

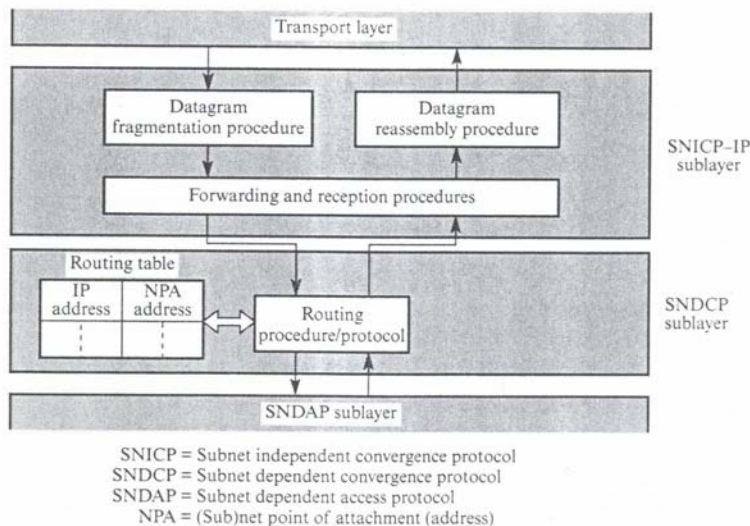
Smerovanie a prepájanie v sieťach

Dátové siete zabezpečujú prenos dát od zdroja k cieľu. Aby mohol takýto prenos fungovať, musia byť zavedené **mená a adresy**. Každému koncovému bodu je priradená jednoznačná identifikácia – **meno** a určenie, kde sa v danej sieti nachádza – **adresa**. Na základe mena možno nájsť, napríklad pomocou zoznamu príslušnú adresu (napr. telefónny zoznam). Na základe adresy potom možno poslať dáta od zdroja k cieľu. Pri prechode dát uzlami siete hovoríme o procese **prepájania (relaying)**. Je to proces prijatia dát a na základe smerovacieho rozhodovacieho procesu následné odoslanie ďalšiemu uzlu smerom k cieľu.

Ďalším procesom je **smerovanie (routing)**. Je to proces, ktorý prebieha na uzloch vo vnútri siete. Každý uzol sa musí na základe informácií (hlavne cieľovej adresy) v prenášaných dátach rozhodnúť, ktorému ďalšiemu uzlu tieto dáta poslať, aby sa nakoniec dostali do cieľa. Uzol musí mať znalosti o topológii siete, musí vedieť cez ktorého suseda (susedia sú všetci, s ktorými má spojenie) je možné dané dáta doručiť do cieľa čo najlepšie. Výsledkom procesu smerovania je zostavenie tzv. **smerovacej tabuľky**, v ktorej je napríklad napísané, že do cieľov A,B,D a F sa dostanem cez suseda X, a do cieľov M,N,P sa dostanem cez suseda Y.

Mená a adresy môžu byť jednotlivým staniciam pridelované náhodne, alebo s určitou logikou. Napríklad adresy môžu byť pridelované hierarchicky (telefónne čísla), alebo pridelovaním celých rozsahov adries koncovým zariadeniam pripojených na jeden uzol siete (IP priestory v TCP/IP sieťach). Vhodné priradenie adries môže do veľkej miery uľahčiť **smerovací proces**. Na smerovačoch si potom stačí napríklad pamätať, že všetky adresy začínajúce číslom 124 smerujem na suseda X, a všetky adresy začínajúce číslom 125 smerujem na suseda Y.

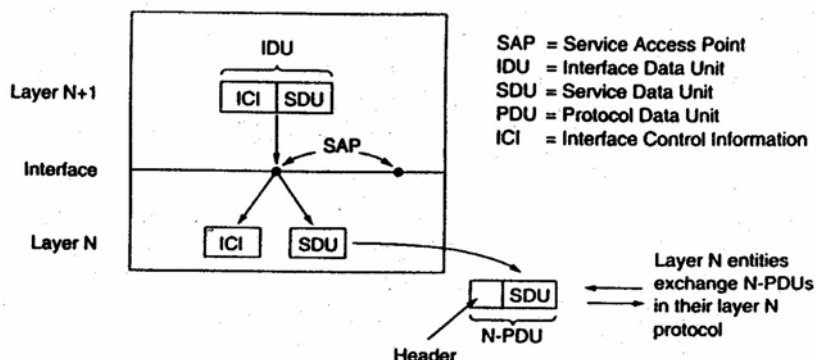
základné úlohy sieťovej vrstvy



adresovanie v sieťových vrstvách

každá vrstva potrebuje mechanizmus na určenie zdroja a cieľa – teda adresovanie

- pre každú vrstvu je definovaný tzv. Service access point
 - cez SAP sú sprístupňované služby
 - každý SAP predstavuje unikátnu adresu



TCP/IP a RM OSI

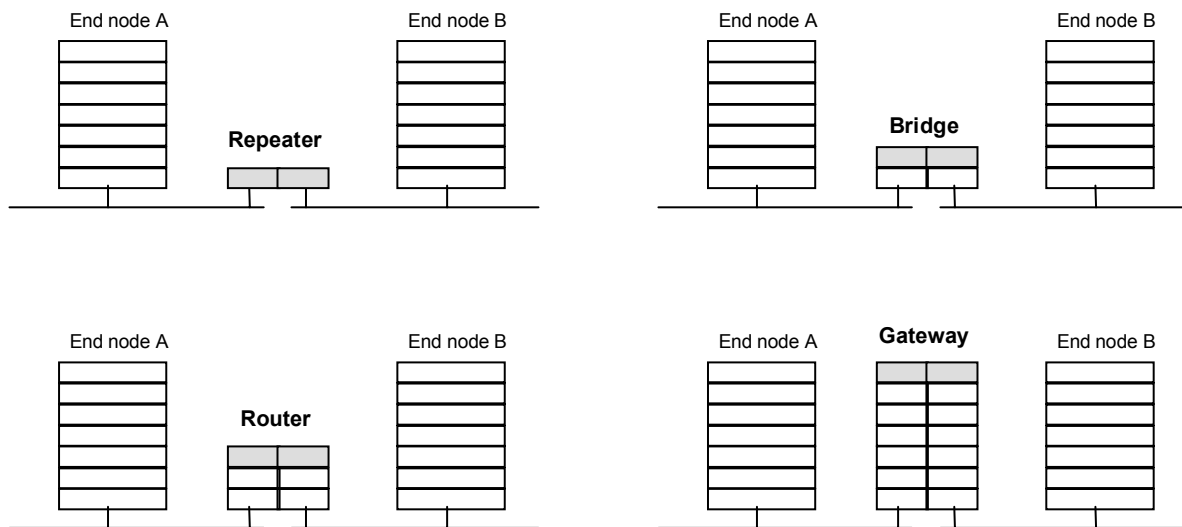
V súčasnosti je najrozšírenejšou sústavou komunikačných protokolov rodina protokolov TCP/IP. Predstavuje de-fakto štandardy pre dátovú sieťovú komunikáciu. Tieto protokoly boli najskôr vytvorené, sú široko používané a až neskôr sú postupne štandardizované. Na druhej strane existujú aj protokoly, ktoré boli najprv štandardizované a až následne bola snaha o ich zavedenie do praxe. To je prípad protokolov OSI. V každom prípade je vhodné používať to najlepšie z oboch skupín. Z OSI sa používa referenčný model (RM OSI) na popis funkcionality a z TCP/IP sú použité samotné protokoly. Aj v nasledujúcom budeme hovoriť o Protokoloch TCP/IP a Referenčnom modeli RM OSI.

