-mailová adresa v riadku začínajúcom s „Reply-to:“, pri odpovedi na e-mail sa použije práve táto adresa.

Ďalším aplikačným protokolom (popísaným v RFC 1035) sa realizuje komunikácia s DNS servermi na získanie prekladu doménových mien (*host names*) na IP adresy a naopak. Správy v tomto protokole sú zložené z tzv. DNS záznamov, z ktorých každý obsahuje jeden z možných prekladov, keďže vo všeobecnosti každá IP adresa môže mať pridelených niekoľko doménových mien, ale aj každé doménové meno môže byť pridelené viacerým IP adresám. Prvý prípad je typický pre hostingové servery a druhý napríklad pre veľmi vyťažené webové stránky, ktoré sú poskytované z viacerých počítačov, aby sa rozložila záťaž. Princíp fungovania služby DNS už takisto poznáte z modulu „Internet: princípy a tvorba webu“.

Aktivita	Pomocou programu **nslookup** v príkazovom riadku zistite, na koľkých IP adresách počúva server www.google.com.



Logo projektu eMule

Okrem spomínaných aplikačných protokolov sú veľmi populárne rôzne protokoly v peer-to-peer sieťach ako napríklad **BitTorrent** alebo **eMule**.

Všetky spomínané aplikačné protokoly majú zverejnené svoje RFC špecifikácie a každý človek si tak môže vytvoriť vlastný program, komunikujúci týmto protokolom. No nie všetky protokoly sú verejné. Typickým zástupcom uzavretých aplikačných protokolov je **Skype**. Pre uzavreté protokoly platí, že neexistujú verejne prístupné RFC, ktoré by ich popisovali a nevieme teda vytvoriť vlastnú implementáciu sieťovej aplikácie, ktorá by týmto protokolom komunikovala.

3. Transportná vrstva modelu TCP/IP

Protokoly transportnej vrstvy zabezpečujú komunikáciu medzi dvoma procesmi bežiacimi vo všeobecnosti na rôznych počítačoch. Transportná vrstva už nerozlišuje typ aplikácie (webové stránky, FTP prenos, DNS požiadavka a podobne). Transportná vrstva na strane odosielateľa dostane od niektorej aplikácie nejakú správu pre nejakú cieľovú aplikáciu. Hlavnou úlohou transportnej vrstvy je umožniť prenos správ, alebo toku dát, medzi dvoma procesmi, t.j. spustenými sieťovými aplikáciami, alebo ich vláknami, na rôznych koncových zariadeniach. Transportná vrstva pritom predpokladá, že nižšie vrstvy referenčného modelu TCP/IP prenesú segmenty z počítača na počítač. Jej úlohou teda ostáva na cieľovom počítači určiť cieľový proces, ktorému je správa určená.

Reálne je určenie cieľového procesu zložitejšie, keďže za istých okolností môže byť viac procesov alebo vlákien pridelených k jednému číslu portu. Presné určenie cieľového procesu alebo vlákna tak určuje okrem cieľového portu aj cieľová IP adresa v prípade protokolu UDP, alebo dokonca až štvorčíslie (zdrojový a cieľový port a zdrojová a cieľová IP adresa) v prípade protokolu TCP.

Transportná vrstva najprv rozdelí správy z odosielajúcej aplikácie tak, aby žiaden kúsok nepresiahol maximálnu dovolenú veľkosť. Ku každej časti správy sa dodá **hlavička** vybraného transportného protokolu a vznikne **segment**. Ten sa potom posiela na spracovanie nižším vrstvám modelu TCP/IP.

V hlavičke každého segmentu sa nachádzajú dve dôležité čísla: **zdrojový port** a **cieľový port**. Každý proces alebo vlákno, ktoré chce komunikovať po sieti má pridelené svoje číslo portu na danej stanici. Prijímajúca stanica teda vie ktorému procesu zaslať obsah správy z doručeného segmentu na základe čísla cieľového portu v hlavičke segmentu. Na základe zdrojového portu z hlavičky segmentu zasa vie, na aký port má zaslať prípadnú odpoveď.

Číslo portu si môže serverový proces (lepšie povedané jeho programátor) vybrať aký chce z rozsahu 0 až 65535. Dobre známe služby už majú všeobecne známe čísla portov (well known ports). Napríklad HTTP server obvykle počúva na porte 80, FTP server na porte 21 a podobne.

Známe čísla portov si môžete pozrieť napríklad na webovej stránke http://www.iana.org/assignments/port-numbers

Na internete sa používajú hlavne transportné protokoly TCP a UDP.

3.1. UDP protokol transportnej vrstvy

Hlavička protokolu UDP (user datagram protocol) je maličká. K číslam portov pridáva už iba informáciu o počte bajtov segmentu a kontrolný súčet.

zdrojový port	cieľový port
dĺžka	kontrolný súčet
správa aplikačnej vrstvy (alebo jej časť)	

Obrázok 3. UDP segment

Kontrolný súčet sa používa na odhalenie chýb v dátach spôsobených prenosom. Chyba môže nastať ľubovoľným negatívnym vplyvom pri prenose (silné magnetické pole, nečistota v optickom vlákne a pod.). Chyba sa prejaví zmenou niektorých bitov segmentu na opačné.

Ak chybu odhalíme, segment sa zahodí a aplikácia časť správy z tohto segmentu nedostane. Okrem dôjdeného chybného segmentu sa môže stať aj to, že niektoré segmenty na cieľový počítač vôbec neprídu - stratia sa na ceste k cieľovému počítaču. Ak si nemôžeme stratu časti správy dovoliť, napríklad pri kopírovaní súboru, je potrebné použiť protokol TCP, ktorý dokáže zabezpečiť spoľahlivý prenos všetkých častí správy.

Na vytvorenie kontrolného súčtu vezmeme obsah správy plus čísla portov, pozrieme sa na nich ako na postupnosť bitov a rozdelíme ich na 16 bitové kúsky. Tým nám vznikne mnoho 16 bitových čísiel, ktoré všetky po jednom sčítame, pričom ignorujeme prenesené bity mimo 16 bitový rozsah.

Napríklad ak máme urobiť kontrolný súčet z dát 01100110 01100110 01010101 01010101 10001111 00001111, potrebujeme sčítať tri 16 bitové čísla. Najprv sčítame prvé dve a k výsledku pripočítame tretie.

```
0110011001100110
0101010101010101
================

1011101110111011

1011101110111011
1000111100001111
================
0100101011001010
```

Nakoniec ešte vytvoríme z výsledku inverzné číslo, t.j. zmeníme vo výsledku všetky 1 na 0 a všetky 0 na 1. Takto vzniknuté číslo prehlásime za kontrolný súčet a vložíme ho do hlavičky UDP segmentu. Príjemca segmentu urobí opäť súčet 16 bitových čísiel doručeného segmentu rovnakým spôsobom, ako to robil odosielateľ a nakoniec k súčtu pripočíta dôjdený kontrolný súčet. Ak sa vo výsledku nachádza aspoň jedna nula, segment obsahuje chybu. Inak predpokladáme, že segment chybu neobsahuje (ale nevieme to na 100%).

```
vypočítaný súčet:      0100101011001010
dôjdený kontrolný súčet: 1011010100110101
                       ================
                       1111111111111111
```

3.2. Transportný protokol TCP

Nižšie vrstvy referenčného modelu TCP/IP zabezpečujú nasmerovanie datagramov

k cieľovému zariadeniu. Niektoré datagramy sa však môžu z rôznych dôvodov na ceste k príjemcovi stratiť. Tento nepríjemný fakt umožňuje eliminovať transportný protokol TCP (Transport control protocol). Na rozdiel od protokolu UDP je TCP protokol spojovaný a potvrdzovaný. Komunikácia dvoch procesov je podmienená vytvorením spojenia. V rámci tohto spojenia musí mať odosielateľ segmentov potvrdené, že všetky segmenty boli úspešne prenesené k príjemcovi. Ak nebol nejaký segment prenesený k príjemcovi, je vynútené jeho opätovné zaslanie. Dôsledkom je spoľahlivý prenos správ medzi procesmi.

Hlavička TCP segmentu už používa oveľa viac hodnôt ako hlavička protokolu UDP.

<table>
<tr><td colspan="8">zdrojový port (16 b)</td><td colspan="2">cieľový port (16 b)</td></tr>
<tr><td colspan="10">sekvenčné číslo (32 b)</td></tr>
<tr><td colspan="10">číslo potvrdenia (32 b)</td></tr>
<tr><td>dĺžka hlavičky (4 b)</td><td colspan="2">(nič) (6 b)</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>veľkosť okna príjemcu (16 b)</td></tr>
<tr><td colspan="8">kontrolný súčet (16 b)</td><td colspan="2">umiestnenie urgentnej časti (16 b)</td></tr>
<tr><td colspan="10">voliteľné nastavenia</td></tr>
<tr><td colspan="10">správa aplikačnej vrstvy (alebo jej časť)</td></tr>
</table>

Obrázok 4. TCP segment

Zdrojový a **cieľový port**, ako aj **kontrolný súčet** už poznáme z UDP protokolu.

Od chvíle, keď sa vytvorí spojenie, posielajú si aplikácie navzájom mnoho správ. Na správy odoslané z jednej z komunikujúcich aplikácií sa môžeme pozerať ako na tok dát. Teraz si predstavte, že v tomto toku dát si očíslujeme každý bajt. **Sekvenčné číslo** predstavuje poradové číslo prvého bajtu tej časti toku dát, ktorá je odosielaná v tomto segmente. Číslovanie toku dát sa začína náhodným číslom a po dosiahnutí maximálnej hodnoty 2^{32}-1 sa pokračuje opäť od nuly.

Protokol TCP predpokladá obojstrannú komunikáciu, t.j. proces na jednej strane komunikácie posiela nejaké správy procesu na druhej strane a naopak. **Číslo potvrdenia** sa vzťahuje na číslovanie toku dôjdených (t.j. nie odoslaných) dát. Odosielateľ tohto segmentu týmto číslom hovorí druhej strane komunikácie, že aké sekvenčné číslo jej toku dát očakáva ako nasledujúce. Ako uvidíme neskôr, tento mechanizmus zabezpečí, aby sa v prípade straty alebo poškodenia niektorých segmentov zabezpečilo ich opätovné zaslanie.

Možný obsah voliteľných nastavení v TCP segmente si môžete pozrieť v RFC 1323, alebo stručnejšie napríklad na stránke http://www.securityfocus.com/infocus/1223

Dĺžka hlavičky je v hlavičke kvôli časti **voliteľné nastavenia**, ktoré môžu mať rôznu veľkosť.

Príznaky URG, ACK, PSH, RST, SYN a FIN majú každý veľkosť iba jeden bit. Ak majú hodnotu 1 tak sú aktivované, ak 0 tak sú vypnuté.

- URG - urgentné. Niektoré aplikačné protokoly môžu využívať tento príznak na označenie toho, že sa prerušuje zasielaný tok dát, aby sa oznámila nejaká správa. Napríklad, počas posielania veľkého súboru chceme robiť nejaké riadiace príkazy (prerušiť posielanie, informovať o nejakej udalosti na stanici odosielateľa). Hodnota **umiestnenie urgentnej časti** je v tomto prípade nastavená na pozíciu posledného bajtu urgentnej správy v segmente.
- ACK – „acknowledgement“. Hovorí, že hodnota sekvenčné číslo je vyplnená relevantnou hodnotou.
- PSH - "push" t.j. posuň. Hovorí, že dáta majú byť okamžite posunuté aplikačnej vrstve.
- RST - "reset". Ide o oznam násilného ukončenia spojenia (napr. niekto ukončil komunikujúci proces).
- SYN - "synchronize". Označenie segmentov použitých pri začiatku spojenia.
- FIN - "finalize". Označenie segmentov použitých pri uzatváraní spojenia.

Veľkosť okna príjemcu informuje o veľkosti voľnej časti pamäte príjemcu vyhradenej pre prijaté segmenty, t.j. veľkosti okna prijatých segmentov (keďže oba procesy môžu byť príjemcovia aj odosielatelia, každý má svoje okno príjemcu). Okrem okna príjemcu má každý proces vytvorené aj okno odosielateľa, ale o jeho veľkosti neinformuje. Konkrétny význam okien príjemcu a odosielateľa si vysvetlíme nižšie.

3.2.1. Manažment nadviazania spojenia

Po nadviazaní spojenia už budú klientský aj serverový proces z hľadiska TCP protokolu rovnocenné – oba budú odosielať a prijímať správy od toho druhého.

Pred tým, ako si začnú oba komunikujúce procesy navzájom posielať správy, je potrebné, aby sa medzi nimi nadviazalo spojenie. Pri nadväzovaní spojenia si obidva konce musia pripraviť niekoľko premenných, na základe ktorých sa počas prenosu bude riadiť prijímanie a odosielanie dát, potvrdzovanie prijatých segmentov a opätovné odoslanie stratených segmentov.

Nadviazanie TCP spojenia je realizované trojfázovým "potrasením rúk" (three-way handshake).

1. **Odoslanie SYN segmentu k serveru.** V prvej fáze si klientský proces (t.j. ten iniciuje spojenie) vygeneruje náhodné úvodné sekvenčné číslo, od ktorého sa začnú číslovať bajty odosielaných dát z aplikačnej vrstvy. Okrem tohto čísla si vytvorí aj okno odosielaných a prijímaných segmentov nejakej úvodnej veľkosti. Úvodné sekvenčné číslo sa vloží do hlavičky. Samozrejme sa do hlavičky napíše číslo zdrojového a cieľového portu. Okrem toho je v hlavičke ešte nastavený príznak SYN na 1 označujúci, že ide o prvý segment „potrasenia rúk“.
2. **Odoslanie SYN/ACK segmentu ku klientovi** (sú zapnuté príznaky SYN aj ACK). Serverová aplikácia akceptovala spojenie a vytvorila nový proces alebo vlákno, ktoré bude komunikovať iba s týmto klientským procesom na druhej strane tejto komunikácie. K tomuto novému procesu sa tiež vygenerujú úvodné premenné, t.j náhodné úvodné sekvenčné číslo pre dáta odosielané na klienta, a ešte okná prijatých a odoslaných segmentov. V hlavičke sa nastaví cieľový port podľa zdrojového portu z prijatého segmentu a zdrojový port podľa cieľového portu prijatého segmentu. Ako sekvenčné číslo pošle vygenerované úvodné číslo a ako číslo potvrdenia pošle číslo o 1 väčšie ako sekvenčné číslo z prijatého segmentu, čo znamená že v najbližšom segmente očakáva, že ako sekvenčné číslo bude uvedené práve toto číslo.
3. **Odoslanie ACK segmentu k serveru**. V poslednej, tretej fáze klient pošle serveru segment so zapnutým ACK príznakom. Sekvenčné číslo je o 1 väčšie ako v prvom segmente a číslo potvrdenia je o 1 väčšie ako v sekvenčné číslo v segmente, ktorý prišiel zo servera.

Musíme počítať s tým, že prípadné zaslanie segmentu oznamujúceho koniec spojenia sa môže stratiť na ceste k cieľovej stanici.

Po inicializácii môže jedna aj druhá strana už posielať dáta. Klientský proces ich môže začať posielať už ako súčasť tretieho segmentu „potrasenia rúk“. Serverový proces tak môže urobiť až po prijatí tohto tretieho segmentu komunikácie. Kto začne posielať dáta závisí od aplikačného protokolu.

Príklad nadviazania spojenia (uvádzame iba podstatné položky):

1. Klient posiela: zdrojový port: 12345, cieľový port: 80, sekvenčné číslo 300000 (náhodne vygenerované), číslo potvrdenia: 0, SYN=1, ACK=0
2. Server posiela: zdrojový port: 80, cieľový port: 12345, sekvenčné číslo 200000 (náhodne vygenerované), číslo potvrdenia: 300001, SYN=1, ACK=1
3. Klient posiela: zdrojový port: 12345, cieľový port: 80, sekvenčné číslo 300001, číslo potvrdenia: 200001, SYN=0, ACK=1

3.2.2. Manažment ukončenia spojenia

Korektné ukončenie TCP spojenia je potrebné urobiť tak, aby obe komunikujúce strany vedeli, že druhá strana vie, že spojenie sa končí. Požiadavka na ukončenie spojenia môže prísť od jednej alebo druhej strany - často záleží aj na aplikačnom protokole.

1. Prvá stanica (iniciátor ukončenia) posiela FIN segment. Tento segment má nastavený príznak FIN na 1. Príznak ACK môže byť 0 alebo 1 podľa toho, či bol bezprostredne pred tým zaslaný nejaký dátový segment z druhej stanice.
2. Druhá stanica posiela FIN/ACK segment, t.j. príznaky FIN aj ACK sú nastavené na 1. Odosielanie nasleduje po prijatí FIN segmentu.
3. Prvá stanica odosiela ACK segment. FIN je nastavené na 0 a ACK na 1. V tejto chvíli prvá stanica zapne časovač na "dosť dlhý čas" (často 30 s, minúta, alebo dve minúty). Ak počas tohto času dostane FIN/ACK segment, znamená to, že sa jeho ACK segment stratil a posiela ho znova (viď Prenos dát nižšie). Ak nedostane už žiadne segmenty po zastavení časovača sa spojenie zavrie.
4. Druhá stanica dostane ACK segment. Spojenie sa zavrie.

Aktivita

Spustite v programe Wireshark zachytávanie sieťovej komunikácie. Navštívte niekoľko webových stránok a analyzujte niektoré TCP a UDP segmenty. Porovnajte ich hlavičky. Nájdite v komunikácii začiatok a koniec niektorého TCP spojenia.