Referenčný model RM OSI je vhodný na popis architektúry sietí a funkcionality. Veľmi presne definuje vrstvy, služby, interfejsy a protokoly. Popisuje 7 vrstiev podľa SNA (aby nebol štandard IBM – 5vrstvový), výsledkom čoho je fakt, že vrstvy 2 a 3 (Linková a Sieťová) sú preplnené, vrstvy 5 a 6 (Relačná a Prezentačná) skoro nič nerobia. V každej vrstve hovorí o error-control a má silný klasický „**komunikačne orientovaný**“ prístup, nie počítačovo orientovaný, pretože podporuje hlavne prepájanie okruhov (connection oriented) a nie prepájanie paketov (connectionless). Vynechané sú Data security, encryption ako aj network management. Samotné servisné primitívy sú interrupt oriented, čo je nevhodné pre vyššie jazyky.

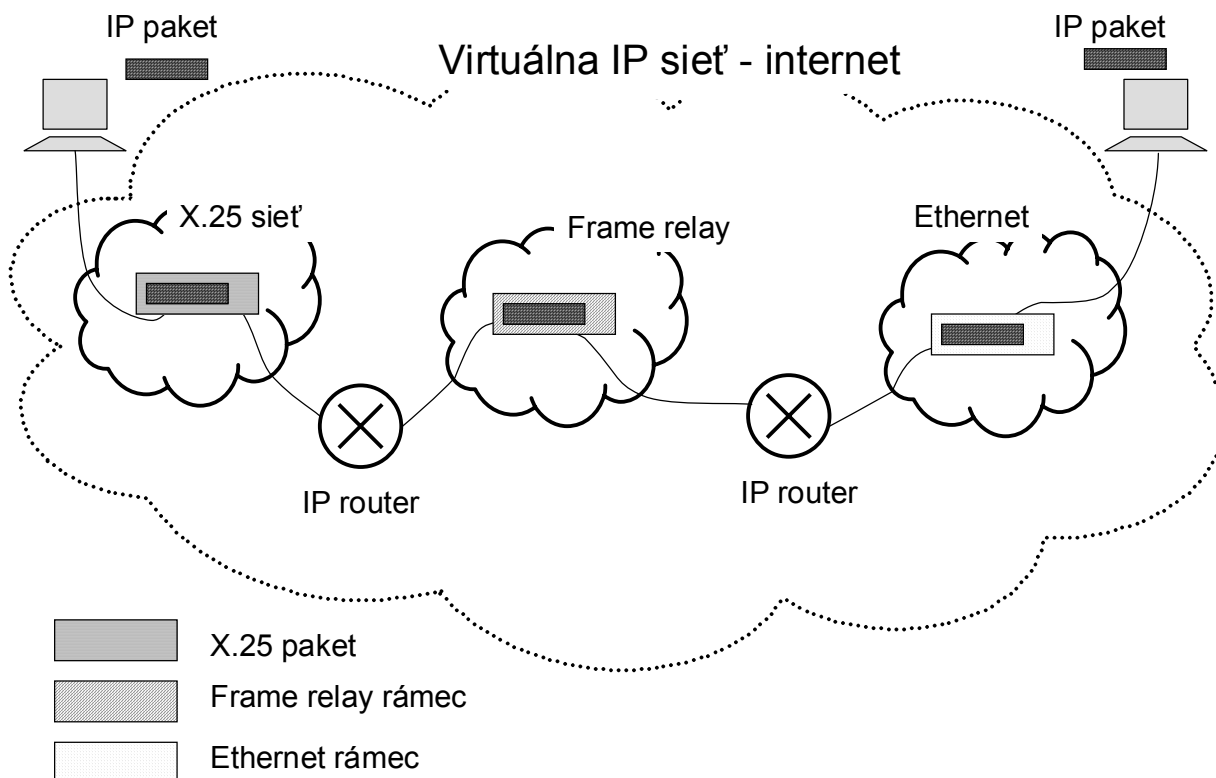
Protokoly TCP/IP sú široko používané, sú funkčné a podporované takmer všetkými výrobcami. Vznikli počas budovania výskumnej siete ARPANET, vybudovanej v rámci projektu sponzorovaného DoD (ministerstvo obrany USA) počas studenej vojny. Hlavnou požiadavkou bolo vybudovať komunikačnú sieť, ktorá bude odolná aj voči jadrovej vojne. Špecifikácie protokolov sú vydávané vo forme tzv. **RFC (Request For Comment)**, ktoré sú široko dostupné na Internete. **Internet** (s veľkým počiatočným písmenom **I**) je konkrétna celosvetová sieť, ktorá vznikla pospájaním rôznych druhov fyzických sietí pomocou protokolov rodiny TCP/IP. Vo všeobecnosti existujú aj iné **internety**, teda virtuálne siete zložené z rôznych sietí. Pokiaľ sú poprepájané pomocou protokolov TCP/IP, používame pre ne označenie **TCP/IP internet**.

Internetworking

Vo všeobecnosti možno dátové siete prepájať na rôznych vrstvách. Podľa toho je použité príslušné zariadenie, opakovač (repeater), bridge (most), smerovač (router) a brána (gateway).