3.2.3. Potvrdzovanie doručenia segmentov - metóda posuvného okna (sliding window)

Ako sme si už povedali, nižšie vrstvy modelu TCP/IP nezaručujú doručenie všetkých segmentov k príjemcovi. Datagramy, ktoré tieto segmenty prenášajú, sa môžu stratiť, poškodiť alebo prísť do cieľovej stanice v inom poradí, než v akom boli vyslané.

Komunikácia je obojstranná. Odosielateľom môže byť klient aj server.

TCP protokol je potvrdzovaný. To znamená, že odosielateľ dostáva potvrdenia o tom, ktoré segmenty boli úspešne doručené k príjemcovi. Segmenty, ktoré z nejakých dôvodov neboli úspešne doručené sú opätovne zasielané. Cieľom je aby boli cieľovému procesu doručené všetky správy a v tom istom poradí, v akom boli vyslané odosielajúcim procesom.

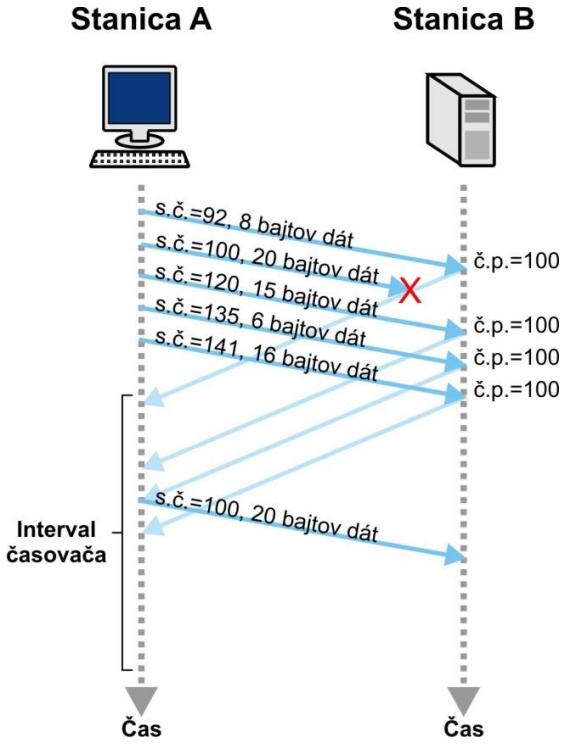
Tok dát, ktorý odosiela odosielateľ príjemcovi je rozdeľovaný do segmentov. Každý segment má pridelené **sekvenčné číslo**, ktoré je číslom prvého bajtu ním prenášanej časti tohto toku dát.

Keďže odosielateľ dát musí byť nejakým spôsobom informovaný príjemcom o tom, či všetky odoslané segmenty boli doručené príjemcovi, príjemca to odosielateľovi „oznamuje“ pomocou **čísiel potvrdenia** v TCP hlavičke. TCP protokol používa takzvané **kumulatívne potvrdenie**: ak odosielateľ dostane segment s číslom potvrdenia X, tak to znamená, že príjemca prijal v poriadku všetky segmenty s číslami menšími ako X a očakáva, že mu príde segment so sekvenčným číslom X, ktorý zatiaľ nedostal.

Odosielateľ má vyhradené miesto v pamäti nazývané **okno odosielateľa**. V ňom si uchováva všetky odoslané datagramy, ktoré zatiaľ nemá potvrdené. Veľkosť tohto okna určuje, koľko segmentov za sebou môže odoslať bez toho, aby niektorý z nich bol potvrdený. Ak mu príde od príjemcu potvrdenie s nejakým číslom potvrdenia X, môže z okna odosielateľa vymazať všetky segmenty s menším číslom potvrdenia a uvoľní tak miesto v okne pre ďalšie zatiaľ neodoslané segmenty.

Príjemca má vyhradené miesto v pamäti, ktoré je nazývané okno príjemcu. Do tohto okna si ukladá všetky segmenty, ktoré prijal mimo poradia. Vo chvíli, keď prijme segmenty, ktoré sú v prúde dát pred nimi a vyplnia chýbajúce časti toku doručených dát, sú tieto dáta odovzdané cieľovému procesu až po prvý chýbajúci segment. Podľa pravidiel kumulatívneho potvrdenia posiela príjemca odosielateľovi potvrdzovací segment s číslom potvrdenia prvého doteraz nedoručeného segmentu v toku prijatých dát. Na obrázku 5 príjemca (Stanica B) prijme postupne segmenty s číslami 92, 120, 135 a 141. Keď prijme segment s číslom 92 a okno prijímateľa má prázdne, odovzdá dáta z tohto segmentu aplikačnej vrstve a očakáva segment so

sekvenčným číslom 100. Pošle teda segment s číslom potvrdenia 100. Následne dostane segmenty so sekvenčnými číslami 120, 135 a 141. Keďže očakáva segment so sekvenčným číslom 100, uloží si tieto segmenty do okna príjemcu a pre každý z nich pošle segment s číslom potvrdenia 100.



Obrázok 5. Opätovné zaslanie datagramu pred vypršaním časového limitu

Odosielateľ musí nejako zareagovať na situáciu, že sa segment úspešne nepreniesol. Musí teda nejako identifikovať samotnú udalosť straty segmentu, ktorá sa stala niekde na ceste medzi cieľovými stanicami. Odosielateľ považuje za stratu takú situáciu, v ktorej nastala jedna z dvoch udalostí:

- Prišli aspoň 3 potvrdzovacie segmenty s rovnakým číslom potvrdenia. Pri viacnásobnom prijatí rovnakého potvrdzovacieho čísla totiž vie, že tieto potvrdenia boli odoslané príjemcom po prijatí segmentov so sekvenčnými číslami väčšími ako toto opakované číslo potvrdenia. Segment s týmto číslom je zaslaný znova, lebo sa pravdepodobne stratil (mohol aj iba meškať, lebo išiel pomalšou trasou po internete).
- Vypršal časový limit. Odosielateľ zapína časovač vždy vtedy, keď sa zmení najstarší segment v okne odosielaných segmentov. Toto môže nastať, ak vložíme segment do prázdneho okna odosielateľa, alebo ak došlo potvrdenie, ktoré potvrdilo doručenie dovtedy najstaršieho segmentu (niektoré segmenty sa odstránili) a iný segment v okne sa stal najstarším. Časovač sa vždy zapne s nejakým časovým intervalom. Ak počas tohto časového intervalu nedôjde potvrdenie, ktoré by potvrdilo doručenie najstaršieho segmentu, vyprší časový limit a tento segment sa pošle znova.

V skutočnosti sa robí ešte to, že ak príjemca dostane segment mimo poradia, pošle ihneď dva rovnaké potvrdzovacie segmenty, aby sa ešte viac urýchlilo opätovné zaslanie strateného segmentu.

Na obrázku 5 vidíme situáciu, že odosielateľ dát (Stanica A) dostane viackrát segment s rovnakým číslom potvrdenia. Po troch segmentoch s rovnakým číslom potvrdenia 100 pošle tento segmentu so sekvenčným číslom 100 ešte raz ešte pred vypršaním časového limitu. Segmenty so sekvenčnými číslami 120,135 a 141 neposiela, lebo predpokladá, že ich Stanica B má uložené v okne príjemcu, čo je v tomto prípade pravda.

Nastavenie intervalu časovača sa prispôsobuje na základe predchádzajúcich skúseností s časom, za aký sa potvrdili predchádzajúce segmenty. Je snaha, aby tento interval nebol zbytočne veľký, aby nedochádzalo k zbytočným opätovným

poslaniam segmentov, pokiaľ nedošlo k strate, len segmenty mali väčšie zdržanie. Tiež by nemal byť príliš veľký, aby sme zabránili pomalým reakciám na stratu segmentu.

Na veľkosť okna odosielateľa má vplyv hlavne frekvencia výskytu stratených segmentov. Ak máme veľké okno odosielateľa, tak za jednotku času sa odosiela veľa dát. Keď sa začnú strácať segmenty, znamená to, že niektorá časť prenosovej cesty nestíha prenášať dáta s takou prenosovou rýchlosťou. Zmenou veľkosti okna odosielateľa tak odosielateľ prispôsobuje prenosovú rýchlosť aktuálnemu zaťaženiu siete.

Aktivita	Spustite si demo na demonštráciu algoritmu posuvného okna na stránke http://www3.rad.com/networks/2004/sliding_window/. Nastavte si nenulové percento stratených segmentov (loss).
Aktivita	Spustite v príkazovom riadku príkaz **„netstat -aon“** a zistite koľko serverov vám na vašom počítači počúva a na akých portoch. Koľko máte vytvorených aktívnych spojení? Využite správcu úloh systému Windows (cez klávesovú skratku CRTL-ALT-DELETE) a zistite podľa stĺpca pid, aké programy aktuálne používajú protokol TCP a aké UDP.

4. Sieťová vrstva modelu TCP/IP

Aplikačná vrstva vytvára správy pre transportnú vrstvu. Transportná vrstva delí správy na menšie časti, pridáva k nim hlavičky transportnej vrstvy, čím vznikajú segmenty. Segmenty sú ďalej odovzdávané sieťovej vrstve, ktorá ku každému segmentu pridá ďalšiu hlavičku, čím vznikne **datagram**.

Úlohou sieťovej vrstvy je preniesť datagram od odosielateľa k príjemcovi a to čo možno najefektívnejšie, t.j. najrýchlejšou cestou. Zatiaľ čo protokoly transportnej vrstvy sa realizovali iba na koncových zariadeniach komunikácie, sieťová vrstva sa týka okrem koncových zariadení aj všetkých smerovačov (routrov) na ceste od odosielateľa k príjemcovi. Zjednodušene sa dá povedať, že správy sú spracované cieľovou aplikáciou, segment cieľovým počítačom a datagram aj všetkými smerovačmi na ceste od odosielateľa k príjemcovi.

Smerovače sú zariadenia, ktoré sú priamo napojené na viac zariadení (ďalšie smerovače, prepínače (switch-e), koncové zariadenia). Z každého z týchto zariadení smerovač dostáva mnoho datagramov a jeho úlohou je každý z týchto datagramov poslať tým správnym smerom na jeho ceste k cieľovej stanici. Na to, aby smerovač vedel, kam má ktorý datagram odoslať, používa hlavičku datagramu a svoju **smerovaciu tabuľku**, ktorá pre každú cieľovú IP adresu datagramu určuje, ktorým rozhraním má byť daný datagram zaslaný.

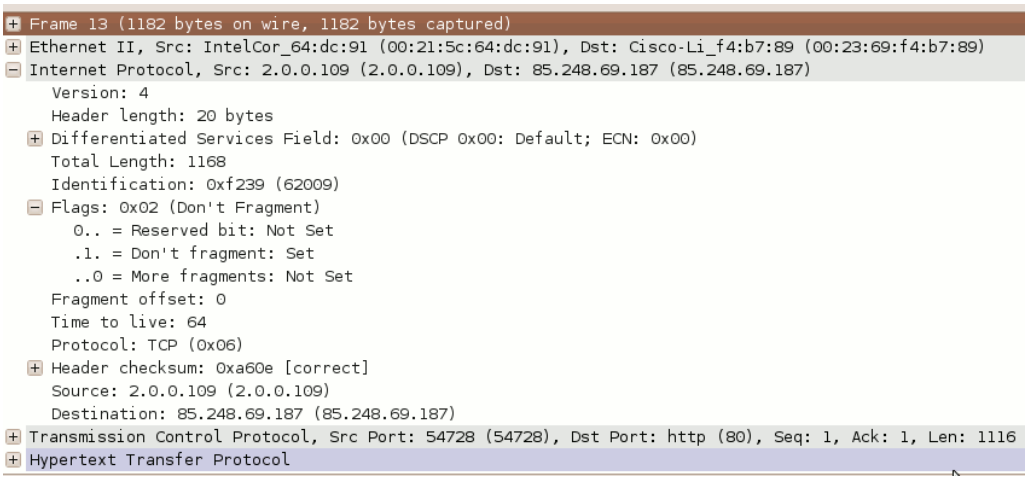
Obsah smerovacej tabuľky menia **smerovacie algoritmy**, ktoré sú opakovane spúšťané smerovačmi, aby reagovali na zmeny v ich okolí (zahltenia resp. uvoľnenia spojení, odstávky iných smerovačov, prerušenia spojov, vytvorenia spojov a tak podobne) a zasielali datagramy stále aktuálne najlepšími smermi.

4.1. Sieťový protokol IP verzie 4

Na stránke http://inetcore.com/project/ipv4ec/index_en.html môžete vidieť javascript, ktorý odrátava počet voľných IPv4 adries.

Hlavička IP protokolu, ktorá sa pridáva pred jej telo (čo je obvykle TCP alebo UDP segment), je pomerne zložitá. Pre naše účely stačí, že vieme, že obsahuje zdrojovú (source) a cieľovú (destination) IP adresu. Keď budeme vysvetľovať protokol ICMP budeme využívať aj hodnotu Time to live (TTL), ktorá určuje, cez koľko maximálne smerovačov môže tento datagram putovať.

V protokole IPv4 sa používajú 32 bitové IP adresy. Tieto adresy sa zapisujú ako 4 osembitové čísla v desiatkovej sústave oddelené bodkami. Napríklad IP adresa 158.197.31.4 je vlastne špeciálne zapísané 32 bitové číslo, ktoré v binárnom zápise vyzerá nasledovne 10011110 11000101 00011111 00000100, lebo 158 v desiatkovej sústave sa binárnej sústave zapíše ako 10011110, 197 ako 11000101, 31 ako 11111 a 4 ako 100.



Obrázok 6. Pohľad na hlavičku IP datagramu v programe Wireshark

Každé sieťové rozhranie pripojené do počítačovej siete, má vlastnú IP adresu. Sieťové rozhranie je logická časť zariadenia ktorá je obvykle priradená jednému fyzickému pripojeniu, typicky sieťovej karte. Jedno koncové zariadenie má obvykle aktívne iba jedno rozhranie pripojené do počítačovej siete. Smerovač má samostatné rozhranie pre každú sieťovú zásuvku (nazývanú port), a pre každé má inú IP adresu.

Sú situácie, keď je vhodné nastaviť na jednej sieťovej karte viac rozhraní s rôznymi IP adresami, napríklad pre virtuálne počítače.

Internet je sieť sietí. Zariadenia z rôznych sietí medzi sebou vedia spolu komunikovať iba cez smerovače, avšak zariadenia, ktoré sú v jednej sieti dokážu komunikovať v rámci svojej siete aj bez účasti smerovača. Zariadenia v rámci jednej siete musia mať IP adresy patriace do rovnakej siete. Čo to znamená, že dve IP adresy patria do rovnakej siete?

Každá stanica má aj virtuálne rozhranie na samého seba (localhost).

Každá IP adresa sa delí na dve časti: časť určujúcu sieť a časť určujúcu stanicu v tejto sieti. IP adresy dvoch počítačov v jednej sieti musia mať časť určujúcu sieť rovnakú. V minulosti, keď bol IP adries dostatok, vznikla kategorizácia IP adries, ktorá iba na základe hodnôt prvých pár bitov IP adresy určovala, aká časť 32 bitovej adresy bude časť určujúca sieť. Boli určené 4 triedy IP adries A až D. V triede A bola sieť určená prvými 8 bitmi a zvyšných 24 bitov označovalo stanicu v tejto sieti. Trieda B mala prvých 16 bitov pre sieť a ďalších 16 pre stanice, trieda C mala prvých 24 bitov pre sieť a posledných 8 pre stanice.

Port na smerovači (routri) alebo prepínači (switch-i) predstavuje zásuvku do ktorej sa vkladá sieťový kábel. Netreba si to mýliť s portom transportnej vrstvy.

Trieda D má všetkých 32 bitov určujúcich sieť, čo znamená, že žiadna stanica nemôže mať adresu typu D. Na prvý pohľad to možno vyzerá nelogicky, no tieto IP adresy sa používajú na multicast. V skratke ide o to, že pri multicaste chce viac staníc komunikovať tak, že datagram s cieľovou multicastovou IP adresou, ktorý vyšle ľubovoľná z týchto staníc, chcú prijať všetky stanice prihlásené do tohto multicastu. Typickým príkladom využitia multicastu môžu byť videokonferencie alebo vysielanie rádií a televízie.

V nasledujúcej tabuľke sú uvedené triedy IP adries. Všimnite si úvodné bity, ktoré určujú triedu adries. Písmenkom **s** je označený "ľubovoľný" bit patriaci sieťovej časti IP adresy a písmenkom **x** je označený bit v časti určenej pre identifikáciu stanice v tejto sieti.

trieda	možné adresy v triede
A	0ssssssx xxxxxxxx xxxxxxxx xxxxxxxx
B	10ssssss ssssssss xxxxxxxx xxxxxxxx
C	110sssss ssssssss ssssssss xxxxxxxx
D	1110ssss ssssssss ssssssss ssssssss

4.1.1. Adresácia CIDR (Classless InterDomain Routing)

Pôvodné delenie má mnohé nevýhody. Keď si vezmeme triedu A, tak tá môže obsahovať zhruba 2^{24} t.j. vyše 16 miliónov počítačov. Mnohé protokoly využívajú obežníkové správy (broadcast messages), t.j. správy určené pre všetky počítače v sieti. Takéto správy od každého z počítačov môžu výrazne zaťažovať sieť. V sieti typu B vieme prideliť IP adresu vyše 65 tisícom počítačov a v sieti typu C môžeme mať maximálne 254 počítačov. Tento skok je dosť výrazný. Čo robiť s organizáciami, ktoré majú zhruba 1000 počítačov a chceli by ich mať v jednej sieti? Ak by sa im pridelila sieť typu B, tak vyše 64 tisíc IP adries ostane nevyužitých. Keďže IP adries je málo, takýto luxus si nemôžeme dovoliť.