Najväčším prínosom protokolov TCP/IP je fakt, že umožňujú prepájanie rôznych druhov sietí. Základom je prepájanie na tretej vrstve, realizované pomocou IP (Internet Protocol) protokolu a zariadení *router*. IP vrstva je spoločná pre všetky siete. Z pohľadu vyšších vrstiev sa potom všetky siete tvária ako rovnocenné. Virtuálnej sieti, ktorá vznikne prepojením viacerých fyzických sietí pomocou IP protokolu hovoríme **IP internet**. Problematika **Internetworkingu (poprepájania rôznorodých sietí)** si vyžaduje vyriešiť hlavne zachovanie formátu rámcov jednotlivých sietí a zachovanie pôvodných adresných schém. Najjednoduchšie je to riešiteľné pridaním nadradenej vrstvy, ktorá zavedie **univerzálny formát dát** (do ktorého je možné zabaliť – enkapsulovať pôvodné rámce) a **univerzálnu adresnú schému**, na ktorú je možné namapovať pôvodné číslovania. Preklady medzi formátmi a adresami pri IP vykonávajú routery. Nasledujúci obrázok znázorňuje takto poprepájané rôzne fyzické siete a enkapsuláciu IP paketu do fyzických rámcov jednotlivých sietí.



Obrázok – Internetworking, prepojenie rôznych druhov sietí pomocou IP

IP - Internet Protocol

Základom celej rodiny protokolov TCP/IP je protokol sieťovej (tretej) vrstvy – IP protokol (**I**nternet **P**rotocol). Bol definovaný ako RFC 791 v roku 1981. Ide o verziu číslo 4 protokolu IP označovanú ako **IPv4**.

Základnou požiadavkou na sieťovú IP vrstvu je vytvoriť sieť s prepájaním paketov založenú na spoločnej sieťovej vrstve bez spojovej orientácie. Jej úlohou je vkladať pakety do ľubovoľnej siete a každý paket nechať **samostatne** putovať do jeho cieľa. IP poskytuje vyšším vrstvám **nespol'ahlivú**, „best effort“ datagramovú službu. Táto úloha je veľmi podobná pošte a funguje tak, že IP vrstva rozseká veľkú správu na menšie časti a tie zabalí do obálok (paketov). Následne tieto obálky (pakety) označí cieľovou adresou a pošle ich smerom k cieľu (každú samostatne, presne ako poštové zásielky). Preto tieto obálky (pakety) môžu ísť rôznymi cestami, môžu prísť rôzne neskoro, môžu prísť v rôznom poradí a navyše nemusia do cieľa vôbec dôjsť.

Je potrebné, aby sa všetky druhy sietí prepájaných na 3. vrstve pomocou IP protokolu tvárili rovnocenne, resp. aby boli navzájom dosiahnuteľné. Preto je zavedené **jednotné adresovanie** použitím unikátnych adries, nezávislých od typu adries konkrétnej fyzickej siete. Toto univerzálne adresovanie je v IP protokole realizované pomocou tzv. **IP adries**. Následne je potrebné zabezpečiť previazanie adries konkrétnej fyzickej siete s týmito univerzálnymi (IP) adresami.

IP adresovanie

V protokole IPv4 sú definované tzv. IP adresy. IP adresa je 32-bitové číslo (v dvojkovej sústave zapisateľné pomocou 32 jednotiek a núl). Môže nadobúdať hodnoty od 0 do 2^{32} (4 294 967 295). K dispozícii je preto cca 4 miliardy adries.

Formáty zápisu IP adries

Pre počítačové spracovanie je vhodný **binárny zápis**. Ide o zápis v binárnej sústave, t.j. pomocou núl a jednotiek. Adresa potom predstavuje postupnosť 32 núl a jednotiek, ktorú je kvôli prehľadnosti zvykom zapisovať po osmiaciach.

Príklad: 11000000 10101000 01111011 00001100

Pretože binárny zápis je pre ľudí relatívne dlhý a neprehľadný, pri bežnej komunikácii sa používa zápis vo forme štyroch desiatkových čísel oddelených bodkou, tzv. **Dotted decimal notation**.

Príklad: 192.168.123.12, čo je ekvivalent adresy 11000000 10101000 01111011 00001100

Každá stanica pripojená do IP siete, každý router a každé zariadenie má pridelenú IP adresu. Ešte presnejšie, spravidla každý interfejs, cez ktorý je IP zariadenie pripojené do IP siete má pridelenú IP adresu. Zariadenie pripojené cez viacero interfejsov má potom pridelených viacero adries (multihomed hosts alebo routre), dokonca je možné pridelit' viacero IP adries aj na jeden fyzický interfejs.

Význam IP adries

Protokol IP je v súčasnosti používaný hlavne na prepájanie rôznych sietí, pričom nemusia byť rovnakého typu, rozsiahlosti, počtu staníc atď. Principiálne by bolo možné adresy pridel'ovať jednotlivým zariadeniam postupne od čísla 0, 1, 2 ... až po 4 294 967 295. To by vyžadovalo, aby každé zariadenie, ktoré je pripojené do siete informovalo všetky ostatné zariadenia o svojej adrese a mieste, kde sa nachádza. Tak by si museli všetky sieťové zariadenia (routre) pamätať cestu teoreticky až ku 4 miliardám staníc. Vzhľadom na výpočtovú kapacitu zariadení na začiatku 80-tych rokov, kedy bol protokol IPv4 vytvorený, bol takýto prístup nereálny.

Preto bola do adresácie zavedená určitá **hierarchia** a princíp, že zariadenia v jednej fyzickej sieti (napr. v sieti fy. IBM) majú pridelené adresy, ktoré majú rovnaký začiatok, tzv. prefix. To znamená, že IP routrom mimo siete IBM stačí poznať tento prefix na to, aby vedeli správne smerovať všetky IP adresy z tejto siete, z tohto **adresného priestoru**.

IP adresy v jednej skupine majú počiatočnú časť (**prefix**) spoločnú, adresy sú pridel'ované v celých skupinách pre jednotlivé siete. Táto časť označuje samotnú sieť, koniec identifikuje konkrétnu stanicu v danej sieti. Otázne samozrejme je, aká veľká časť adresy má byť vyhradená ako spoločná. Je zrejme, že dlhý prefix umožňuje vytvoriť viacero sietí, avšak príliš dlhý prefix zas uberá miesto pre označenie staníc v sieti a tak umožňuje adresovať iba niekoľko staníc v sieti, t.j. malé siete.