Bolo teda určené, že IP adresa môže mať ľubovoľnú časť IP adresy, ktorá určuje sieť. To, koľko bitov IP adresy tvorí identifikácia siete, určuje **maska siete**. Maska siete má rovnako ako IP adresa 32 bitov. Maska vždy vyzerá tak, že v ľavej časti majú všetky bity hodnotu 1 a zvyšok (v pravej časti) tvoria nuly. Ak by sme napríklad chceli, aby sieťová časť IP adresy mala veľkosť 27 bitov, maska by vyzerala tak, že by mala 27 bitov s hodnotou 1 a zvyšných 5 bitov by boli nuly teda 11111111 11111111 11111111 11100000. Aj maska sa zapisuje podobne ako IP adresa, teda ako 4 osembitové čísla v desiatkovej sústave oddelené bodkami. Teda maska s 27 jednotkami v binárnom zápise sa zapíše ako 255.255.255.224, lebo 11111111 je v desiatkovej sústave 255 a 11100000 je v desiatkovej sústave 224.

Informácia, že máme IP adresu 158.197.31.170 s maskou 255.255.255.224, sa dá zapísať aj skrátene, a to **úplnou IP adresou** 158.197.31.170/27. Číslo 27 za lomkou hovorí, koľko bitov masky siete tvoria jednotky.

Načo je nám vlastne dobré vedieť, aká časť IP adresy predstavuje sieť? Ako sme už spomínali, pokiaľ chcú komunikovať dva počítače z rovnakej siete, vedia komunikovať priamo, ale pokiaľ nie sú z rovnakej siete, musia komunikovať cez smerovač. Ako neskôr uvidíme, aj smerovač na základe svojej smerovacej tabuľky musí zisťovať, do ktorej siete má poslať ktorý datagram.

Môžu byť stanice s IP adresami 158.197.31.170/27 a 158.197.31.155/27 v rovnakej sieti? To zistíme tak, že si vypočítame pre každú z týchto adries **adresu siete** a ak sú rovnaké ich adresy siete, potom sú tieto dva počítače v rovnakej sieti. Ak nie sú ich adresy sietí rovnaké, tieto dva počítače nemôžu byť v rovnakej sieti. Adresa siete sa počíta tak, že sa vykoná logický AND IP adresy a masky siete medzi dvojicami bitov na príslušných pozíciách.

```
IP adresa:    10011110 11000101 00011111 10101010 (158.197.31.170)
    maska:    11111111 11111111 11111111 11100000 (255.255.255.224)
              -----------------------------------
adresa siete: 10011110 11000101 00011111 10100000 (158.197.31.160)

   IP adresa: 10011110 11000101 00011111 10011011 (158.197.31.155)
       maska: 11111111 11111111 11111111 11100000 (255.255.255.224)
              -----------------------------------

adresa siete: 10011110 11000101 00011111 10000000 (158.197.31.128)
```

Ako vidíme, stanice s IP adresami 158.197.31.170/27 a 158.197.31.155/27 majú iné adresy siete, a teda nemôžu byť v jednej sieti a ak chcú komunikovať, musia to realizovať prostredníctvom smerovača. Ak by sme ich predsa len zapojili do jednej siete, nevedeli by sa na sieťovej vrstve "porozprávať". Adresa siete je špeciálna adresa, ktorá nemôže byť pridelená žiadnemu rozhraniu žiadnej stanice.

Okrem adresy siete nemôže byť žiadnemu rozhraniu pridelená ani tzv. **obežníková adresa siete** (broadcast address). Tá sa vypočíta ako logický OR medzi IP adresou a inverznou maskou siete. Inverzná maska má vymenené všetky bity masky za opačné.

```
IP adresa:       10011110 11000101 00011111 10101010 (158.197.31.170)
inverzná  maska: 00000000 00000000 00000000 00011111
-------------------------------------------------------
obežník. adr.:   10011110 11000101 00011111 10111111 (158.197.31.191)

IP adresa:       10011110 11000101 00011111 10011011 (158.197.31.155)
inverzná  maska: 00000000 00000000 00000000 00011111
                 ----------------------------------
obežník. adr.:   10011110 11000101 00011111 10011111 (158.197.31.159)
```

Obežníková adresa sa používa, ak chceme, aby odoslaný datagram bol prijatý a spracovaný všetkými rozhraniami v danej sieti. Môžete si všimnúť, že adresa siete tvorí dolnú hranicu rozsahu adries siete a obežníková adresa tvorí hornú hranicu všetkých možných adries v sieti. Všetky adresy medzi adresou siete a obežníkovou adresou môžu byť pridelené rozhraniam zariadení v sieti. Napríklad v sieti s adresou siete 158.197.31.160/27 môžu byť rozhraniam pridelené IP adresy 158.197.31.161 až 158.197.31.190, teda celkovo 30 IP adries.

4.1.1. Delenie siete na podsiete

Predstavte si, že ste sieťoví administrátori v novej firme, ktorá dostala od poskytovateľa pripojenia k dispozícii sieť 158.197.28.0/22. Môžete teda prideľovať počítačom v sieti IP adresy z rozsahu 158.197.28.1 až 158.197.31.254, čo je dokopy 1022 adries. Je však požiadavka, aby kvôli bezpečnosti boli finančné a manažérske oddelenie v samostatnej a zvyšok organizácie môže byť v druhej podsieti.

V tejto sieti je normálnou adresou počítača napríklad aj adresa 158.197.30.0

Máme k dispozícii sieť 158.197.28.0/22. Ak zvýšime počet jednotiek v maske o 1 môžeme ju rozdeliť na dve podsiete 158.197.28.0/23 s rozsahom možných IP adries počítačov 158.197.28.1 až 158.197.29.254 a sieť 158.197.30.0/23 s rozsahom možných IP adries počítačov 158.197.30.1 až 158.197.31.254.

Ak by sme zvýšili počet jednotiek v maske až o 2, dostali by sme 4 podsiete 158.197.28.0/24, 158.197.29.0/24, 158.197.30.0/24 a 158.197.31.0/24. Na rovnakom princípe delenia sietí na podsiete rôznym počtom jednotiek v maskách funguje delenie adries na celom svete.

Najvyššou autoritou na prideľovanie IP adries je organizácia IANA. Ako si môžete prečítať v jej zozname pridelených IP adries, existuje ešte mnoho nepridelených rozsahov IP adries. Mnohé boli pridelené regionálnym organizáciám, ktoré rozdeľujú IP adresy v rámci "kontinentov". V našej oblasti je to organizácia RIPE NCC.

Organizácia IANA:
http://www.iana.org/

Zoznam celosvetovo pridelených adries:
http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml

Organizácia RIPE NCC:
http://www.ripe.net/

4.1.2. Špeciálne IP adresy

Medzi IP adresami je zopár špeciálnych adries, ktoré nesmú byť použité na internete na adresáciu konkrétnych zariadení alebo sietí. Uveďme si tie najzákladnejšie.

Adresu **0.0.0.0** je možné použiť ako zdrojovú adresu stanice lokálnej siete (nie cieľovú adresu). Používa sa pri úvodnom pripojení, keď stanica ešte nevie svoju IP adresu a žiada DHCP server, aby jej nejakú pridelil. Táto adresa sa tiež používa v smerovacích tabuľkách (viď nižšie).

Adresa **255.255.255.255** je obežníková (broadcast) adresa pre lokálnu sieť. Tá sa používa ako cieľová adresa pre všetky rozhrania v lokálnej sieti.

Adresa **127.0.0.0/8** je tzv. loopback, alebo slučka, ktorá určuje "sieť" vlastného zariadenia – samého seba. V praxi sa používa z tohto rozsahu iba IP adresa 127.0.0.1 označovaná ako localhost alebo aj "moja IP adresa". Datagramy s cieľovou adresou v rozsahu siete 127.0.0.0/8 za normálnych okolností nikdy neopustia počítač.

Na realizáciu VPN (Virtual Private Network), sa najčastejšie používa implementácia OpenVPN dostupná na http://openvpn.net/

Pomocou VPN môžeme dosiahnuť, že počítače v iných sieťach majú vytvorenú virtuálnu spoločnú lokálnu sieť.

Adresy z rozsahu sietí **10.0.0.0/8**, **172.16.0.0/12** a **192.168.0.0/16** sú adresy sietí, ktoré sa nevyskytujú vo verejnom internete. Sú určené pre privátne siete. Sú používané hlavne v sieťach s NAT smerovačom, alebo vo VPN sieťach.

Aktivita	Spustite v príkazovom riadku príkaz „ipconfig /all“ a zistite vašu IP adresu, masku a IP adresu smerovača. Určte, v akom rozsahu môžu byť IP adresy vo vašej sieti.
Aktivita	Spustite si v prehliadači stránku http://www.ip-adress.com/ a uvažujte, odkiaľ môže táto stránka vedieť to, čo zobrazuje.
Aktivita	Spustite si v prehliadači stránku http://www.db.ripe.net/whois a vložte do vyhľadávacieho okna svoju IP adresu. Čo nové o vašej adrese ste sa dozvedeli?

4.2. Smerovacia tabuľka a smerovanie datagramov

Každá stanica aj každý smerovač sú schopné odoslať datagram smerom na ľubovoľnú cieľovú adresu na internete. Na určenie toho správneho smeru odoslania datagramu sa používa smerovacia tabuľka. Na základe cieľovej adresy z hlavičky datagramu vieme s pomocou tabuľky určiť, cez ktoré rozhranie je potrebné tento datagram poslať a aj to, ktoré zariadenie bude spracovávať datagram ako ďalšie na jeho ceste k cieľu. Táto druhá informácia je užitočná pre nižšiu vrstvu sieťového rozhrania.

Vezmime si príklad smerovacej tabuľky na smerovači so 4 rozhraniami.

V skutočnosti môžu mať smerovacie tabuľky aj ďalšie stĺpce. Najčastejšie takzvanú metriku, ktorá určuje „rýchlosť“ daného spoja využívanú vtedy, ak je viac možností pre rovnakú kombináciu masky a cieľa.

cieľ	maska	brána	rozhranie
200.23.24.0	255.255.255.0	0.0.0.0	1
200.23.16.0	255.255.248.0	0.0.0.0	3
200.23.24.0	255.255.248.0	0.0.0.0	2
0.0.0.0	0.0.0.0	200.23.1.1	4

Všimnite si, že riadky tabuľky sú usporiadané podľa počtu jednotiek v maske zostupne. Predstavme si teraz, že na tento smerovač príde datagram s cieľovou adresou 200.23.25.8. Do ktorého rozhrania sa má tento datagram odoslať? Algoritmus výberu postupuje tak, že vypočítava logický AND cieľovej IP adresy datagramu a masky v príslušnom riadku smerovacej tabuľky (rovnako sa počíta adresa siete). Ak výsledok je zhodný z hodnotou v stĺpci cieľ, pravidlo je úspešné a datagram sa pošle do toho rozhrania, ktoré uvádza príslušný riadok smerovacej tabuľky. Pravidlá sa vyhodnocujú zhora nadol.

Vyskúšajme si teda výpočet tohto algoritmu pre cieľovú adresu 200.23.25.8. Najprv vypočítame AND s maskou v prvom riadku. Výsledok 200.23.25.0 je rôzny od 200.23.24.0 takže pravidlo je neúspešné. Potom sa vytvorí AND adresy 200.23.25.8 s maskou 255.255.248.0. Výsledok 200.23.24.0 je zhodný s cieľom v treťom riadku, pravidlo je úspešné a datagram je zaslaný cez rozhranie 2.

Všimnime si, že posledný riadok smerovacej tabuľky má masku 0.0.0.0, s ktorou keď urobíme logický AND, výsledok bude vždy 0.0.0.0 bez ohľadu na cieľovú IP adresu. Z toho vyplýva, že ak sa testuje aj posledné pravidlo, je úspešné vždy.

Informácie v stĺpci brána sú potrebné pre nižšiu vrstvu (spomenieme neskôr, keď ju budeme preberať). Ak je v stĺpci brána uvedené 0.0.0.0, znamená to, že cieľová stanica pre ktorú je určený datagram, je v rovnakej sieti ako dané rozhranie. Napríklad vieme, že stanica s IP adresou 200.23.25.8 sa nachádza v rovnakej sieti ako rozhranie smerovača s číslom 2. Samozrejme aj IP adresa tohto rozhrania smerovača je potom z rovnakej siete (napríklad 200.23.25.1).

Ak je v stĺpci brána uvedená nejaká konkrétna IP adresa, v našom príklade 200.23.1.1, ide o adresu najbližšieho smerovača, cez ktorý pôjde datagram na svojej ceste k cieľovej stanici. Ak ide o posledný riadok s cieľom aj maskou rovnou 0.0.0.0, hovoríme o **predvolenej bráne** (default gateway), teda typicky o smerovači, cez ktorý sa spájame s našim poskytovateľom internetového pripojenia.

Smerovaciu tabuľku majú aj všetky rozhrania bežných staníc. Každý bežný počítač má aspoň dve rozhrania – jedno pre sieťovú kartu a druhé pre slučku k sebe samému (loopback, localhost). Ďalšie rozhranie môže byť napríklad pre WiFi.

Aktivita	Vypíšte si svoju smerovaciu tabuľku. V príkazovom riadku zadajte príkaz „route PRINT“ alebo „netstat -r“ vo Windowse alebo „route -n“ v Linuxe. Povedzte, ktoré adresy nebudú zaslané na váš predvolený smerovač, t.j. mimo vašu sieť.

4.2.1. Smerovacie protokoly

Nastavenie smerovacej tabuľky môže byť buď statické alebo dynamické. Pri statickom smerovaní nastavuje smerovaciu tabuľku administrátor ručne. Nevýhoda tohto prístupu môže byť pri výpadkoch alebo zahltení niektorých smerovačov v sieti, lebo ak by aj existovala alternatívna fungujúca cesta k cieľu, smerovač by mohol datagram poslať rovno smerom k nefunkčnému zariadeniu. Pri dynamickom smerovaní sa o nastavenie smerovacej tabuľky stará smerovací algoritmus niektorého z možných smerovacích protokolov.

Smerovacie algoritmy a protokoly, ktoré ich implementujú, slúžia na zistenie aktuálne najoptimálnejšej cesty k všetkým dostupným uzlom siete a na modifikáciu smerovacích tabuliek smerovačov tak, aby datagramy po svojej aktuálne najoptimálnejšej ceste k cieľu naozaj putovali.

Smerovacie protokoly sa delia na **protokoly pre autonómny systém** a na **protokoly medzi autonómnymi systémami**. Autonómny systém tvorí množina smerovačov a ich prepojení spravovaná jedným administrátorom, typicky lokálnym alebo národným poskytovateľom internetového pripojenia. Smerovacie protokoly medzi autonómnymi systémami sú určené na nastavenie smerovania medzi týmito nezávislými autonómnymi systémami.

Najznámejšie smerovacie protokoly pre vnútro autonómnych systémov sú RIP a OSPF. Protokol RIP využíva smerovací algoritmus DVA (distance vector algorithm), pri ktorom si každý smerovač nastavuje smerovaciu tabuľku iba podľa informácií od susedných smerovačov. Naproti tomu protokol OSPF využíva algoritmus LSA (link state algorithm), ktorý je založený na známom Dijkstrovom algoritme. V tomto algoritme musí mať každý smerovač informácie od všetkých smerovačov v autonómnom systéme.

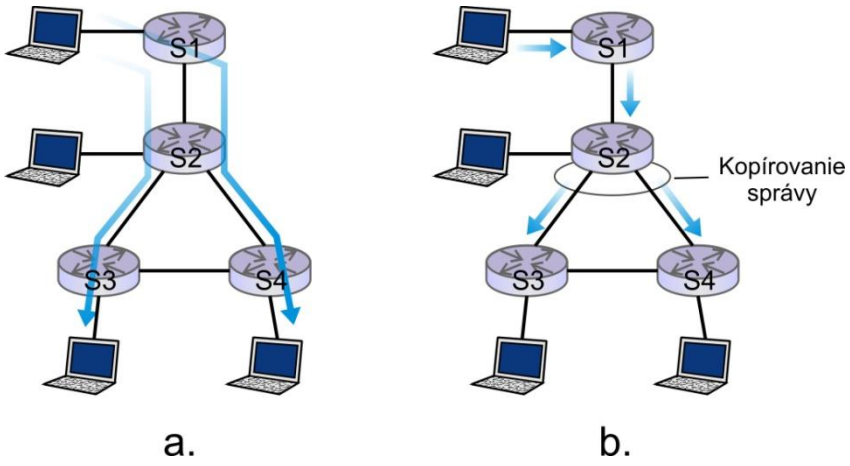
Výber toho, či použiť protokol RIP alebo OSPF je na administrátorovi siete. Oba prístupy majú svoje výhody aj nevýhody. Protokol OSPF je mladší a ponúka oproti protokolu RIP niekoľko vylepšení, ako je autentifikácia smerovačov, integrovaná podpora pre multicastové smerovanie a pre hierarchické smerovanie.

Na komunikáciu medzi autonómnymi systémami je v súčasnosti štandardom smerovací protokol BGP (border gateway protocol). BGP poskytuje každému autonómnemu systému získanie informácií o dostupnosti sietí od susedných autonómnych systémov, prenos týchto informácií ku všetkým smerovačom vo vnútri autonómneho systému, zistenie "dobrých" ciest k sieťam na základe informácií o dostupnosti siete a politiky riadenia autonómneho systému a zasielanie informácií o dostupnosti svojich sietí zvyšku internetu. Táto posledná vlastnosť je kľúčová, inak by vznikali izolované skupiny sietí, ktoré by o sebe navzájom nevedeli.

4.3. Multicastové smerovanie

Okrem smerovania pre **unicast**, teda smerovania komunikácie dvoch staníc navzájom, je vo veľkej miere využívané aj smerovanie pre **multicast**, teda komunikáciu viacerých staníc.

Hlavnou výhodou multicastového smerovania oproti unicastu je, že stanica odosiela každú správu iba raz a dostanú ju všetci členovia multicastovej skupiny. Táto správa sa kopíruje vždy na tých smerovačoch, kde sa rozdeľujú cesty od odosielateľa k ostatným cieľovým staniciam, ktoré sú členmi danej multicastovej skupiny.



Obrázok 7. Vysielanie viacerým príjemcom cez (a) unicast a cez (b) multicast

Multicast sa typicky používa pri rozhlasovom a televíznom vysielaní, ale aj napríklad vo viacpoužívateľských videokonferenciách, gridových výpočtoch, či distribúcii aktualizácií softvérov.

Multicastové smerovanie má za úlohu preniesť správy medzi všetkými členmi danej multicastovej skupiny. Multicastová skupina je identifikovaná špeciálnou IP adresou, ktorá nie je pridelená žiadnej stanici. V protokole IPv4 sme hovorili o triedach IP adries. IP adresy triedy D sú práve IP adresy multicastových skupín. V protokole IPv6 je multicastové adresovanie možné aj v rámci siete či autonómneho systému.

Zaujímavosťou je, že IPv6 nemá podporu pre obežníkové (broadcast) správy v rámci siete.

Multicastový smerovací protokol musí zabezpečiť, aby každý smerovač, na ktorý je napojený člen multicastovej skupiny, dostával spoločné správy multicastovej skupiny. Techník a protokolov na multicastové smerovanie medzi smerovačmi je niekoľko. Najznámejšie protokoly sú DVMRP (distance-vector multicast routing protocol (RFC 1075)) a protokol PIM (protocol independent multicast routing protocol (RFC 3973, RFC 4601,RFC 3569,RFC 4607).

Multicast má podporu aj pre celý internet cez protokol MSDP (Multicast source discovery protocol (RFC 3618, RFC 4611)). Opäť ide o typ smerovania medzi autonómnymi systémami, ktoré funguje ako multicastový variant BGP protokolu.

Multicastové smerovacie protokoly sú určené na komunikáciu medzi smerovačmi. Na komunikáciu medzi stanicou a jej predvolenou bránou (default gateway) slúži protokol IGMP (Internet group management protocol). Má za úlohu umožniť stanici prihlásiť sa k nejakej multicastovej skupine. Protokol IGMP umožňuje smerovaču informovať všetky lokálne siete o ponúkaných multicastových skupinách, ktoré smerovač pozná a ku ktorým sa v prípade záujmu môžu stanice pripojiť. Tieto skupiny sa smerovač dozvie buď od multicastového smerovacieho protokolu, alebo jednoducho preto, že aj iná stanica v niektorej z jeho sietí je členom danej multicastovej skupiny.

Protokol IGMP je popísaný v RFC 3376.

Stanica sa môže prihlásiť k niektorej z ponúkaných skupín tiež cez protokol IGMP. Stanica môže tiež požiadať o členstvo aj v inej multicastovej skupine, ktorá nebola

ponúkaná (používateľ si ju mohol nájsť napríklad cez nejakú webovú stránku). Ak stanica požiada o členstvo v skupine, je už na smerovači, aby pomocou multicastového smerovacieho protokolu zabezpečil, že bude prijímať správy pre túto multicastovú skupinu, aby ich mohol posielať tejto stanici.

IGMP vyžaduje, aby smerovač raz za čas informoval o možnosti byť členom multicastovej skupiny aj stanici, ktorá už je členom. Ak táto stanica neodpovie opätovnou požiadavkou o členstvo v skupine, je z tejto skupiny automaticky vylúčená.

4.4. Aplikačný protokol DHCP

Aj keď DHCP nie je sieťový, ale aplikačný protokol, používa sa na nastavenie sieťových parametrov a preto ho preberáme v časti o sieťovej vrstve.

IP adresa a maska sa dajú nastaviť manuálne cez príslušné konfiguračné nastavenia v operačnom systéme. Aplikačný protokol DHCP však ponúka možnosť nastaviť si IP adresu, masku siete, predvolenú bránu a lokálne predvolené DNS servery dynamicky, bez nutnosti manuálneho nastavovania.

DHCP je aplikačný protokol využívajúci transportný protokol UDP. Typický scenár, kedy sa použije DHCP protokol je pri pripojení zariadenia do siete. Toto zariadenie netuší nič o tom, aká je sieťová adresa tejto siete, ktorú IP adresu si má nastaviť, aká je dĺžka masky a podobne. Tým pádom netuší ani to, na ktorom počítači je spustený DHCP server, na ktorý sa má napojiť a popýtať si od neho potrebné údaje. Jediné, čo vie, je, že ak je DHCP server v danej sieti, tak počúva na porte 67. Vyšle teda DHCP správu (nazývanú DHCP discover) pre všetky počítače v danej sieti, t.j. nastaví ako cieľovú adresu obežníkovú adresu lokálnej siete 255.255.255.255 a ako adresu zdroja nastaví 0.0.0.0, teda stanicu lokálnej siete. Keďže cieľová adresa je obežníková, túto správu musia prijať všetky zariadenia.

Ak je v sieti DHCP server, ponúkne tomuto počítaču novú IP adresu, ale poskytne mu aj informáciu o životnosti IP adresy (kedy končí platnosť), predvolenú bránu a predvolené DNS servery. Potom si už žiadajúci počítač môže nastaviť ponúkanú IP adresu a začať komunikovať s inými počítačmi v sieti alebo cez prednastavený smerovač aj s ostatnými sieťami.

Keďže každá IP adresa pridelená DHCP serverom má svoju životnosť, môže sa stať, že počítaču už pomaly končí pôžička IP adresy (DHCP lease) a môže si požiadať o predĺženie platnosti. Takto sa dá dosiahnuť, že počítač s dynamicky nastavenou IP adresou môže byť pripojený veľmi dlho a stále (bez prestávky) mať pridelenú rovnakú IP adresu.

Protokol DHCP je rozšírením pôvodného protokolu BOOTP, ktorý bol pôvodne určený na pomoc pri bootovaní počítačov po sieti. Vtedy musí DHCP server povedať bootovanému počítaču, kde sa nachádza bootovací súbor. Typicky tento bootovací súbor je uložený na nejakom FTP serveri.

DHCP je veľmi obľúbený protokol najmä súvislosti s masovým rozšírením notebookov a iných mobilných zariadení, ktoré sa bežne pripájajú do rôznych sietí (doma, v škole, v práci, v kaviarni, na stanici, …) a všade si nastavia IP adresu dynamicky, bez potreby ručného nastavovania. Ďalšou výhodou je to, že sa dá vyriešiť aj stav, že má prevádzkovateľ rozsah pridelených IP adries menší ako počet počítačov, ktoré sa v danej sieti zvyknú vyskytovať (nie všetky naraz). Potom sa niektoré IP adresy dajú použiť raz pre jeden, raz pre iný počítač.

Aktivita

Spustite v príkazovom riadku príkaz **ipconfig /all**. Pokúste sa na základe výpisu tohto programu zistiť, či vám vašu IP adresu pridelil DHCP server, aká je vaša predvolená brána a DNS servery. Kedy končí platnosť výpožičky vašej IP adresy?

Aktivita	Ak je váš počítač nastavený tak, že IP adresu získava dynamicky cez DHCP server, spustite v programe Wireshark zachytávanie sieťovej komunikácie. Reštartujte sieťovú kartu . Nájdite a preskúmajte vo vašej komunikácii pakety s DHCP požiadavkou a odpoveďou.

4.5. NAT: Preklad sieťových adries (network address translation)

Úlohou NAT smerovača je sprostredkovať komunikáciu staníc v privátnej neverejnej sieti s verejnou sieťou. NAT je bežnou doplnkovou službou aj lacných smerovačov pre domácnosti. Typická situácia je, že poskytovateľ pripojenia pridelí pre domácnosť alebo malú firmu jedinú IP adresu, ale tí chcú pripojiť na internet viac staníc ako jednu. Keďže každé zariadenie by malo mať pridelenú vlastnú IP adresu, tak bez NAT smerovača by sa muselo žiadať od poskytovateľa pripojenia viac IP adries (a obvykle aj za vyššiu cenu).

Predstavme si, že nám poskytovateľ pripojenia pridelil verejnú IP adresu 138.76.29.7, ale my chceme pripojiť do internetu až 3 počítače. Vieme, že každé rozhranie smerovača musí mať tiež svoju IP adresu. Nastavíme teda pre rozhranie smerovača označené WAN (wide area netvork) IP adresu 138.76.29.7. Na LAN (local area network) rozhraní smerovača nastavíme nejakú neverejnú IP adresu určenú pre privátne siete (pozri vyššie na kapitolu Špeciálne IP adresy) napríklad 10.0.0.4/8 a počítačom v tejto našej privátnej sieti nastavíme postupne IP adresy 10.0.0.1/8, 10.0.0.2/8 a 10.0.0.3/8 (toto nastavenie môžeme vykonať alternatívne aj v DHCP serveri a nechať počítače, nech si tieto IP adresy nastavujú dynamicky).

Keby sme použili obyčajný smerovač bez NAT, tak by nebol problém odoslať datagramy z tejto siete von do internetu, keďže smerovače sa riadia iba cieľovými IP adresami datagramov, ale problém by bol nejaké datagramy prijať. Predstavme si, že chceme z počítača s IP adresou 10.0.0.1 získať webovú stránku z webového servera na adrese 128.119.40.186. Pri pokuse o vytvorenie TCP spojenia dostane tento server datagram s cieľovou IP adresou 128.119.40.186 a portom 80 (známy port pre HTTP). Zdrojová IP adresa by však bola 10.0.0.1. Keďže takáto adresa sa vo verejnom internete nenachádza, odpoveď zo servera by nikdy neprišla, lebo smerovače na internete by nevedeli, kade tento datagram posielať.

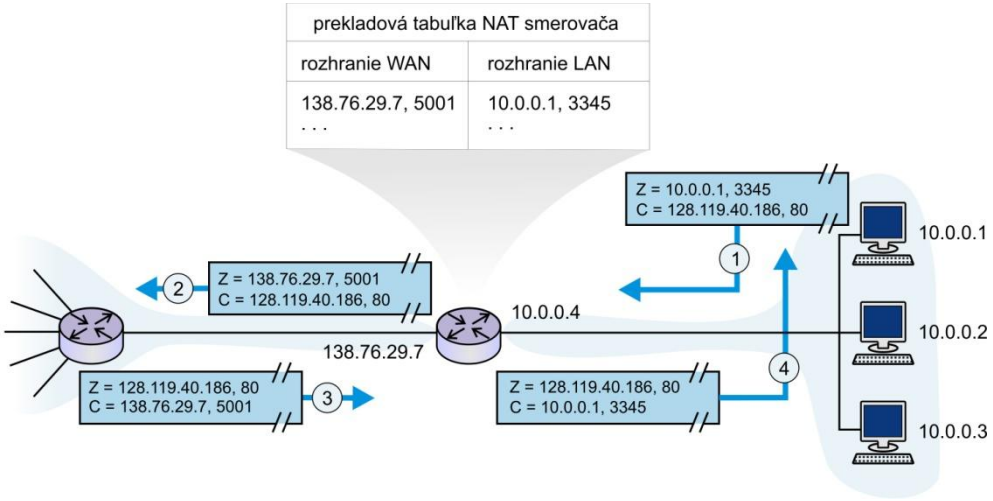
NAT smerovač ukrýva za seba celú privátnu sieť tak, že sa z internetu nejaví ako smerovač, za ktorým je sieť ďalších počítačov, ale ako obyčajný počítač. Ľubovoľný datagram z ľubovoľného počítača v privátnej sieti, ktorý je preposlaný cez NAT smerovač do internetu, je zmenený tak, aby to vyzeralo, že odosielateľ je WAN rozhranie NAT smerovača. Keďže WAN rozhranie NAT smerovača má verejne prístupnú IP adresu, datagramy určené pre túto IP adresu prídu správne k tomuto rozhraniu. Úlohou pre NAT smerovač je teraz zmeniť pre datagramy prichádzajúce z internetu cieľovú adresu tak, aby to bola správna IP adresa a správny port počítača v privátnej LAN sieti (teda v našom príklade adresa 10.0.0.1).

Celý proces je znázornený na obrázku 8.

1. Stanica v privátnej sieti s neverejnou IP adresou 10.0.0.1 vyšle datagram s cieľovou adresou 128.119.40.186 a cieľovým portom 80. Táto stanica očakáva odpoveď na porte 3345, nastaví teda zdrojový port 3345 a zdrojovú IP adresu 10.0.0.1. Keďže adresa 128.119.40.186 nie je v lokálnej sieti, datagram je poslaný cez predvolený smerovač 10.0.0.4.
2. NAT smerovač vezme tento datagram, otvorí jeden z voľných portov pre WAN rozhranie, na našom obrázku port 5001, a zapíše si do prekladovej tabuľky, že ak v budúcnosti príde na WAN rozhranie datagram s cieľovou IP adresou 138.76.29.7 a cieľovým portom 5001, má poslať tento datagram počítaču 10.0.0.1 na port 3345. Následne zmení datagram tak, že zmení zdrojovú IP adresu na IP adresu

WAN rozhrania a zdrojový port na 5001 a odošle datagram smerom k cieľu 128.119.40.186.

3. Keď príde na WAN rozhranie datagram s cieľovou IP adresou 138.76.29.7 a cieľovým portom 5001, Nat smerovač zmení v datagrame cieľovú IP adresu na 10.0.0.1 a cieľový port na 3345 a odošle tento datagram do privátnej siete.
4. Z pohľadu počítača 10.0.0.1 to ani nevyzerá tak, že má neverejnú IP adresu – vlastne ani nevie, že je „za“ NAT smerovačom.



Obrázok 8. Príklad komunikácie cez NAT smerovač

Počet počítačov v privátnej sieti síce nie je limitovaný, ale keďže čísla portov môžu byť z rozsahu 0 až 65535, tak môže byť súčasne aktívnych maximálne 65536 spojení. Nakoľko bežný počítač máva aktívnych iba pár spojení súčasne (alebo pár stovák, ak práve intenzívne používa P2P softvéry), dá sa predpokladať, že okolo 100 počítačov v privátnej sieti by nemalo mať problém komfortne komunikovať s internetom.

Výhodou NAT smerovača je to, že aj v čase, keď začína byť IP adries nedostatok, stále sa dá pripájať mnoho nových zariadení. Nevýhodou je to, že ak chcete otvoriť na niektorom z počítačov serverovú aplikáciu, žiadni klienti z internetu sa na vás nenapoja, lebo nemáte verejnú IP adresu. Ide o takzvaný problém prechodu cez NAT (NAT traversal problem). Týmto problémom dosť trpia P2P softvéry, ktoré predpokladajú, že každý peer je súčasne klient, ktorý sťahuje súbory, ale aj server, ktorý zdieľa a poskytuje stiahnuté súbory.

Problém prechodu cez NAT sa dá vyriešiť niekoľkými spôsobmi:

- Ručné nastavenie časti NAT prekladovej tabuľky na smerovači umožňuje špecifikovať, kam majú byť preposielané datagramy určené pre určité porty WAN rozhrania smerovača. Dá sa napríklad nastaviť aj to, že nové pripojenia na všetky porty WAN rozhrania majú byť preposielané k určenému počítaču. Ručné nastavenie je trochu nepohodlné, ale vcelku efektívne, ak chceme nejaký server prevádzkovať dlhodobo.
- Niektoré NAT smerovače poskytujú na svojom LAN rozhraní službu IGD (Internet gateway device) alebo UPnP (Universal Plug and Play), ktorá umožňuje stanici v lokálnej sieti zistiť si IP adresu NAT smerovača na WAN rozhraní, zistiť aktuálny stav prekladovej tabuľky a hlavne pridávať a odoberať riadky v prekladovej tabuľke NAT smerovača. Tým pádom, ak chce nejaká služba začať počúvať na nejakom svojom porte, môže si nastaviť prekladovú tabuľku na NAT smerovači a tiež informovať okolie o tom, na akej IP adrese a porte bude NAT smerovač počúvať (a ten bude odtiaľ preposielať na túto stanicu s daným portom)
- Tretie riešenie je využiť nejaký počítač s verejnou IP adresou a všetku komunikáciu daného spojenia preposielať cez neho. Toto riešenie využíva napríklad Skype. Ak sú obaja účastníci za svojimi NAT smerovačmi, vyberie sa stanica s verejnou IP adresou, na ktorú sa napojí volaný aj volajúci a celý

rozhovor je potom preposielaný cez túto stanicu.

Aktivita	Zapojte si k počítaču nejaký WiFi smerovač a napojte sa na jeho webového rozhranie. Preskúmajte možnosti nastavenia – NAT, UPnP, port forwarding, klonovanie MAC adresy, DHCP server, šifrovanie WEP, WPA.

4.6. Protokol ICMP

Protokol ICMP (Internet Control Message Protocol) používajú stanice a smerovače, aby sa navzájom informovali o situácii v sieťovej vrstve. Najčastejšie použitie ICMP je na chybové správy o nedostupnosti siete, stanice, protokolu alebo portu transportnej vrstvy.

ICMP protokol je sieťový protokol, ktorý sa prenáša v tele IP datagramu podobne ako TCP alebo UDP protokoly. Niekedy je preto považovaný za transportný protokol. My ho však za transportný považovať nebudeme, lebo neprenáša žiadne dáta aplikačnej vrstvy (nerealizuje „transport“ správ aplikačnej vrstvy).