Preto bol celý priestor IP adries rozdelený na tzv. triedy – Triedu A, B, C, D a E. Počiatkové bity adresy určujú, o ktorú triedu adresy ide (do ktorej triedy daná adresy patrí). Princíp znázorňuje nasledovná tabuľka:

IP adresa	trieda
0xxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx	A
10xxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx	B
110xxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx	C
1110xxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx	D
1111xxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx	E

Adresy triedy E sú určené pre experimentálne účely, resp. budúce použitie.
 Adresy triedy D sú tzv. **multicast adresy** (pozri ďalej).
 Adresy triedy A, B, C (**unicast-ové adresy**) sú určené pre označenie staníc, resp. interfejsov staníc pripojených do IP siete.

Pozor!!!

V nasledujúcom texte, ak nebude uvedené inak, bude sa pod pojmom IP adresa rozumieť IP adresa triedy A, B, resp. C. (Tento predpoklad býva implicitne používaný aj v praxi.)

Následne boli prijaté tieto pravidlá:

- Adresy z triedy A budú pridelené pre veľké siete a prefix bude dlhý 8 bitov.
- Adresy z triedy B budú pridelené pre stredne veľké siete a prefix bude dlhý 16 bitov
- Adresy z triedy C budú pridelené pre malé siete a prefix bude dlhý 24 bitov

IP adresa pridelená zariadeniu v sebe zahŕňa identifikáciu siete (*netid* – už spomínaný prefix) a identifikáciu zariadenia v danej sieti (*hostid*). Podľa dohody majú niektoré adresy špeciálny význam. Platí pravidlo, že nuly znamenajú „tento konkrétny“, jednotky znamenajú „všetko“, „všetci“.

adresa siete

- na označenie siete sa používa nultá IP adresa v danej sieti (*hostid* = 00000..)
- používa sa pri routovaní, routre si v tabuľkách ukladajú iba tieto adresy
príklad: 01010101 00000000 00000000 00000000 (85-a sieť)

smerovaný (directed) broadcast

- posledná adresa v danej sieti (*hostid* = 1111..)
- nikdy nesmie byť použitá ako zdrojová adresa
príklad: 01010101 11111111 11111111 11111111 (smerovaný broadcast do 85-jej siete)

limitovaný (limited) broadcast

- je to broadcast na lokálnej sieti, bez potreby znalosti samotnej adresy siete (rovnako dobre funguje na všetkých sieťach)
- sú to samé 1-ky, t.j. 11111111 11111111 11111111 11111111
- možné využiť pri start-up procedúre, kedy ešte nie je známa adresa siete (ani hosta samotného)
- nikdy nesmie byť použitá ako zdrojová adresa
- tento broadcast routre nešíria mimo siete, v ktorej bol vygenerovaný (spôsobilo by to zaplavenie všetkých sietí)

interný loopback

- adresa vyhradená pre interný loopback (vnútornú spätnú slučku)
- používa sa napr. na testovanie sieťovej karty, pri odlaďovaní programov a pod.
- nikdy by sa nemala objaviť na sieti
- pre tieto účely sú vyhradené adresy 172.x.x.x, spravidla sa používa **127.0.0.1**

Tento host na tejto sieti

- sú to samé nuly, t.j. 00000000 00000000 00000000 00000000
- *platné iba pri štartovaní systému na označenie zdroja*
- *nikdy nesmie byť použitá ako cieľová adresa*

konkrétny host na tejto sieti

- sieťová časť je vyplnená nulami, časť pre *hostid* označuje id zariadenia v danej sieti (*netid* = 000...)
- 00000000 00000001 00000001 00000001

Maskovanie

IP adresa obsahuje sieťovú časť *netid* (jej veľkosť je daná triedou A,B,C) a časť pre označenie hosta *hostid*. Jednotlivé datagramy v IP sieti sú cez sieť smerované (routované) zariadeniami nazývanými **router**. Sú to

väčšinou zariadenia, ktoré majú aspoň 2 interfejsy a spájajú rôzne siete. Routers majú tabuľku s adresami sietí a na základe týchto tabuliek vedú, cez ktorý interfejs možno danú sieť dosiahnuť. Každý IP datagram, ktorý majú smerovať obsahuje cieľovú IP adresu. Preto na základe porovnania sieťovej časti (*netid*) tejto cieľovej adresy s *netid* časťou adresy vo svojej routovacej tabuľke vedú príslušný datagram nasmerovať na správny interfejs.

Samotné porovnanie je realizované pomocou tzv. **maskovania**, kedy sa prijatá IP adresa binárne vynásobí s tzv. **maskou**. Masky sú 32 bitové binárne čísla, ktoré určujú, koľko bitov z IP adresy určuje sieť (*netid*) a koľko bitov určuje konkrétne zariadenie (*hostid*).

Pre triedy adres A, B, C sú masky nasledovné:

IP maska	trieda
11111111 00000000 00000000 00000000	A
11111111 11111111 00000000 00000000	B
11111111 11111111 11111111 00000000	C

Príklad maskovania:

(na miestach, kde násobíme jednotkou zostane pôvodné číslo, na miestach násobenia nulou sú výsledkom nuly)

IP adresa (triedy C):	11000000 10101000 01111011 00001100
maska	<u>11111111 11111111 11111111 00000000</u>
výsledok po odmaskovaní	11000000 10101000 01111011 00000000

Z príkladu vidno, že odmaskovaním ľubovoľnej adresy dostaneme adresu siete. Routerom potom naozaj stačí pracovať iba s adresami sietí a tieto porovnávať s odmaskovanými IP adresami.

Nevýhody IP adresovania

- adresa označuje jednoznačne zariadenie, ale aj sieť – teda príslušnosť zariadenia k sieti. Pre routovanie je to dobrá vlastnosť, lebo routeri nemusia držať vo svojich tabuľkách všetky adresy osobitne, iba adresu siete. Z hľadiska **mobility** je to však nevýhoda, lebo keď sa zariadenie premiestni do inej siete, tak si musí zmeniť adresu.
- zápis adresy v binárnom tvare je **Big endian style** (most significant byte first) – t.j. niektoré počítače musia konvertovať (napr. Pentium:)
- keď napr. v sieti s adresami typu C narastie počet zariadení cez 254, tak všetkým treba zmeniť IP adresu, čo spravidla vyžaduje odstavenie celej siete (**zlá škálovateľnosť**)
- triedy A, B, C napevno určujú veľkosť pridelovaných adresných priestorov pre jednotlivé siete (organizácie) a preto sú pridelované spravidla väčšie priestory, ako reálne treba. Zvyšné, nepoužívané adresy sú natvrdo alokované pre danú sieť (**neefektívne využívanie**)

Poznámka:

Aby boli adresy na Internete jednoznačné, prideluje ich IANA (Internet Assigned Number Authority), resp. ňou poverené organizácie.

Spôsoby riešenia nedostatku IP adres

Pôvodne sa zdalo, že IP adres, resp. priestorov bude dosť – ale rýchlo sa „minuli“ (hlavne A, a už aj B triedy). Preto bolo potrebná nájsť spôsoby, ako tento nedostatok adres riešiť.

Proxy ARP

Je metóda vhodná na prepojenie dvoch fyzických sietí routerom, pričom obe siete používajú adresy z toho istého adresného priestoru (A, B resp. C triedy). Má viacero nevýhod – funguje iba keď je použitý protokol ARP (pozri neskôr), nezahŕňa bezpečnostné pravidlá, adresy treba zariadeniam pridelovať korektne a spravidla ručne. Je nepoužiteľné pre zložitejšie topológie. (RFC 1027)

Princíp:

Zariadenie proxy ARP „klame“ stanice v jednej sieti, že má IP adresu zariadenia (ktoré je v druhej sieti). Potom majú jednotlivé stanice v ARP tabuľke viacero IP adres previazaných s tou istou fyzickou adresou (adresou zariadenia proxy ARP) – teoreticky všetky IP adresy z druhej siete. Preto v prípade, že je na stanicach implementovaná ochrana (napr. proti spoofingu), tak táto metóda nefunguje.

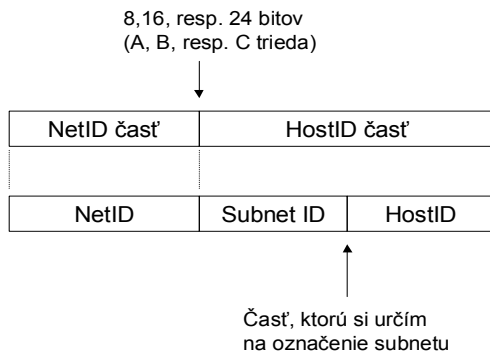
Subnetting

Je technika na rozdelenie jedného adresného priestoru medzi viacero fyzických sietí. Táto metóda je veľmi rozšírená technika, pretože bola štandardizovaná.

Princíp:

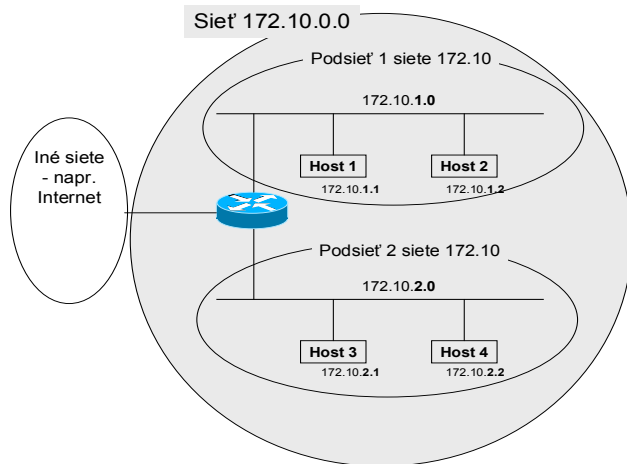
Pôvodné adresné priestory triedy A, B a C prideliť IANA. S prideleným priestorom si môžem robiť čo chcem, sám si ho rozdeľujem na menšie – ľubovoľne veľké, dokonca môžem vytvoriť aj hierarchické delenie.

V nasledovnom príklade je ešte možné rozdeliť sieť 172.10.1.0 na viacero podsietí s jemnejšie delenými maskami. S adresami to pri subnettingu vyzerá nasledovne:



Príklad 1:

Sieti (organizácii) sa prideliť jeden adresný priestor triedy B, ktorý si môže organizácia vo vnútri rozdeliť na tzv. subnety, napr. na 254 adresných priestorov veľkosti C (každý pre 254 staníc/hostov). Nasledujúci obrázok znázorňuje takéto rozdelenie jednej siete:

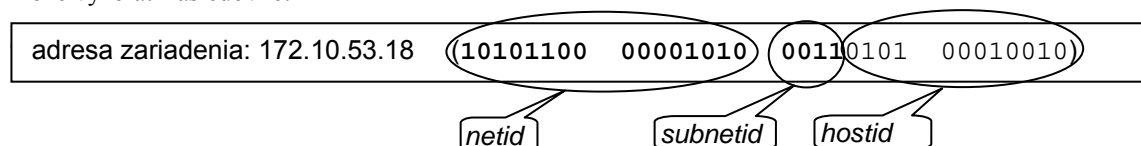


Subnetting možno implementovať pomocou tzv. **subnet masiek**, t.j. masiek, ktoré označujú počet bitov pre označenie *netid* a *subnetid*

Príklad 2:

Adresný priestor triedy B rozdelíme na 16 subnetov, t.j. v treťom Bajte vyhradíme 4 bity na subnet. Maska je teda 11111111 11111111 11110000 00000000, alebo inými slovami 20-bitová maska

Následne je možné definovať jednotlivé subnety - 0000, 0001, 0010 1110, 1111. Na konkrétnom príklade to môže vyzeráť nasledovne:



Dôležité!!!:

Nultá podsieť (samé nuly) a posledná podsieť (samé jednotky), t.j. v predošlom príklade podsiete 0000 a 1111 sa nepoužívajú, lebo by nebolo jasné, či 10101100 00001010 **00000000** 00000000 označuje nultú podsieť alebo celú sieť triedy B, a pri 10101100 00001010 **11111111** 11111111 by nebolo jasné, či sa jedná o broadcast do poslednej podsiete (16-tej v poradí) alebo broadcast do celej siete triedy B (do všetkých podsietí).

Samozrejme, v spojení s maskou by bolo možné použiť aj tieto podsiete, ale dohoda a prax je taká, že sa tieto adresné priestory nepoužívajú (adresy z týchto častí zostanú nepoužitú).