ICMP sa používa aj na diagnostiku siete. Ak chceme zistiť či je stanica s nejakou IP adresou dostupná, použijeme program **ping**, ktorý vyšle ICMP paket nazvaný "echo request" a ak je stanica dostupná (a nemá nastavený firewall tak, že ICMP pakety neprijíma), tak odpovie ICMP paketom nazvaným "echo reply", ktorý program ping zachytí a vypíše to, ako dlho na túto odpoveď čakal. V prípade, že paket „echo reply“ nepríde, môže prísť od niektorého smerovača na ceste medzi odosielateľom a príjemcom ICMP paket informujúci o nedostupnosti cieľového počítača (destination host unreachable) ak je počítač vypnutý, alebo siete (destination network unreachable) ak je v sieti nejaký problém, o čom program ping hneď informuje používateľa. Poslednou možnosťou je, že cieľový počítač je síce dostupný, ale je nastavený tak, aby na ICMP pakety neodpovedal. V tomto prípade ping nedostane žiadnu odpoveď, o čom môže vypísať informáciu „request timed out“, teda, že sa nedočkal odpovede v nejakom danom časovom intervale.

Veľmi užitočným pomocníkom na diagnostiku siete je aj program **tracert**. Využíva nastaviteľnú životnosť datagramu. Každý datagram má v hlavičke nastavenú číselnú hodnotu TTL (time to live), ktorá hovorí, cez koľko smerovačov ešte môže tento datagram putovať k cieľu. Hodnota TTL je na každom smerovači znížená o 1. Ak na niektorom smerovači dosiahne datagram hodnotu TTL=0, vyhlási sa za datagram, ktorému skončila životnosť a zahodí sa. Smerovač, ktorý tento datagram zahadzuje, posiela pôvodnému odosielateľovi ICMP paket nazvaný „TTL expired“ (vypršala životnosť).

Tracert vysiela datagramy najprv s TTL=1, potom s TTL=2 a tak ďalej, až pokiaľ niektoré datagramy "neprežijú" celú cestu k cieľovej stanici. To znamená, že najprv prídu ICMP správy „TTL expired“ z najbližšieho smerovača, potom z druhého na ceste k cieľu, a tak ďalej až príde odpoveď od samotného cieľa.

Ak je niektorý smerovač na ceste vypnutý, či nefunkčný, dostaneme, podobne ako v prípade programu ping, ICMP paket informujúci o nedostupnej sieti od smerovača pred ním. Ak je vypnutá iba cieľová stanica, príde z posledného smerovača správa „destination host unreachable“ (nedostupná stanica).

Aktivita	Vyskúšajte si programy **ping** a **tracert** pre vybraný cieľový počítač (napr. *www.sme.sk*). Ak máte možnosť, vytiahnite z vášho smerovača sieťový kábel, ktorý smeruje do internetu a vyskúšajte programy ping a tracert znova. Porozmýšľajte, ako by sa dalo v prípade nefunkčnosti internetového pripojenia identifikovať chybné zariadenie.

4.7. Protokol IP verzie 6

Protokol IPv4 by sa mal postupne nahrádzať protokolom IPv6. Hlavnou zmenou je dlhšia IPv6 adresa, ktorá už nezaberá 32, ale až 128 bitov. Tým pádom by už nikdy nemal vzniknúť stav, že sa voľné IP adresy začnú rýchlo míňať, ako to začíname pociťovať pri protokole IPv4.

Podľa posledných odhadov by sa mali vyčerpať všetky voľné IPv4 adresy už v polovici roku 2011.

Adresy v protokole IPv6 majú 128 bitov. Zapisujú sa ako osem dvojbajtových slov v šestnástkovej sústave oddelených dvojbodkou. Napríklad IPv6 adresa fe80:0000:0000:0000:0221:5cff:fe64:d39a má prvé dvojbajtové slovo fe80, ktoré v dvojkovej sústave vyzerá nasledovne 1111 1110 1000 0000, ostatné si môžete vyjadriť sami podľa známych prevodov medzi číselnými sústavami.

Okrem úplného tvaru IPv6 adresy máme možnosť aj skrátených vyjadrení. V nasledujúcej tabuľke označujú všetky riadky tú istú adresu.

tvar IPv6 adresy	vysvetlenie
fe80:0000:0000:0000:0221:5cff:fe64:d39a	úplný tvar, každá štvorica bitov je zobrazená ako číslica v šestnástkovej sústave
fe80:0:0:0:221:5cff:fe64:d39a	odstránené úvodné nuly z každého dvojbajtového slova
fe80::221:5cff:fe64:d39a	vyjadrenie „::“ znamená, že je potrebné doplniť toľko nulových dvojbajtových slov, aby bolo dokopy osem dvojbajtových slov; výraz :: sa môže v každej adrese použiť len raz
fe80::221:5cff:254.100.221.154	posledné dve dvojbajtové slová sú vyjadrené ako štyri jednobajtové slová v desiatkovej sústave oddelené bodkami; hovoríme o mixovanej notácii, keďže posledné 4 bajty vyzerajú ako IPv4 adresa

To, aká časť IPv6 adresy predstavuje sieťovú časť sa označuje, podobne ako v IPv4, lomkou a počtom jednotiek v maske v desiatkovej sústave. Napríklad fe80::221:5cff:fe64:d39a/48 určuje, že prvých 48 bitov je sieťová časť a zvyšných 80 bitov určuje stanicu v tejto sieti.

Podobne ako v IPv4 aj v IPv6 je niekoľko adries vyhradených na špeciálne účely.

Celá adresácia, vrátane špeciálnych IPv6 adries je popísaná v RFC 4291

- **0:0:0:0:0:0:0:0** alebo aj **::0** má rovnaký význam ako adresa **0.0.0.0** v IPv4, a určuje nešpecifikovanú adresu zdroja v lokálnej sieti.
- **0:0:0:0:0:0:0:1** alebo aj **::1** určuje localhost, teda samého seba, a je ekvivalentom z 127.0.0.1 z IPv4.
- **0:0:0:0:0:ffff:IPv4** adresa alebo aj **::ffff:IPv4** adresa, napríklad **::ffff:158.197.31.4**, slúži na reprezentáciu rozhraní s IPv4 adresami ako rozhraní s IPv6 adresami (viac o prechode z IPv4 na IPv6 je popísané nižšie).
- **fe8X:hocičo**, **fe9X:hocičo**, **feaX:hocičo** a **febX:hocičo** slúžia ako adresy pre lokálnu sieť. Zaujímavosťou je to, že stanica si sama môže nastaviť vlastnú unikátnu lokálnu IPv6 adresu tak, že zakomponuje do IPv6 adresy s prefixom fe80 svoju MAC adresu, ktorá by mala byť vždy unikátna na svete a ktorá má 6 bajtov. Takto vygenerovaná IPv6 adresa sa dá používať na komunikáciu s počítačmi vo vnútri lokálnej siete. To znamená, že ak nejakí (hoci aj neskúsení) používatelia vezmú niekoľko počítačov a napoja ich cez obyčajný prepínač (switch), alebo dokonca na priamo, tieto počítače si sami nastavia svoje IPv6 adresy bez DHCP servera bez potreby ručnej konfigurácie, a môžu začať komunikovať – hrať hry, zdieľať súbory a pod.
- **ffXY:hocičo** sú určené pre multicast. Namiesto hocičo si treba dosadiť identifikáciu multicastovej skupiny. Podobne ako v IPv4, multicastová adresa nesmie byť pridelená žiadnemu rozhraniu.
    - **ffX2:hocičo** je multicast pre lokálnu sieť, **ffX8:hocičo** je multicast pre

sieť vlastnej organizácie, aj keď má vo vnútri viac podsietí. Tieto multicastové IP adresy si môžu nastavovať administrátori siete sami podľa ľubovôle, lebo tieto adresy nie sú prístupné z verejného internetu. Tieto adresy sa dajú využiť ako náhrada za obežník (broadcast) 255.255.255.255 z IPv4, ktorý je v IPv6 zrušený. Pre každý známejší protokol, ktorý v IPv4 používal obežníky je vyhradená dobre známa multicastová IPv6 adresa. Napríklad ak chceme kontaktovať DHCPv6 server môžeme použiť multicastovú adresu ff02::1:2, keďže ide o dobre známu lokálnu multicastovú IPv6 adresu, a datagramy s touto cieľovou adresou podľa štandardu spracúvajú všetky DHCPv6 servery.
- **ffXe:hocičo** je multicast pre celý internet. Tieto adresy sú prideľované lokálnymi organizáciami pod organizáciou IANA (u nás RIPE NCC).

4.7.1. Prechod z IPv4 na IPv6

Prechod na IPv6 je náročný proces. Keďže do internetu je zapojených niekoľko miliárd zariadení s rôznym hardvérovým a softvérovým vybavením, nedá sa jednoducho spraviť to, že sa vyhlási deň a hodina, keď sa všetci prepnú na vyššiu verziu IP protokolu. Okrem operačných systémov koncových zariadení musia byť na prechod postupne pripravené aj všetky sieťové aplikácie a všetky smerovače. Tiež je potrebné vykonať príslušné zmeny aj na DNS serveroch. Je teda nutné riešiť to, aby bolo súčasne možné fungovať nad oboma verziami IP protokolov.

Istú pomoc pre aplikácie fungujúce nad IPv6 protokolom je možnosť reprezentácie IPv4 adresy v IPv6 formáte. To znamená, že po sieti môžu putovať IPv4 datagramy ale sieťová vrstva na cieľovom počítači môže vykonať preklad na IPv6 formát.

Kompletná idea prechodu z IPv4 na IPv6 je popísaná v RFC 4038.

Ak komunikujú dve aplikácie nad IPv6 protokolmi, je možné IPv6 datagramy vložiť do vnútra IPv4 datagramov a preniesť tak tieto datagramy cez IPv4 siete.

5. Vrstva sieťového rozhrania modelu TCP/IP

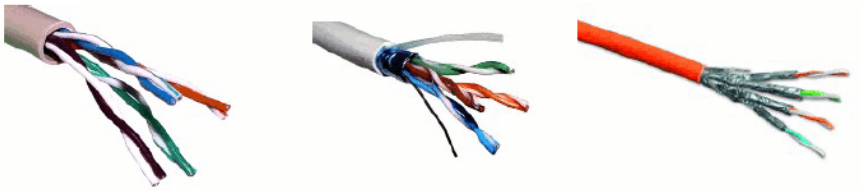
Už vieme, že sieťová vrstva má za úlohu dostať datagram z jedného koncového zariadenia na iné kdekoľvek na internete. Využíva pri tom adresáciu cez IP adresy, ktoré sú prideľované podľa siete, v ktorej sa dané koncové zariadenie nachádza. Ide o hierarchickú adresáciu, ktorá umožňuje určiť presne „polohu“ zariadenia v rámci internetu kdekoľvek na zemeguli.

Vrstva sieťového rozhrania, ktorá je najnižšou vrstvou referenčného modelu TCP/IP je využívaná sieťovou vrstvou na prenos datagramov v rámci jednej siete, presnejšie v rámci každej siete, cez ktoré prechádza datagram na svojej ceste k cieľu. Prenosové technológie tejto vrstvy sú prispôsobené prenosovému médiu, ktorým sa dáta prenášajú. Iná prenosová technológia je vhodná pri komunikácii po kovovom spoji, iná v optickom vlákne a iná pri bezdrôtovom prenose. Datagram pri svojej ceste z jedného miesta na druhé v rámci internetu, môže prechádzať rôznymi typmi spojov a využívať pri tom rôzne prenosové technológie.

5.1. Prenosové médiá

Prenosové média v počítačových sieťach sú obvykle kovové, optické alebo bezdrôtové. Na prenos informácie sa používa buď digitálny alebo analógový (modulovaný harmonický) signál. Prenos digitálnym signálom je presnejší, ale pri prenose na väčšie vzdialenosti nepoužiteľný. Vtedy je vhodnejšie použiť analógový prenos, ktorý je menej citlivý na rušivé vplyvy okolia. Na zmenu digitálneho na analógový signál a naopak slúži zariadenie nazývané **modem**.

Analógový signál sa používa aj pre prenos binárnej informácie cez menej kvalitné, prevažne telefónne káble, kde hrá nižšia citlivosť na vonkajšie vplyvy významnú úlohu. Táto citlivosť na okolité signály v kovových prenosových médiách sa čiastočne rieši jednak pravidelným špirálovitým stočením vodičov (tzv. krútený dvojdrôt) a dodatočným tienením alebo využitím koaxiálneho kábla.



Obrázok 9. Typy krútených dvojdrôtov: a) netienený, b) chránený (screened), c) tienený

Na prenos analógovým signálom sa používajú obvykle také modulačné techniky, ktoré v jedinej zmene vlastnosti prenášaného harmonického signálu (amplitúdy, frekvencie, fázy) vyjadria viac bitov naraz. Digitálny prenos využíva signál s dvoma rôznymi hodnotami meranej veličiny (napätie, svetlo). Pri digitálnom prenose je nutné zabezpečiť synchronizáciu odosielateľa a príjemcu, aby boli jasné hranice toho, ktorá časť signálu predstavuje jeden bit v príde bitov.

Koaxiálny kábel

V súčasnosti najpoužívanejšie káblové prenosové médiá sú krútený dvojdrôt, koaxiálny kábel a optický kábel. Koaxiálny kábel má výborné tienenie a preto aj vysokú teoretickú prenosovú rýchlosť. Kedysi sa často využíval na spájanie počítačov v lokálnej sieti. V súčasnosti sa využíva hlavne na prenos signálov analógovej aj digitálnej televízie.

Krútený dvojdrôt používaný v súčasných lokálnych sieťach je dnes už využívaný tak, že používané prenosové rýchlosti (do 1 Gb/s) sú blízko teoretického maxima.

Konektor RJ 45 používaný v kovovom Ethernete ako koncovka pre krútený dvojdrôt

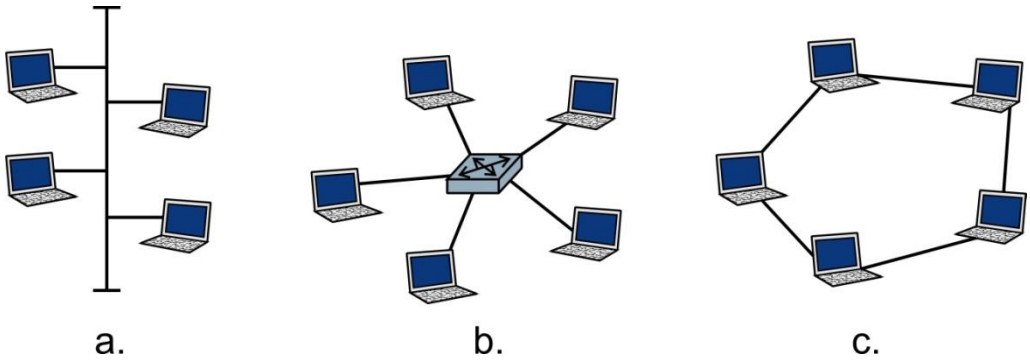
Obrovský prenosový potenciál majú optické káble, pri ktorých využívame iba maličký zlomok ich maximálnej prenosovej rýchlosti. Prenosová rýchlosť je brzdená „pomalými“ koncovými zariadeniami (dnes je to bežne 10 Gb/s). Nevýhodou optických káblov je ich cena, krehkosť, obmedzená ohybnosť a náročné pripájanie.

Optické vláka poznáme jednovidové a mnohovidové. Mnohovidové prenášajú súčasne viac lúčov svetla, jednovidové iba jeden lúč. Jednovidové sú kvalitnejšie a schopné prenášať digitálny signál na veľké vzdialenosti. Oproti mnohovidovým vláknam, ktoré majú priemer 50-62 μm, sú tenšie, majú priemer iba 4-10 μm, a teda sú aj náročnejšie na spájanie.

Pri bezdrôtových prenosoch sa dajú využiť iba Telekomunikačným úradom schválené frekvenčné pásma, keďže rozsah frekvencií „vo vzduchu“ je zdieľaný aj vysielačkami, mobilnými operátormi, radarmi, bezšnúrovými perifémymi zariadeniami, rádiami, televíziami a podobne. Na krátke vzdialenosti (cca do 20 km) je možné využívať digitálne vysielanie, ale na väčšie vzdialenosti je už vplyv okolitého šumu taký veľký, že je nutné používať modulované harmonické signály – napríklad na spojenia s družicou.

5.2. Základné topológie počítačových sietí

Obrázok 10 znázorňuje tri základné topológie počítačových sietí. Pri zbernicovej topológii sú všetky zariadenia pripojené na spoločné komunikačné médium. Signál vyslaný z ľubovoľnej stanice sa šíri všetkými smermi a prijme ho tak každá stanica v sieti. Pri topológii hviezdy je prirodzené použitie nejakého aktívneho centrálneho zariadenia na ktoré je pripojených viac staníc. Týmto centrálnym zariadením môže byť typicky rozbočovač (hub) alebo prepínač (switch). Pri kruhovej topológii sú stanice spojené po dvojiciach tak, aby tvorili kruh. Signály sa v kruhovej topológii šíria buď jedným alebo druhým smerom. V súčasnosti je najčastejšie zapojenie do hviezdy, prípadne, s použitím viacerých centrálnych zariadení, sa vytvára zložitejšia hierarchická topológia.



Obrázok 10. schémy topológií počítačových sieti: a. zbernicová, b. hviezda, c. kruhová

5.3. Prenosové technológie

Každá prenosová technológia vrstvy sieťového rozhrania predstavuje súbor špecifikácií technických zariadení, prenosových médií, koncoviek a zásuviek, metód kódovania signálu, možných topológií siete a riadenia komunikácie zariadení na spoločnej sieti.