Formy zápisu IP adries a masiek

IP adresu je teda potrebné zapisovať vždy aj s maskou, a to v nasledovnom tvare:

a) binárny zápis

host: 10101100 00001010 00110101 00010010

maska: **11111111 11111111 11110000 00000000**

b) dekadický zápis

host: 172.10.53.18 maska: **255.255.240.0**

c) skrátený zápis

host/maska: 172.10.53.18/**20**

Classless Inter-Domain Routin (CIDR)

Je ďalšou metódou na efektívnejšie využitie IP adresného priestoru, niekedy označované ako **Supernetting**, pretože v tomto prípade ide o spájanie viacerých adresných priestorov triedy C do jedného väčšieho bloku. Samozrejme, adresné priestory triedy C musia ísť za sebou (v binárnom zápise).

Princíp:

IANA mi prideli priestor, povedzme s maskou 20 bitov. Tento priestor je podľa pôvodnej konvencie z časti začínajúcej 110xxxxx, t.j. adresné priestory triedy C. Metóda CIDR už nepozera na pôvodné „maskovanie napevno“ podľa tried A, B a C - jednoducho zvyšný priestor prideliuje jemnejšie, presne podľa potreby

Príklad:

Potrebuje priestor pre 1000 staníc. Jeden adresný priestor triedy C nepostačuje pretože obsahuje iba 254 adries. Adresný priestor triedy B je na druhej strane prílišným plytváním, pretože obsahuje cca 65 tisíc adries. Preto použijeme adresy zo štyroch za sebou idúcich priestorov typu C. Číslo 1000 (počet staníc v mojej sieti) je v binárnej sústave zapisateľné pomocou desiatich bitov (max 1024 čísel). Časť *hostid* bude potom dlhá 10 bitov a časť *netid* 32-10=22 bitov, t.j. potrebujem **22 bitovú masku**.

Poznámka:

Keby sa hneď od začiatku prideliovali priestory takýmto spôsobom, t.j. nie iba s pevnými maskami, ale presne podľa potreby, tak by ich dnes nebolo tak málo.

Privátne adresy:

Na šetrenie priestoru boli vyhradené tzv. privátne adresy (RFC 1918 - Address Allocation for Private Internets).

Tieto IP priestory sa podľa dohody nevyskytujú v Internete.

- 1 priestor triedy A **10.x.x.x**
- 16 priestorov triedy B **172.16.x.x až 172.31.x.x**
- 256 priestorov triedy C **192.168.0.x až 192.168.255.x**

Tieto adresy je možné použiť vo vnútorných sieťach pričom je predpoklad, že nebudú pripojené do celosvetovej siete Internet, presnejšie povedané tieto adresy nebudú použité v sieti Internet. Takéto siete je však možné pripojiť pomocou techniky NAT.

NAT – Network Address Translation

NAT znamená preklad IP adries. IP adresy pridelené vnútorným staniciam (vnútorné IP adresy) prekladá tzv. NAT router na globálne IP adresy smerom von, a smerom dnu vykonáva reverzný proces. Zo strany Internetu vnútorné stanice vystupujú pod globálnou adresou. Pri transparentnom NAT vnútorné stanice o tom ani nevedia.

Existuje tzv. **statický NAT**, kedy je pevne nadefinované, ktorá vnútorná adresa sa prekladá na ktorú globálnu (one-to-one mapping), ďalej **dynamický NAT** (preklad množiny vnútorných adries na množinu globálnych)

a **dynamický NAT s overloadom** (preklad/mapovanie väčšej množiny vnútorných adries na menšiu množinu globálnych, často iba na jedinú globálnu adresu).

Multicast

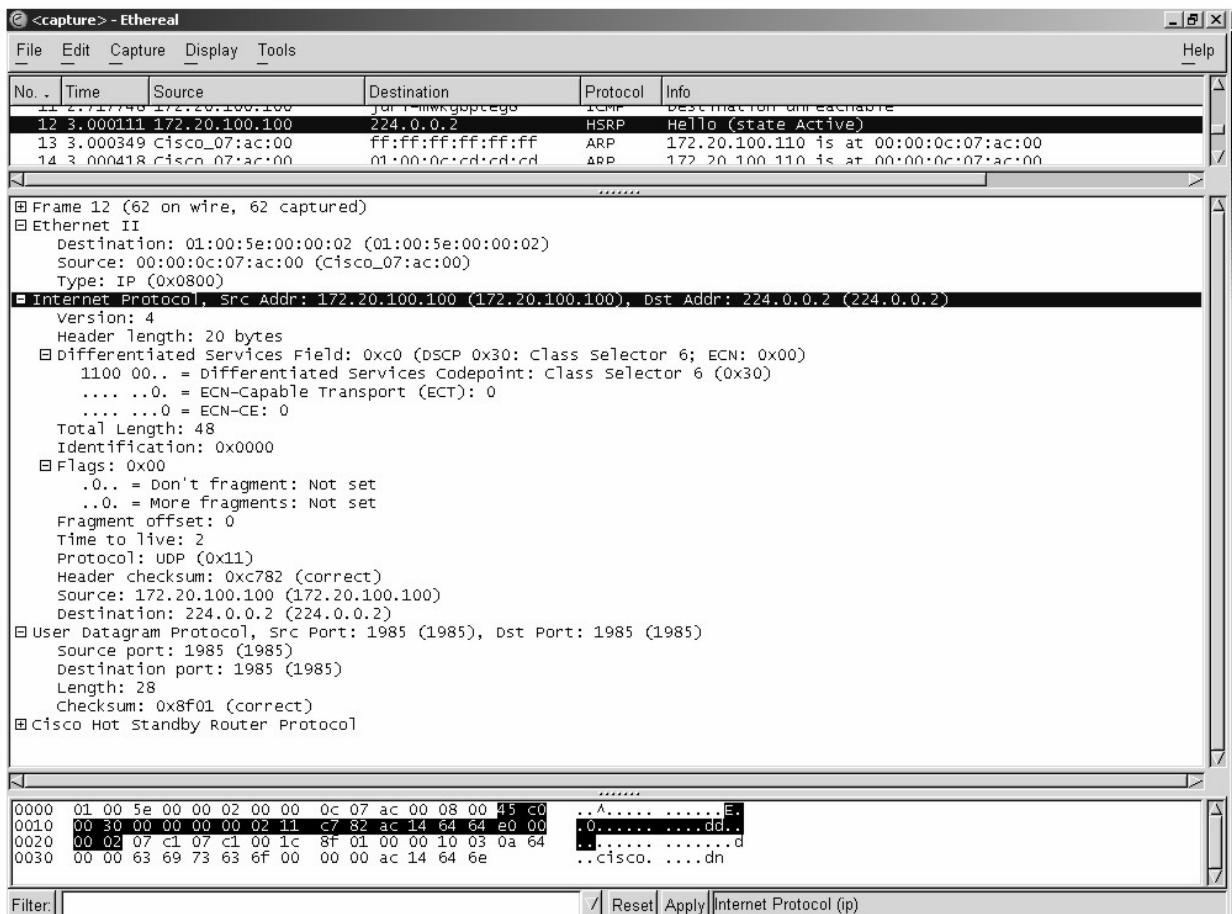
IP multicast je adresovanie (označenie) určitej skupiny príjemcov pomocou jednej IP adresy. V IP protokole sú ako multicastové adresy vyhradené adresy triedy D, (začínajúce 1110) – t.j. 28 bitový priestor (268 435 456 adries). Sú definované tzv. well-know multicast adresy, napr. adresa 224.0.0.1 je určená ako „all hosts“ multicast address, adresa 224.0.0.2 označuje všetky routre (All routers), adresa 224.0.0.5 (All OSPF routers) a podobne.

Unlike IPv4 unicast address assignment, where blocks of addresses are delegated to regional registries, IPv4 multicast addresses are assigned directly by the IANA. Current assignments appear as follows [IANA]:

224.0.0.0	- 224.0.0.255	(224.0.0/24)	Local Network Control Block
224.0.1.0	- 224.0.1.255	(224.0.1/24)	Internetwork Control Block
224.0.2.0	- 224.0.255.0		AD-HOC Block
224.1.0.0	- 224.1.255.255	(224.1/16)	ST Multicast Groups
224.2.0.0	- 224.2.255.255	(224.2/16)	SDP/SAP Block
224.252.0.0	- 224.255.255.255		DIS Transient Block
225.0.0.0	- 231.255.255.255		RESERVED
232.0.0.0	- 232.255.255.255	(232/8)	Source Specific Multicast Block
233.0.0.0	- 233.255.255.255	(233/8)	GLOP Block
234.0.0.0	- 238.255.255.255		RESERVED
239.0.0.0	- 239.255.255.255	(239/8)	Administratively Scoped Block

IP multicast je možné previazať s Ethernetovým multicastom tak, že posledných 23 bitov z IP adresy sa prilepí k Ethernetovej multicast adrese 01.00.5E.00.00.00.

Pr.: Z IP adresy 224.0.0.2 sa stane 01.00.5E.00.00.02 (IP multicast adresy je dosť, dajú sa vyberať tak, aby mali spodných 23 bitov rôznych. Ale ak sa tak nevyberú, tak budú viaceré mapované do rovnakej MAC adresy – skupinky 2⁵). Že takéto previazanie funguje, dokazuje nasledovný obrázok:

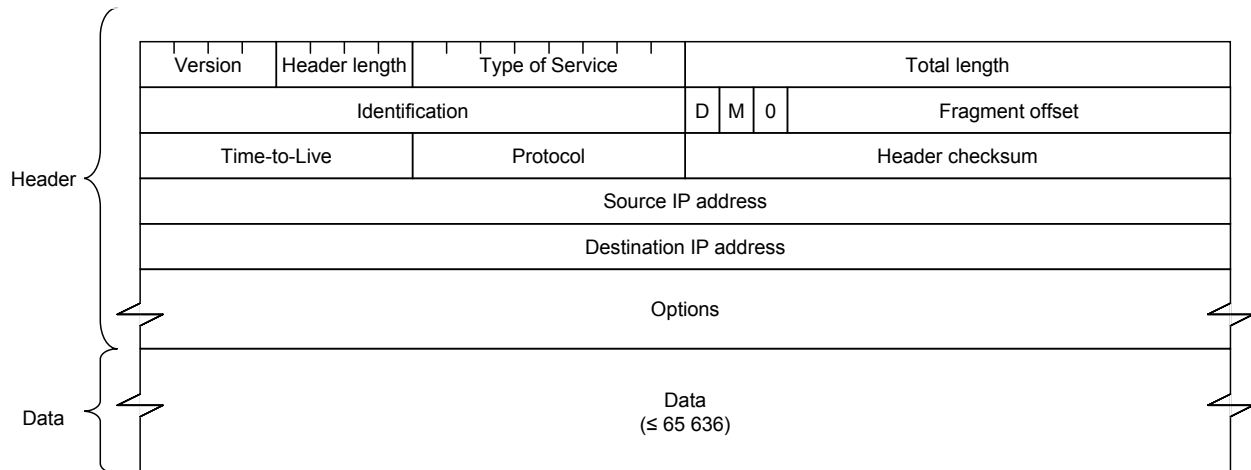


IP protokol – formát paketov

Základnou prenosovou jednotkou protokolu IP je IP datagram (alebo IP paket). Jeho formát je nasledovný (*pozri aj predošlý obrázok*):

formát IP datagramu

Bit order → 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16



Význam jednotlivých polí:

- *Version* – teraz IPv4
 - *Header Length* - udáva dĺžku hlavičky v 32 bitových slovách (bežný header má 20Bajtov – t.j. je tam Header length=5)
 - ďalšie pole je pre QoS (quality of service)
 - pôvodne **Type of Service**

Precedence	D	T	R	Not used
------------	---	---	---	----------

 - Precedence (8 prioritných úrovní)
 - 000 – normal
 -
 - 111 – network control
 - D (delay), T (throughput), R (reliability) – pomoc (tip) pre routre pri rozhodovaní, kadiaľ daný paket routovať
 - teraz Differentiated Service Codepoint **DSCP** (používa 6 bitov)