Najpoužívanejšou prenosovou technológiou na komunikáciu cez drôtové spoje je **Ethernet**, ktorý sa najčastejšie používa pri spojoch realizovaných cez krútené dvojdrôty (tenký Ethernet), koaxiálne káble (hrubý Ethernet) alebo cez optické vlákna. Ďalšími známymi prenosovými technológiami sú **ISDN** (Integrated Services Digital Network) a **DSL** (Digital Subscriber Line) určené pre telefónne káble, prípadne **FDDI** (Fiber Distributed Data Interface) pre káblové siete s kruhovou topológiou.

Medzi bezdrôtovými prenosovými technológiami je najpoužívanejšia technológia bezdrôtovej lokálnej siete (WLAN – Wireless local area network) založená na štandardoch **IEEE 802.11**, ktorá sa označuje ako **WiFi**. Obľúbenou bezdrôtovou technológiou je aj **Bluetooth**.

Všimnite si, že zatiaľ čo protokoly vyšších vrstiev spravuje organizácia IANA v RFC špecifikáciách, na tejto vrstve už štandardy technológií spravuje organizácia IEEE.

V tomto texte sa zameriame iba na najrozšírenejšie technológie Ethernet a WiFi. Obe tieto technológie používajú vlastnú adresáciu zariadení cez **MAC adresy** (MAC = media access control), niekedy nazývané aj ako fyzické adresy, hardvérové adresy alebo ethernetové adresy. MAC adresy sú prideľované výrobcom zariadenia a sú nezávislé od siete, v ktorej sa nachádzajú. Každý výrobca má organizáciou IEEE daný povolený rozsah MAC adries, ktoré môže zariadeniam prideľovať, aby sa zabezpečila jedinečnosť každej adresy a nemohli sa tak stretnúť v jednej sieti dve zariadenia s rovnakou MAC adresou. Ide vlastne o akési „výrobné číslo“ sieťovej karty, ktoré je s ňou pevne zviazané a nemenné bez ohľadu na polohu na zemeguli. Na základe MAC adresy nevieme určiť, kde sa daná sieťová karta (a teda zariadenie v ktorom je namontovaná) v rámci internetu nachádza, pokiaľ sa nenachádza v našej sieti.

Ako analógiu si môžeme predstaviť, že IP adresa je niečo ako poštová adresa, na ktorej sa človek nachádza, a MAC adresa je niečo ako rodné číslo, ktoré sa nemení v závislosti od toho, kde sa jeho nositeľ nachádza.

MAC adresa má 6 bajtov. Obvykle sa píše hexadecimálne po jednotlivých bajtoch oddelených dvojbodkou. Napríklad A1:B2:C3:D4:56:FF.

Sieťová vrstva posiela vrstve sieťového rozhrania na spracovanie datagramy. Tá každý datagram zabalí do **rámca** (pridá k datagramu svoju hlavičku a pätu). Rámec už predstavuje finálnu postupnosť jednotiek a núl, ktoré budú vyslané do spoja. Ide teda o posledné obalenie pôvodnej odosielanej správy pri ceste vrstvami od aplikačnej až po vrstvu sieťového rozhrania.

5.4. Ethernet

Popri množstve prenosových technológií patrí Ethernet, definovaný v štandarde IEEE 802.3, medzi tie najrozšírenejšie technológie v káblových LAN sieťach.

Hlavička rámca Ethernetu je veľmi jednoduchá. V prvých 8 bajtoch, nazývaných preambula, obsahuje len synchronizačné bity, ktoré vyzerajú tak, že sa striedajú hodnoty 1 a 0 a na konci sú dve jednotky (101010...10101011). Potom nasledujú cieľová a zdrojová MAC adresa a označenie typu datagramu, ktorý je prenášaný v rámci. Po tejto hlavičke už nasleduje samotné telo rámca, teda obvykle IP datagram, ktorý môže mať veľkosť 46 až 1500 bajtov(platí pre Ethernet s prenosovou rýchlosťou 100 Mb/s). Po tele rámca nasleduje päta rámca, konkrétne 4 bajty kontrolných bitov CRC kontroly (pozri nižšie).

8 bajtov	6 bajtov	6 bajtov	2 bajty	46 – 1500 bajtov	4 bajty
preambula	cieľová MAC adresa	zdrojová MAC adresa	typ protokolu v tele rámca	telo rámca	CRC

Obrázok 11. Rámec Ethernetu

5.4.1. Kontrola cyklickým polynómom (CRC)

Pri prenose binárnej informácie spojom môže dôjsť k znehodnoteniu signálu. Chyby sú spôsobované rôznymi vplyvmi externého prostredia na fyzický spoj (interferenciou prostredia s prenášaným signálom). Najcitlivejšie na rušivé vplyvy okolia sú hlavne netienené drôty a bezdrôtové spoje. Toto znehodnotenie signálu má často za dôsledok zmenu niektorých z prenášaných bitov na opačné. Odhaľovanie chýb pri prenose rámcov slúži na to, aby si príjemca bol dostatočne istý tým, že prijal presne tie isté dáta, ktoré boli odoslané odosielateľom.

Kontrola sa realizuje tak, že sa k odosielaným dátam pribalia ďalšie tzv**. kontrolné bity**. Chyba môže nastať v pôvodných dátach, ale aj v kontrolných bitoch. Odhalenie všetkých druhov chýb nie je stopercentné. Čím viac je kontrolných bitov, tým je väčšia šanca na odhalenie prípadnej chyby. Samozrejme závisí aj na spôsobe, akým sa samotná kontrola realizuje. Najpoužívanejšia forma kontroly je **CRC** (cyclic redundancy check), ktorú prekladáme ako **kontrola cyklickým polynómom**, keďže delenie cyklickým polynómom je využité v „matematickom pozadí“ tejto metódy. My sa však pozrieme iba na to, ako by sme si vedeli sami spraviť CRC kontrolu „na papieri“.

Predpokladom kontroly CRC je, že odosielateľ aj príjemca používajú rovnaký kontrolný polynóm. V sieťach typu Ethernet sa používa polynóm stupňa 32. Pre účely vysvetlenia fungovania použijeme iba polynóm stupňa 3: $\mathbf{1}x^3+\mathbf{0}x^2+\mathbf{1}x+\mathbf{1}$. My z neho použijeme len jeho koeficienty 1011.

Ethernet používa CRC polynóm $x^{32}+ x^{26}+ x^{23}+ x^{22}+ x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^8+ x^7+ x^5+ x^4+ x^2+ x + 1$

Predpokladajme, že odosielateľ chce poslať dáta 10101 a chce k nim pridať CRC kontrolu s využitím tohto nášho kontrolného polynómu. Na výpočet obsahu kontrolných bitov budeme postupne používať bitovú operáciu XOR. Začneme tým, že k pôvodným dátam dopíšeme toľko núl, aký je stupeň kontrolného polynómu, v našom prípade tri. Dostávame 10101**000**. Ukážme si ako prebieha výpočet.

```
    10101000
XOR 1011
    ----
    00011000
   XOR 1011
       ----
       01110
    XOR 1011
        ----
         101
```

Vidíme, že koeficienty kontrolného polynómu 1011, sme použili na operáciu XOR vždy so zarovnaním na prvú jednotku predchádzajúceho výsledku. V tomto prípade to znamená, že na začiatku si číslo 1011 (za ním si predstavíme 4 nuly) napíšeme pod číslo 10101000 so zarovnaním doľava a urobíme XOR. Výsledkom je 00011000. Keď si odmyslíme prvé tri nuly tak máme zvyšok 11000. Pod neho si zapíšeme známe

číslo 1011 so zarovnaním doľava (za ním si predstavíme ešte jednu nulu) a vypočítame XOR. Výsledok je 01110. Urobíme ešte jeden XOR s číslom 1011. Výpočet končí, keď výsledok 101 je už maximálne taký dlhý, ako je stupeň kontrolného polynómu. Odosielame dáta 10101 spolu so zvyškom 101 teda 10101101.

Čo s tým spraví príjemca? Vezme doručenú postupnosť 10101101 a použije na ňu ten istý postup ako odosielateľ, s tým rozdielom, že na začiatku nedopisuje nuly. Ak je konečný výsledok výpočtu nula, považujeme postupnosť za neporušenú. Inak, teda ak výsledok nie je nula, odhalili sme chybu. Napíšme si výpočet na strane príjemcu

```
    10101101
XOR 1011
    ----
    00011101
   XOR 1011
       ----
       01011
    XOR 1011
        ----
        0000
```

Pekné animácie o tom, ako výpočet CRC prebieha cez výpočtový obvod, môžete nájsť napríklad na Wikipedii: http://en.wikipedia.org/wiki/Computation_of_CRC

Ako vidíme, výpočet „na papieri“ je vcelku jednoduchý. Reálne sa ale výpočet realizuje inak, aby bol výpočet rýchly a hlavne jednoducho hardvérovo realizovateľný.

CRC je metóda, ktorá má dve výborné vlastnosti. Prvou je ľahká a výpočtovo rýchla hardvérová implementácia a druhá je vysoká úspešnosť odhaľovania chýb – až 99,99999998 % chýb pri použití 32 kontrolných bitov. Naproti tomu kontrolný súčet, ktorý sa realizuje v transportnej vrstve, je náročné počítať hardvérovou implementáciou a je určený skôr na softvérové spracovanie. Pritom úspešnosť kontrolného súčtu je iba niečo vyše 95 percent.

5.4.2. Prístupová metóda CSMA/CD

Základné myšlienky protokolu CSMA/CD pochádzajú z protokolu ALOHA. Sieť ALOHAnet vznikla v roku 1970 na univerzite na Havaji na realizáciu lacnej bezdrôtovej siete.

Ethernet v kovových drôtoch využíva prístupovú metódu CSMA/CD (*carrier sense multiple access and collision detection*). Ethernet je technológiou, ktorá predpokladá zbernicovú topológiu alebo topológiu hviezdy. Predpokladá, že na jednom spoji môže byť pripojených súčasne viac zariadení, ktoré chcú všetky cez tento spoj komunikovať. Hovoríme, že ide o prístupovú metódu s viacnásobným prístupom.

Ethernet predpokladá, že jednotlivé uzly okrem toho, že zdieľajú spoločné komunikačné médium, používajú aj rovnaké vysielacie frekvenčné pásma. Pri súčasnom vysielaní ľubovoľných dvoch uzlov tak dochádza k interferencii ich vysielaní, čím sa oba vyslané signály navzájom znehodnotia. V takom prípade hovoríme o **kolízii**. Všetky prístupové metódy s viacnásobným prístupom sa musia s kolíziami nejako vysporiadať, nakoľko základná podmienka na úspešný prenos rámca je, aby počas celej doby vysielania a prenosu rámca nedošlo ku kolízii. Tieto prístupové metódy majú teda dva hlavné problémy: **ako odhaliť kolíziu** a **ako sa zotaviť z toho, ak kolízia nastane**.

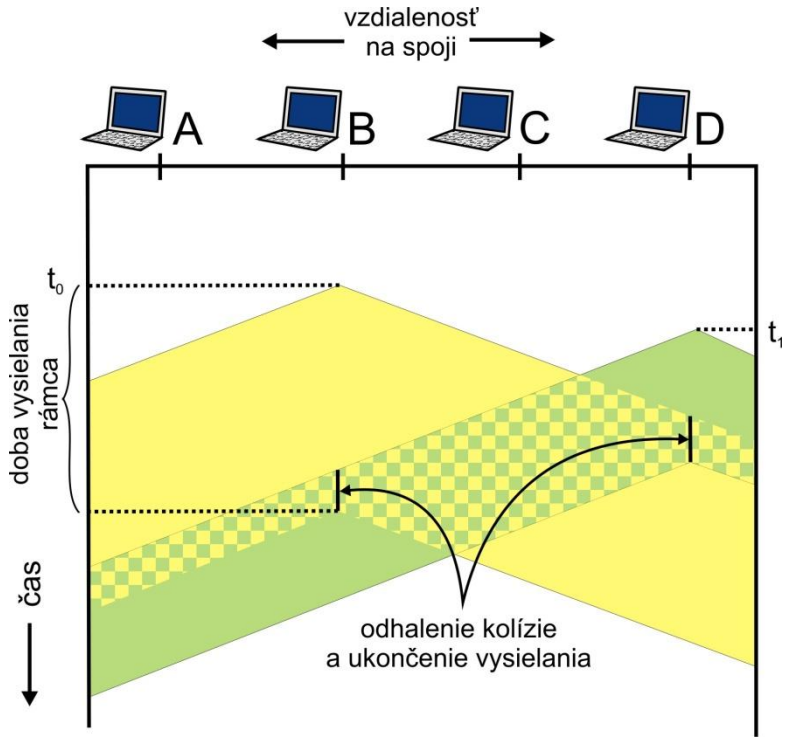
Odhalenie kolízie počas vlastného vysielania je jednoducho realizovateľné v káblových spojoch. Väčšie problémy už spôsobuje odhalenie kolízie pri bezdrôtových spojeniach, kedy energia vysielania je obvykle mnohokrát väčšia ako energia signálov prijímaná z iných uzlov. Preto sa pri bezdrôtových spojoch CSMA/CD nepoužíva. Namiesto toho sa pri bezdrôtových spojoch používajú iné protokoly napríklad CSMA/CA alebo WiMAX.

V snahe zabrániť kolízii, prístupová metóda CSMA/CD určuje, že uzol nesmie začať vysielať, ak na spoji registruje vysielanie niektorého iného uzla. To znamená, že nezačne vysielať, keď je evidentné, že by spôsobil kolíziu. Kolízie však môžu naďalej nastávať. Dôvodom je to, že signál vysielaný jedným uzlom sa šíri spojom „iba“ takmer svetelnou rýchlosťou a pokiaľ dôjde k iným uzlom, ubehne nejaký čas, počas ktorého môžu tieto uzly začať vysielať, neuvedomujúc si, že spôsobia kolíziu, keďže pri začiatku svojho vysielania neregistrujú vysielanie iného uzla.

Aj počas svojho vysielania musí uzol počúvať, či náhodou niektorý iný uzol nezačal vysielať. Ak počas vysielania nastane kolízia, je okamžite toto vysielanie prerušené. Keby sme nezisťovali kolízie počas vlastného vysielania, obsadili by sme na dlho

spoj, aj keď je jasné, že sa odosielaný rámec znehodnotí a zahodí.

Na obrázku 12 vidíme situáciu, kedy stanica B začala vysielať v čase t_0. Stanica D v čase t_1 sa rozhodla vysielať. Nakoľko k nej ešte signál zo stanice B neprišiel, tento signál ešte nezaregistrovala a začne tiež vysielať. Keď k niektorej stanici príde signál druhej stanice, nastáva kolízia, ktorá sa po čase odhalí a vysielanie sa ukončí.



Obrázok 12. Šírenie signálov zo staníc B a D a ich kolízia

Samotný algoritmus prístupovej metódy CSMA/CD funguje nasledovne (uvádzame pre prípad 100 megabitového Ethernetu):

1. Ak uzol má čo vysielať, a na spoji neregistruje žiadne vysielanie, začne vysielať svoj rámec.
2. Ak uzol má čo vysielať a na spoji registruje cudzie vysielanie, začne vysielať, až keď ho prestane registrovať.
3. Ak vysielanie celého rámca prebehlo bez toho, aby bolo počas vysielania registrované aj cudzie vysielanie, považuje sa rámec za úspešne odoslaný.
4. Ak počas vysielania uzol zistí, že vysiela aj iný uzol, ukončí vysielanie rámca a vyšle tzv. **signál JAM**. Signál JAM je signál takej intenzity, aby určite všetky uzly na spoji zaregistrovali, že došlo ku kolízii. Tento signál JAM sa vysiela po dobu 48 bitových intervalov. Bitový interval je čas, ktorý je potrebný na vyslanie jedného bitu v danom spoji. Napríklad, ak je prenosová rýchlosť 1 Mb/s, tak bitový interval je 1/1 000 000 sekundy teda 1 mikrosekunda.
5. Po prerušení a prijatí signálu JAM začnú všetky stanice, ktoré chcú **vysielať fázu súperenia**: Po *m*-tej kolízii v poradí si stanica vyberie náhodné číslo *K* z intervalu <0, min{2^m-1; 1023}>. Potom čaká *K**512 bitových intervalov a potom začne algoritmus CSMA/CD odznova (len s hodnotou *m* o 1 väčšou).

Fáza súperenia má za úlohu prispôsobiť čas opätovného odoslania rámca vzhľadom na aktuálne požiadavky na vysielanie v danom spoji distribuovaným spôsobom. Ak je potenciálnych vysielateľov málo, stačí im výber z malej množiny možností, aby si s veľkou pravdepodobnosťou vybrali iné hodnoty *K*. Ak je potenciálnych vysielateľov veľa, tak je potrebný väčší interval možností, aby si dvaja nevybrali rovnaké *K* a spôsobili tak ďalšiu kolíziu. Keďže to, koľko je potenciálnych vysielateľov jednotlivé uzly nevedia, tak sa používa tento spôsob postupného zväčšovania intervalu možných náhodných čísiel.

Spomínaný čas 512 bitových intervalov sa niekedy nazýva aj **slot time**. Predstavuje čas potrebný na vyslanie rámca minimálnej dĺžky. Minimálna dĺžka rámca je definovaná preto, aby sa dali definovať rozumné maximálne vzdialenosti dvoch uzlov na jednom spoji. Predstavme si, že chceme prevádzkovať prístupovú metódu CSMA/CD na spoji s rýchlosťou 100 Mbit/s. To znamená, že na odoslanie jedného bitu potrebujeme 10^{-8} sekúnd. Signál sa v medenom drôte šíri rýchlosťou približne 200 000 km/s t.j. $2*10^8$ m/s. Keď to dáme dokopy tak pokiaľ uzol vyšle 1 bit, signál prejde vzdialenosť 2 metre. Pri odoslaní posledného bitu minimálneho rámca (64 bajtov = 512 bitov) sa signál predstavujúci prvý bit tohto rámca už dostal do vzdialenosti 1024 metrov od vysielateľa. Teraz si predstavme situáciu, že uzol na druhom konci spoja začne vysielať tesne pred príchodom prvého bitu tohto rámca. Na to, aby náš prvý odosielateľ zistil, že jeho rámec bude v kolízii, musí sa o tejto kolízii dozvedieť ešte pred odoslaním posledného bitu svojho rámca. Z toho vyplýva, že by druhá stanica mala byť teoreticky vzdialená maximálne 512 metrov, aby aj signál od nej stihol prísť k prvej stanici pred tým ako prvá stanica odošle posledný bit svojho rámca.

Dôležitosť minimálnej dĺžky rámca môžeme ukázať aj na predchádzajúcom obrázku. Ak by bola vyznačená doba vysielania rámca veľmi krátka, kolíziu by stanica B počas svojho vysielania nemusela zaregistrovať a teda aj keby došlo ku kolízii, považovala by rámec za úspešne odoslaný.

Norma 100 megabitového Ethernetu *100Base-TX* hovorí o povolenej vzdialenosti iba 100 metrov (pre istotu, aby bolo odhalenie kolízie v každom hardvérovom prevedení naozaj úspešné) alebo 200 metrov v prípade použitia opakovača uprostred medzi stanicami. V prípade použitia optických vlákien, kde rýchlosť šírenia je skoro 300 000 km/s sa používa norma 100Base-FX, ktorá určuje maximálnu vzdialenosť uzlov na spoji na 412 metrov.

Pri spojoch s väčšou prenosovou rýchlosťou, t.j. 1 Gb/s a viac, je potrebné buď výrazne skrátiť maximálnu vzdialenosť, alebo zväčšiť minimálnu veľkosť rámca. Podľa normy 1000Base-T (t.j. v medenom drôte) je minimálna veľkosť rámca zhruba 8 krát väčšia, konkrétne 520 bajtov a maximálna vzdialenosť ostáva na 100 metroch bez opakovača a 200 s opakovačom. Pre optické vlákna podľa normy 1000-Base-SX je maximálna vzdialenosť až 550 metrov.

Pre väčšie vzdialenosti je už potrebné mať spojenia bez zdieľaného spoja. To znamená, že sa už nepoužíva prístupovú metódu CSMA/CD, ale nejaký protokol typu bod-bod (point-to-point) bez kolízií - takzvaný **plný duplex**. Plný duplex znamená, že si môžu oba uzly vysielať rámce navzájom v spoji súčasne bez vzniku kolízie. Fyzicky sa plný duplex realizuje tak, že buď stanice vysielajú na disjunktných frekvenčných pásmach alebo v rôznych fyzických spojoch (napr. dva rôzne drôty).

Prístupová metóda CSMA/CD sa snaží o to, aby sa podľa možností využívala plná prenosová rýchlosť spoja. Ak chce vysielať viac staníc, mali by si podeliť prístup k spoju tak, aby si čas vysielania podelili spravodlivo.

5.4.3. Rozbočovač (hub) a opakovač (repeater)

Rozbočovač slúži ako centrálne v hviezdicovej topológii. Funguje veľmi jednoducho. Ak cez niektorú zásuvku príde signál, tento signál je obvykle utlmený a skreslený. Rozbočovač tento signál regeneruje, odstráni vzniknuté skreslenia a vyšle ho do všetkých ostatných zásuviek, okrem tej, cez ktorú signál prišiel. Rozbočovač je zariadenie, ktoré prenášaným dátam nerozumie, nevie, kto je odosielateľ a kto príjemca. Preňho sú to iba jednotky a nuly. Ak dve ľubovoľné zariadenia napojené cez hub začnú naraz vysielať, spôsobia kolíziu. Hovoríme tomu, že sú v spoločnej **kolíznej doméne**. Rozbočovač je zariadenie, ktoré sa už v súčasnosti veľmi nevyužíva.

Opakovač je zariadenie podobné rozbočovaču, na ktoré je možné pripojiť iba dva spoje. Používa sa na regenerovanie signálu v dlhých spojoch, kde je bez opakovača útlm a skreslenie signálu výrazné.

5.4.4. Prepínač (switch) a most (bridge)

Prepínač je tiež zariadenie, ktoré má slúžiť ako centrálne zariadenie v hviezdicovej topológii. Rovnako ako rozbočovač sa stará o regeneráciu signálu. Prepínač však, na rozdiel od rozbočovača, už rozumie hlavičke rámca a teda pozná MAC adresu odosielateľa aj príjemcu rámca. To využíva tak, že ak vie, cez ktorý spoj je napojený príjemca rámca, pošle tento rámec iba do tohto spoja. Ostatné stanice v takomto prípade nedostanú cudzie rámce. To má hneď niekoľko výhod.

Prvou výhodou je „bezpečnosť" a odľahčenie spojov. Stanice, aj keď sú v jednej sieti, si navzájom nesledujú komunikáciu.

Bezpečnosť je len ilúziou, keďže existujú jednoduché a účinné útoky na prepínač alebo na lokálnu sieť, ktoré sú schopné tento princíp obísť alebo zablokovať napr. ARP poisoning. Útočník potom vie dostávať rámce od všetkých staníc napojených na prepínač.

Druhou výhodou je to, že stanice napojené na prepínač, už **nie sú v rovnakej kolíznej doméne**, teda, ak ľubovoľné dve stanice vysielajú v rovnakom čase, kolízia nevznikne. Na spoločnej kolíznej doméne sú už iba prepínač a daná stanica. Dokonca aj to už v súčasnosti neplatí, keďže káble Ethernetu umožňujú plný dupex (full duplex), t.j. stanica vysiela a prijíma nezávisle cez iné dvojice káblov. Signály z prepínača a zo stanice sa tak nikdy nestretnú. Prepínač je teda zariadenie, ktoré úplne odstraňuje kolízie v sieti a umožňuje tak efektívne využitie celej šírky pásma spojov a nezávislú komunikáciu zariadení. Dokonca, ak ide o prepínač typu „ulož a prepošli" (store-and-forward) (je ich väčšina), t.j. že nezačne vysielať rámec skôr, ako ho celý prijme, umožňuje aj prepojenie rôzne rýchlych spojov (napr. 100Mbit/s a 1Gb/s).

Prepínač, podobne ako rozbočovač, je **transparentné zariadenie**, to znamená, že o jeho existencii v sieti uzly nevedia. Uzly komunikujú s cieľovými uzlami tak, že zapíšu do hlavičky rámca cieľovú MAC adresu a rámec vyšlú do spoja vždy rovnako, bez ohľadu na to, či sú uzly napojené priamo alebo cez jeden či viac prepínačov. Prepínač podľa základnej charakteristiky ani nemá svoju MAC adresu. To však neplatí pre drahšie manažovateľné prepínače, ktoré poskytujú (obvykle webové) rozhranie na ich ďalšie nastavenie.

Predpokladom toho, aby rámce putovali iba k uzlu (stanici alebo smerovaču), ktorému sú určené, je to, že prepínač vie, cez ktorú zásuvku je napojený daný uzol. Túto informáciu má uloženú v **prepínacej tabuľke**. Konkrétne v niektorom zázname prepínacej tabuľky, kde má uložené 3 informácie: MAC adresu uzla, číslo zásuvky, cez ktorú je tento uzol dostupný a časovú pečiatku o tom, kedy bol tento záznam pridaný/aktualizovaný.

Záznamy, ktoré dlho neboli obnovené (neprišiel žiaden nový rámec z danej stanice), sú po čase z prepínacej tabuľky vymazané.

Prepínaciu tabuľku nie je potrebné ručne aktualizovať, pretože prepínač je **samoučiaci**, preto si dopĺňa a aktualizuje prepínaciu tabuľku sám cez nasledovný algoritmus:

Keď príde nový rámec na niektorú zo zásuviek prepínača:

1. Prečítaj si adresu odosielateľa a ulož si ju do prepínacej tabuľky spolu s číslom zásuvky, cez ktorú rámec prišiel. Ak záznam s danou adresou už existoval, iba mu aktualizuj časovú pečiatku.
2. Prečítaj si adresu príjemcu a skús ju nájsť v prepínacej tabuľke.
3. Ak si našiel záznam s touto adresou, potom
   - Ak je cieľ dostupný cez tú istú zásuvku ako je adresa odosielateľa zahoď rámec (lebo v danom spoji už cieľová stanica tento rámec prijala)
   - Inak pošli rámec príjemcovi cez správnu zásuvku
4. Ak si nenašiel záznam s touto adresou, potom pošli rámec do všetkých zásuviek okrem tej, cez ktorú tento rámec prišiel (chovaj sa ako rozbočovač)

Je potrebné poznamenať, že aj keď prepínač oddeľuje kolízne domény, nerozdeľuje **obežníkovú (broadcast) doménu**. Ak niekto odošle obežníkový rámec (s cieľovou adresou FF:FF:FF:FF:FF:FF), prepínač odosiela takýto rámec do všetkých zásuviek okrem tej, z ktorej rámec prišiel.

Most je zariadenie podobné opakovaču, ktoré naviac rozumie hlavičke rámca

a rovnako ako prepínač si pamätá, cez ktorú z jeho dvoch zásuviek je prístupná stanica s akou MAC adresou. Rámce určené pre stanicu dostupnú cez rovnakú zásuvku ako odosielateľ sú zahodené.

5.5. Bezdrôtové siete – IEEE 802.11 (WiFi)

Nástup bezdrôtových spojení je taký veľký, že počet bezdrôtových spojení už presiahol počet spojení cez drôtové spojenia. Bezdrôtovo sa už nepripájame do internetu iba počítačmi, ale aj mobilnými telefónmi, či rôznymi zabezpečovacími systémami.

Bezdrôtové siete musia riešiť dva hlavné problémy:

- **bezdrôtové spojenie**, t.j. schopnosť komunikácie dvoch a viacerých zariadení "vzduchom"
- **mobilita spojenia**, t.j. schopnosť stanice viac-menej kontinuálne komunikovať aj pri zmene miesta cez ktoré sa pripája (napr. prechode medzi BTS anténami mobilných operátorov)

Pri realizácii bezdrôtového spojenia musí tvorca protokolu zohľadniť niektoré špecifiká bezdrôtového spojenia ako sú:

- **Zoslabovanie signálu** – Signál ktorý prijímame je oveľa slabší ako signál, ktorý vysielame.
- **Zmena frekvencie signálu** – Prejavuje sa pri odraze od objektov v priestore a pri vzájomnom pohybe vysielača a prijímača.
- **Interferencie z iných zdrojov žiarenia** – Vo voľnom prostredí nie je signál chránený pred interferenciami z okolia tak ako v kábloch. Interferencie môžu spôsobovať iné zariadenia vysielajúce na rovnakej frekvencii, ale aj na prvý pohľad nevinné zdroje žiarenia ako sú mikrovlnné rúry, či elektromotory.
- **Problém skrytej stanice** (Hidden terminal problem) a **zoslabovanie signálu** spôsobujú to, že ak na jednom prenosovom pásme komunikuje viac zariadení, môže sa stať, že dve zariadenia si vzájomne rušia komunikáciu s tretím zariadením aj keď o tom nevedia, lebo sa kvôli prekážke medzi nimi alebo veľkej vzájomnej vzdialenosti nepočujú, aj keď sa obe s tretím zariadením niekde „medzi nimi“ počuť môžu.

Technológií a štandardov pri bezdrôtovej komunikácii je veľké množstvo. Líšia sa od seba hlavne prenosovými rýchlosťami, maximálnou vzdialenosťou komunikácie dvoch antén aj spôsobom napojenia a vysielania. My sa bližšie pozrieme na technológiu WLAN sietí a štandardy 802.11 verzie b,g,a,n (WiFi). Ďalšie známe technológie sú Bluetooth (z ktorého vznikol štandard 802.15), WiMAX (štandard 802.16) alebo technológie využívané u mobilných telefónnych operátorov: GSM, GPRS, EDGE a UMTS.

Štandard 802.11 definuje bezdrôtovú LAN sieť (WLAN), ktorá sa zo všetkých bezdrôtových prenosových technológií najviac podobá Ethernetu. Niekedy sa označuje ako bezdrôtový ethernet.

Štandard 802.11 počas svojho vývoja prešiel niekoľkými verziami, ktoré sa líšia prenosovými rýchlosťami a frekvenčnými pásmami na ktorých vysielajú.

- verzia **802.11b** pracuje na frekvenčných pásmach v rozsahu 2,4-2,485 GHz a umožňuje prenosové rýchlosti do 11 Mbit/s
- verzie **802.11g** a **802.11a** umožňujú rovnaké prenosové rýchlosti do 54 Mbit/s, ale líšia sa frekvenčnými pásmami. 802.11g pracuje na frekvenciách 2,4-2,485 GHz a 802.11a na frekvenciách 5-6GHz.
- najnovšia verzia **802.11n** ktorá sa štandardizovala v októbri 2009 pracuje tiež na frekvenčných pásmach v rozsahu 2,4-2,485 GHz, ale umožňuje prenosové rýchlosti až do 300 Mbit/s.

Frekvenčné pásma sú delené na niekoľko kanálov šírky 20 až 40MHz. Niektoré kanály sa navzájom prekrývajú. Signály z prekrývajúcich sa pásiem sa môže navzájom ovplyvňovať a teda aj znehodnocovať. Tak napríklad 802.11g má 13 kanálov šírky 20 MHz (v Amerike len 11 kanálov), ale ak chceme vybrať iba neprekrývajúce sa, tak okrem krajných (prvého a trinásteho) si už môžeme vybrať iba jeden "stredný" – šiesty, siedmy alebo ôsmy. Ak sú v blízkosti aj iné prístupové body bezdrôtových sietí, tak je vhodné, aby sa nastavili kanály koordinovane tak, aby sa navzájom neprekrývali a nespôsobovali interferencie a teda aj spomalenie prenosových rýchlostí.

Ak sa chceme napojiť na prístupový bod (access point), musíme poznať jeho meno (SSID - Service Set Identifier) a MAC adresu. Celý postup sa dá zjednodušene opísať nasledovne. Prístupové body vysielajú v istých časových intervaloch (typicky 1 sekunda) takzvaný signálny rámec, v ktorom vysielajú svoje SSID a MAC adresu. Stanica si tiež môže požiadať o okamžité zaslanie signálnych rámcov vyslaním požiadavky hľadania prístupových bodov. Stanica potom ladí cez jednotlivé kanály a odchytáva tieto signálne rámce. Keď si vyberie prístupový bod, na ktorý sa chce napojiť, pošle rámec požiadavky na napojenie na tento prístupový bod. Typicky nasleduje autentifikácia cez WEP, WPA alebo WPA2. Po úspešnej autentifikácii je už bezdrôtové spojenie vytvorené a je možné začať komunikovať (napríklad opýtať si IP adresu cez DHCP).

WLAN umožňuje napojenie cez prístupový bod (access point) aj napojenie ad-hoc. Pri komunikácii cez prístupový bod susedné zariadenia komunikujú vždy cez prístupový bod, nikdy nie priamo medzi sebou. Pri napojení ad-hoc komunikujú zariadenia medzi sebou priamo, alebo sprostredkovane cez iné stanice v prípade, že sa komunikujúce stanice nevidia (napr. pre problém skrytej stanice).

Vo WLAN sieťach sa namiesto prístupovej metódy CSMA/CD (collision detection = detekcia kolízií) používa prístupová metóda CSMA/CA (collision avoidance = vyhýbanie sa kolíziám). Detekcia kolízií, teda zistenie vysielania iného zariadenia počas vlastného vysielania, ktorá úspešne funguje v káblových sieťach sa pri bezdrôtových spojeniach nedá použiť. Dôvodom je zoslabovanie signálu. Keďže práve vysielajúca stanica vysiela signál oveľa väčšej intenzity, ako je intenzita prijímaného signálu vysielaného inými stanicami, nie je možné tento slabý signál počas vlastného vysielania odhaliť. Inými slovami keď stanica vysiela, je hluchá.

To, čo majú obe prístupové metódy spoločné, je počúvanie okolia pred začiatkom vlastného vysielania. Pokiaľ stanica registruje iné vysielanie, nezačne vysielať. Žiaľ aj tento princíp nie je v prípade bezdrôtových spojení až taký účinný ako v káblových spojeniach. Je tu totiž známy problém skrytej stanice a zoslabovanie signálu, ktoré môžu zapríčiniť, že stanica neregistruje vysielanie inej stanice, lebo aj keď je signál tohto vysielania dostatočne silný pre prístupový bod, nemusí byť registrovateľný pre stanicu za prekážkou alebo z opačnej strany prístupového bodu ako vysielajúca stanica.

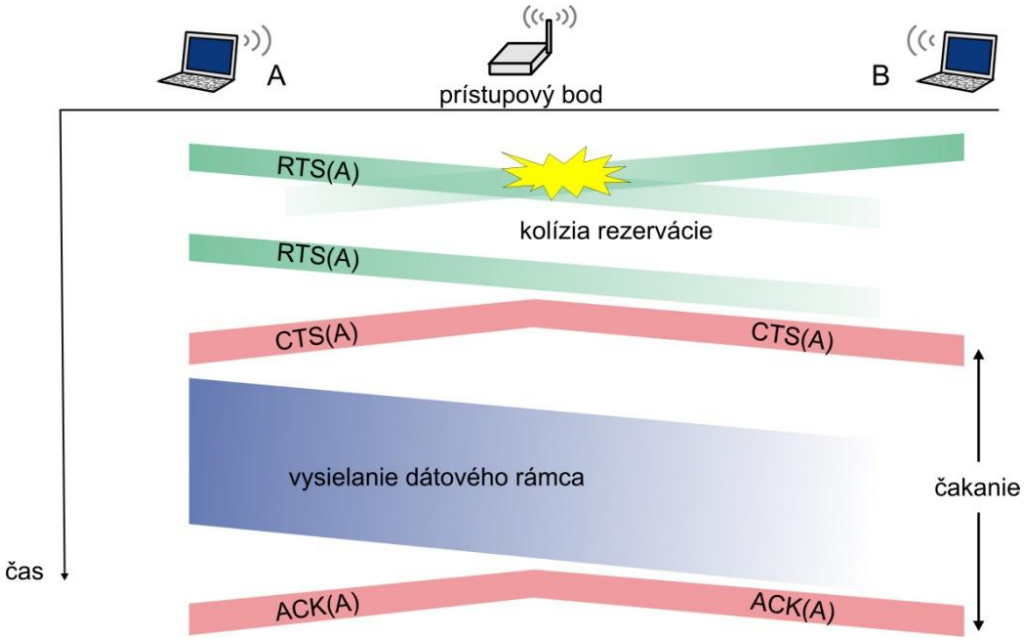
Hlavnou novinkou v prístupovej metóde CSMA/CA je potvrdzovanie odoslaných rámcov cez potvrdzovacie (ACK - acknowledgement) rámce. Na rozdiel od TCP protokolu je potvrdenie realizované po každom rámci (v CSMA/CA nepoužívame okná odosielateľa a príjemcu). Potvrdzovanie je nutné kvôli zvýšenému riziku interferencií, ale hlavne kvôli neschopnosti detekcie kolízií. V ethernete stanica po odoslaní rámca bez odhalenia kolízie pokladá rámec za úspešne odoslaný. Pri bezdrôtovom spojení si musí počkať na potvrdenie.

Rámec pre WLAN má v hlavičke oveľa viac záznamov ako rámec ethernetu, aby sa dali používať potvrdenia, ale aj iné špeciálne informácie. Zaujímavosťou je aj to, že obsahuje až 4 MAC adresy – okrem pôvodného odosielateľa a konečného príjemcu môže byť signál prenášaný aj inými bezdrôtovými sprostredkovateľmi na ceste, ktorí majú vlastné MAC adresy.

Rozšírenie tohto algoritmu o "/CA", teda o collision avoidance = vyhýbanie sa kolíziám, sa snaží eliminovať vznikanie kolízií na maximálne možnú mieru a to hlavne pre zariadenia, ktoré sa navzájom nepočujú. Idea tohto rozšírenia je taká, že prístupový bod, ktorý počujú všetky stanice, vyšle CTS (clear-to-send) rámec všetkým staniciam s informáciou o tom, ktorá stanica bude v najbližšom čase vysielať. Ostatné stanice sú v tom čase ticho a tak počas tohto vysielania nevznikne kolízia ani so zariadeniami, ktoré toto vysielanie nepočujú.

Ak si chce nejaká stanica rezervovať čas na svoje vysielanie, pošle RTS (request-to-send) rámec. Prístupový bod potom pre túto stanicu rezervuje vysielací čas cez CTS rámec. Samozrejme sa úplne všetky kolízie týmto neodstránia. Stále môže dochádzať ku kolíziám RTS rámcov viacerých staníc, alebo ku kolízii RTS a CTS rámcov. Tieto rámce sú však malé a tak kolízie netrvajú tak dlho ako by to bolo v prípade oveľa väčších dátových rámcov.

Na obrázku 15 môžeme vidieť situáciu, že stanica A aj stanica B poslali žiadosť o rezerváciu vysielacieho času (RTS). Tieto signály sa dostali do kolízie a tak ich prístupový bod nedokázal prijať. Keďže staniciam A a B neprišla žiadna odpoveď v podobe CTS rámca, každá z nich si zvolila interval čakania, po ktorom začne vysielať opäť. Keďže stanica A si vygenerovala kratší interval, vyšle RTS rámec skôr. Prístupový bod vyhradí vysielací čas stanici A odoslaním CTS rámca. Počas vyhradeného času môže stanica A pokojne vysielať, lebo ostatné stanice musia čakať a nespôsobia tak kolíziu. Po prijatí celého dátového rámca prístupovým bodom a úspešného overenia CRC kontroly pošle prístupový bod potvrdzovací rámec o úspešnom doručení. Od tejto chvíle môžu všetky stanice opäť žiadať o vysielací čas pomocou RTS rámcov.



Obrázok 15. Príklad rezervácie a vyhradeného času na odosielanie dátového rámca

5.6. Preklad z IP adries na MAC adresy: protokol ARP

Protokol ARP je protokolom sieťovej vrstvy. Uvádzame ho však až na tomto mieste, keď už poznáme problematiku adresácie cez MAC adresy na vrstve sieťového rozhrania.

Keďže máme dvojitú adresáciu (IP a MAC adresy), potrebujeme mať spôsob, ako sa pri znalosti jednej adresy, vieme dopracovať k druhej adrese. Znalosť oboch adries je potrebná na komunikáciu, ktorá využíva sieťovú vrstvu, to znamená minimálne pri komunikácii medzi sieťami, ale napríklad aj na ľubovoľnú komunikáciu v rámci siete, ktorá využíva protokol IP a jeho nadstavby TCP a UDP. Ak sieťovú vrstvu v rámci siete nepoužívame, stačí nám adresácia cez MAC adresy.

Protokol ARP umožňuje preklad z IP adries na MAC adresy. Ten je potrebný v prípade, ak MAC adresu cieľového uzla v našej sieti nepoznáme. Tento stav je dosť bežný, keďže takmer všetky sieťové aplikácie používajú adresáciu iba cez doménové mená alebo IP adresy. Bez MAC adresy by sme neboli schopní vytvoriť hlavičku rámca.

Pred tým, ako si povieme o tom ako funguje ARP protokol, musíme spomenúť **ARP tabuľku**. ARP tabuľku obsahuje každé rozhranie smerovača aj sieťových adaptérov staníc. V ARP tabuľke sa uchovávajú dvojice IP adresa a MAC adresa, ktoré boli v poslednom období zistené. ARP tabuľka nemusí obsahovať preklady adries všetkých uzlov v sieti. Jednotlivé dvojice adries sa z ARP tabuľky môžu dokonca zmazať,

pokiaľ od daného uzla neprišiel dlho žiaden rámec. Čas, za ktorý sa záznam v ARP tabuľke maže, závisí od implementácie v danom operačnom systéme, ale býva to 5 až 20 minút.

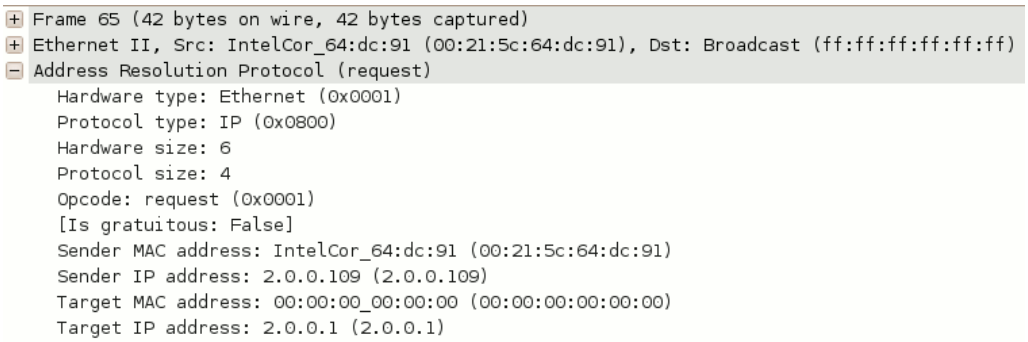
Datagram, ktorý potrebujeme vložiť do nového rámca obsahuje IP adresu príjemcu a odosielateľa. Do tohto nového rámca musíme zapísať MAC adresu príjemcu a odosielateľa. S adresou odosielateľa je to ľahké, keďže ide o tú istú stanicu. MAC adresu príjemcu je potrebné odvodiť od IP adresy príjemcu.

Východiskom na zistenie cieľovej MAC adresy je IP adresa nejakého rozhrania v našej sieti. Nakoľko konečný príjemca nemusí byť v tej istej sieti, musíme sa najprv pozrieť do smerovacej tabuľky, ktorá sa uchováva v každom uzle. Ak cieľová IP adresa nie je z rovnakej siete, musíme si zo smerovacej tabuľky zistiť bránu, cez ktorú ideme posielať náš datagram. Táto brána už musí byť z našej siete.

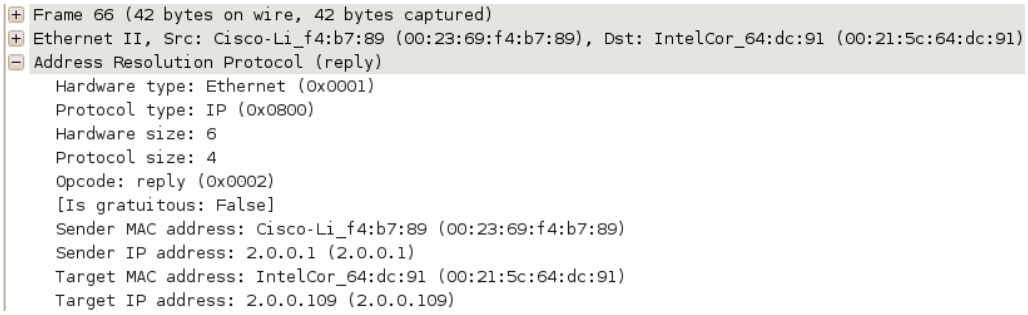
Teraz už máme IP adresu cieľa v lokálnej sieti, potrebujeme zistiť preklad tejto IP adresy na MAC adresu. Najprv sa pozrieme do ARP tabuľky. Ak sa v nej preklad tejto IP adresy nachádza, použijeme ho a vytvoríme rámec. Ak sa v ARP tabuľke táto IP adresa nenachádza, musíme ju zistiť v našej sieti cez ARP protokol:

Stanica vyšle ARP požiadavku všetkým zariadeniam v sieti na obežníkovú MAC adresu FF:FF:FF:FF:FF:FF. Ako MAC adresu odosielateľa zadá svoju MAC adresu. Tento rámec musí byť spracovaný všetkými zariadeniami v sieti. To zariadenie, ktoré zistí, že IP adresa je jeho, pošle ARP odpoveď tejto hľadajúcej stanici. V tejto ARP odpovedi zapíše hľadaný preklad adries. Nakoľko už vie MAC adresu hľadajúcej stanice, pošle jej už ARP odpoveď priamo, teda nie obežníkom=všetkým staniciam. Keď naša stanica dostane túto odpoveď, doplní si záznam do svojej ARP tabuľky a môže vygenerovať hlavičku rámca pre pôvodný datagram.

Príklad takejto komunikácie zobrazujú nasledujúce dva zábery na ARP pakety zachytené v programe Wireshark. ARP požiadavku posiela stanica s IP adresou 2.0.0.109 a s MAC adresou 00:21:5c:64:dc:91 všetkým staniciam v danej sieti na obežníkovú (broadcast) MAC adresu ff:ff:ff:ff:ff:ff, aby zistila, akú MAC adresu má stanica s IP adresou 2.0.0.1. ARP odpoveď posiela stanica s IP adresou 2.0.0.1 už priamo stanici, ktorá vysielala požiadavku, teda stanici s IP adresou 2.0.0.109 a MAC adresou 00:21:5c:64:dc:91. Z tela ARP odpovede vieme ľahko prečítať, že MAC adresa zdrojovej stanice s IP adresou 2.0.0.1 je 00:23:69:f4:b7:89.



Obrázok 13. Príklad ARP požiadavky



Obrázok 14. Príklad ARP odpovede

Rovnaký postup sa aplikuje na každom uzli na ceste k cieľovému príjemcovi, v každej sieti, cez ktorú datagram prechádza. Konkrétne každý smerovač, ktorý prijme rámec cez niektorý svoj port (zásuvku), získa z tohto rámca prenesený IP datagram. Následne podľa svojej smerovacej tabuľky zistí, cez ktoré rozhranie má tento datagram odoslať a zistí si zo smerovacej tabuľky IP adresu nasledujúceho zariadenia na ceste k cieľovej stanici. Ak má túto IP adresu v svojej ARP tabuľke, zabalí datagram do nového rámca s príslušnou cieľovou MAC adresou nasledujúceho zariadenia a odošle tento rámec cez daný port. Ak túto IP adresu nemá v ARP tabuľke, zistí si MAC adresu zariadenia cez ARP protokol.

Aktivita

Zobrazte si vašu ARP tabuľku príkazom **arp -a**. Zapnite si odchytávanie prevádzky v programe Wireshark. Môžete si v ňom nechať vypisovať iba ARP pakety napísaním slova arp do poľa filter.

Ak ste vo Windowse, môžete si aj dočasne vymazať ARP tabuľku príkazom **„arp -d“** (musíte príkazový riadok pustiť ako administrátor) a následne ísť na nejakú webovú stránku, čím si vynútite, aby si váš počítač vyžiadal MAC adresu vašej prednastavenej brány. Príslušné ARP pakety preskúmajte v programe Wireshark.

Na zmazanie ARP tabuľky v Linuxe môžete použiť príkaz:
sudo ip neigh flush all

5.7. Preklad z MAC adries na IP adresy: protokol RARP

Preklad opačným smerom, teda z MAC adries na IP adresy umožňuje protokol sieťovej vrstvy RARP (Reverse ARP). V RARP požiadavke zasiela stanica svoju MAC adresu so žiadosťou o pridelenie IP adresy. Toto riešenie je už považované za zastaralé, nakoľko túto funkcionalitu v plnej miere ponúka protokol DHCP, ktorý naviac poskytuje aj ďalšie informácie ako predvolenú bránu, adresy lokálnych DNS serverov a podobne.

Čo sme sa naučili v tomto module

Zhrnutie

V tomto module sme sa oboznámili s vrstvovým referenčným modelom ISO/OSI. A postupne sme sa na každej vrstve zoznámili s kľúčovými protokolmi ktoré riešia základné úlohy danej vrstvy.

Na aplikačnej vrstve sme si zopakovali protokoly HTTP, SMTP a systém DNS. Spoznali sme protokol DHCP na dynamické prideľovanie IP adries.

Na transportnej vrstve sme sa oboznámili s odľahčeným protokolom UDP a spojovaným a potvrdzovaným protokolom TCP a jeho metódou posuvného okna, ktorá umožňuje prenos dát všetkých dát v správnom poradí. Vysvetlili sme si, čo sú to porty a ako sa používajú na identifikáciu zdrojového a cieľového procesu.

Na sieťovej vrstve sme sa zoznámili s protokolmi IPv4 a IPv6 a adresáciou, ktorú využívajú. Povedali sme si načo nám slúžia smerovacie protokoly. Okrem toho sme si predstavili a využitie NAT smerovača na vytvorenie privátnej siete. Predstavili sme si protokol ICMP využívaný v programoch ping a tracert na diagnostiku siete. Spoznali sme aj protokoly ARP a RARP.

Na vrstve sieťového rozhrania sme si predstavili prenosové médiá a vhodnosť použitia digitálneho a analógového vysielania. Spoznali sme topológie počítačových sietí a zariadenia rozbočovač, opakovač, prepínač a most. Zamerali sme sa na prenosové technológie ethernet a WLAN (WiFi). Predstavili sme si prístupové metódy CSMA/CD a CSMA/CA na komunikáciu v týchto sieťach.

Preverenie výstupných vedomostí

Zistite doma alebo v práci, aký rozsah IP adries spravuje váš poskytovateľ internetového pripojenia. Poskytuje Vám verejnú IP adresu? Koľko smerovačov je na ceste od vášho počítača k serveru dvui.ccv.upjs.sk? Vyrobte snímky obrazoviek, ktoré ukazujú vaše riešenie a zašlite ich ako riešenie zadania do systému Moodle.

Literatúra a použité zdroje

[1] James F. Kurose, Keith W. Ross: Computer Networking: A Top-Down Approach, 4th edition. Pearson Education, Inc., ISBN: 0-321-51325-8, 878 pages, 2008

[2] Libor Dostálek, Alena Kabelová: Velký průvodce protokoly TCP/IP a systémem DNS, 5. aktualizované vydání, Computer Press, Praha, ISBN: 978-80-251-2236-5, 2008

[3] Stephen J.Bigelow: Mistrovství v počítačových sítích. Správa, konfigurace, diagnostika a řešení problémů. Computer Press, Brno, ISBN: 80-251-0178-9, 2004

[4] Peter Tomcsányi, Miloš Šrámek, Peter Palúch: Operačné systémy a počítačové siete. Študijný materiál projektu ĎVUI, ISBN 978-80-8118-031-6

Tento študijný materiál vznikol ako súčasť národného projektu Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika v rámci Aktivity „Vzdelávanie nekvalifikovaných učiteľov informatiky na 2. stupni ZŠ a na SŠ“.

Autori © RNDr. Peter Gurský, PhD.

Názov Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika

Podnázov Počítačové siete

Študijný materiál prešiel recenzným pokračovaním.

Recenzenti doc. Ing. Matilda Drozdová, CSc.

doc. Ing. Ľudovít Trajteľ, PhD.

Počet strán 40

Náklad 300 ks

Prvé vydanie, Bratislava 2010

Vydal Štátny pedagogický ústav, Pluhová 8, 830 00 Bratislava, v súčinnosti s Univerzitou Pavla Jozefa Šafárika v Košiciach, Univerzitou Komenského v Bratislave, Univerzitou Konštantína Filozofa v Nitre, Univerzitou Mateja Bela v Banskej Bystrici a Žilinskou univerzitou v Žiline

Vytlačil BRATIA SABOVCI, s r.o., Zvolen

ISBN 978-80-8118-069-9