DSCP pole (6 bitov)	Not used
---------------------	----------

 - xxxxx0 – standard action
 - xxxx11, xxxx01 experimental, local use
 - default hodnota – 000000
 - určuje rôzne triedy kvality služieb – používané hlavne pri traffic shapingu, spätne kompatibilné s IP precedence
 - na cisco
 - 8 tried - xxx000 class selector – plná spätná kompatibilita s IP precedence
 - 1 trieda - 101110 – expedited forwarding – na úrovni IP precedence 5
 - 12 tried - 001dd0 až 100dd0 Assured Forwarding
(prvé 3 bity na úrovni IP precedence 1 až 4, dd – 3 úrovne pravdepodobnosti zahodenia paketu (01,10,11), t.j. najhoršia je AF13 (001110 – IP prec. iba 1, pravdepodobnosť zahodenia 3)
- *Total length*
 - 16 bitov –teda max veľkosť IP datagramu je 65535 oktetrov
 - (samozrejme platí len pre túto verziu IPv4)
 - takéto datagramy sú encapsulované do rámcov, na Ethernete je to max. 1500B (pri SNAP a IEEE 802.3 je to ešte menej - 1492B), t.j. bude potrebné datagram rozsekať na tzv. fragmenty – tento proces sa volá fragmentácia
 - opačný proces – skladanie – reassembly of fragments
 - na celý proces sú potrebné polia Identification, Flags a Fragment offset
 - Offsety sú udávané v násobkoch 8-ich oktetrov
- *Identification*
 - jedinečné číslo na identifikáciu datagramu, resp. fragmentov
- *Flags* 3 bity, D (=1) – do not fragment, M (=1) – more fragments to come
- *Fragment offset* - v 8-iciach oktetrov (v 64 bitových jednotkách), prvý fragment má offset=0

- *Time to live*
 - každý router pri prechode dekrementuje o 1, príp. viac pri čakaní vo fronte
 - keď je hodnota TTL rovná nule, tak router takýto paket zahodí a zdroju pošle ICMP správu
 - je to ochrana proti nekonečnému cykleniu, resp. blúdeniu paketu sieťou
 - používa sa pri traceroute – posielam ICMP ping, začínam s ttl = 1
 - *Protocol* – na určenie protokolu vyššej vrstvy (UDP=17, TCP=6, ICMP=1, OSPF=89)
 - *Header checksum*
 - na overenie neporušenosti hlavičky (**iba hlavičky!**)
 - IP protokol neoveruje neporušenosť datovej časti – ponechané na vyššiu vrstvu
 - prepočítava sa na každom routri – lebo každý zmení TTL (a niekedy aj iné polia)
 - *Destination a Source address*
 - 32 bitové adresy zdroja a cieľa
 - nemenia sa počas celého prechodu sieťou
 - *IP Options*
 - sú to doplnky pre ladenie siete, testovanie a security (v IPv4 nepoužiteľné)
 - majú rôzne dĺžky, musí byť zarovnané na 32bitové časti – kvôli tomu sa používa tzv. padding
 - dĺžka hlavička je obmedzená veľkosťou poľa *Header Length* a preto na *Options* zostáva relatívne málo
 - dá sa definovať, či pri fragmentácii sa *Options* majú kopírovať do každého fragmentu, alebo iba do prvého
- Niektoré *options*:
- **record route options**
 - každý router pridá do hlavičky svoju IP adresu, pokiaľ tam ešte je miesto – ak nie, tak už datagram len forwardne ďalej
 - **source route options**
 - paket si nesie info, cez ktoré routre má ísť
 - dve formy – striktnú (musí ísť iba cez definované routre – medzi nimi nič), a loose source routing – musí prejsť cez definované routre, ale medzitým aj cez iné
 - **timestamp option**
 - každý router po ceste (príp. len špecifikované) pridá časovú značku, podľa podoptions aj svoju IP

ARP protokol (Address Resolution Protocol)

Mapovanie fyzických adries s adresami vyšších vrstiev môže byť napr. priame, kedy sú IP adresy previazané s fyzickými adresami pomocou určitej funkcie (napr. IP adresa je podčasťou fyzickej adresy). Toto môže byť dost problematické – napr. pri výmene HW sieťovej karty by bolo potrebné zmeniť aj IP adresu. Preto sa používa dynamické previazanie (dynamic binding). Príkladom je protokol ARP, ktorý slúži na previazanie fyzických adries a IP adresami. Pri LAN sieťach ide o previazanie MAC adries s IP adresami. **Stanici umožňuje zistiť fyzickú (MAC) adresu cieľa, aj keď pozná len jeho IP adresu.** Tento protokol má všeobecne definovaný formát, t.j. pre rôzne dlhé adresy (nie iba 8 bit MAC adresu a 32bitovú IP adresu) a preto môže byť použitý aj na previazanie iných typov adries ako Ethernet a IP.

Princíp:

- stanica pošle pomocou (Ethernetového) broadcastu request na všetkých s otázkou “Kto má IP adresu XY?”
- následne dostane odpoveď – “Ja mám IP adresu XY”
- komunikácia prebieha na základe MAC adries –:
- samozrejme, toto nie je potrebné pred každým vysielaním paketu na danú IP adresu
- každý host si drží v dočasnej (cash) pamäti jednotlivé položky po určitý čas (timeout)
- keď niekomu odpovie, tak si aj jeho pridá do pamäte, pretože je predpoklad, že s ním bude komunikovať
- rovnako si ostatní môžu dopĺňať svoje ARP tabuľky na základe informácií z broadcast requestov
- protokol ARP je implementovaný priamo nad Linkovou vrstvou, v *Ethernet vII* rámci má *Type=0x0806*

ARP request

The screenshot shows a Wireshark capture of an ARP request. The packet list pane shows several packets, with packet 51 selected. The packet details pane shows the structure of the ARP request, including Ethernet II, ARP (Type 0x0806), and Address Resolution Protocol (request) fields. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Info
2	0.039543	jur1-mwkgbpteg8	ff:ff:ff:ff:ff:ff	ARP	who has 172.20.100.101? Tell 172.20.100.101
3	1.040963	jur1-mwkgbpteg8	ff:ff:ff:ff:ff:ff	ARP	who has 172.20.100.101? Tell 172.20.100.101
47	81.457098	Cisco_91:0a:f5	ff:ff:ff:ff:ff:ff	ARP	172.20.100.100 is at 00:09:43:91:0a:f5
51	85.502445	jur1-mwkgbpteg8	ff:ff:ff:ff:ff:ff	ARP	who has 172.20.100.100? Tell 172.20.100.101
52	85.503239	Cisco_91:0a:f5	jur1-mwkgbpteg8	ARP	172.20.100.100 is at 00:09:43:91:0a:f5

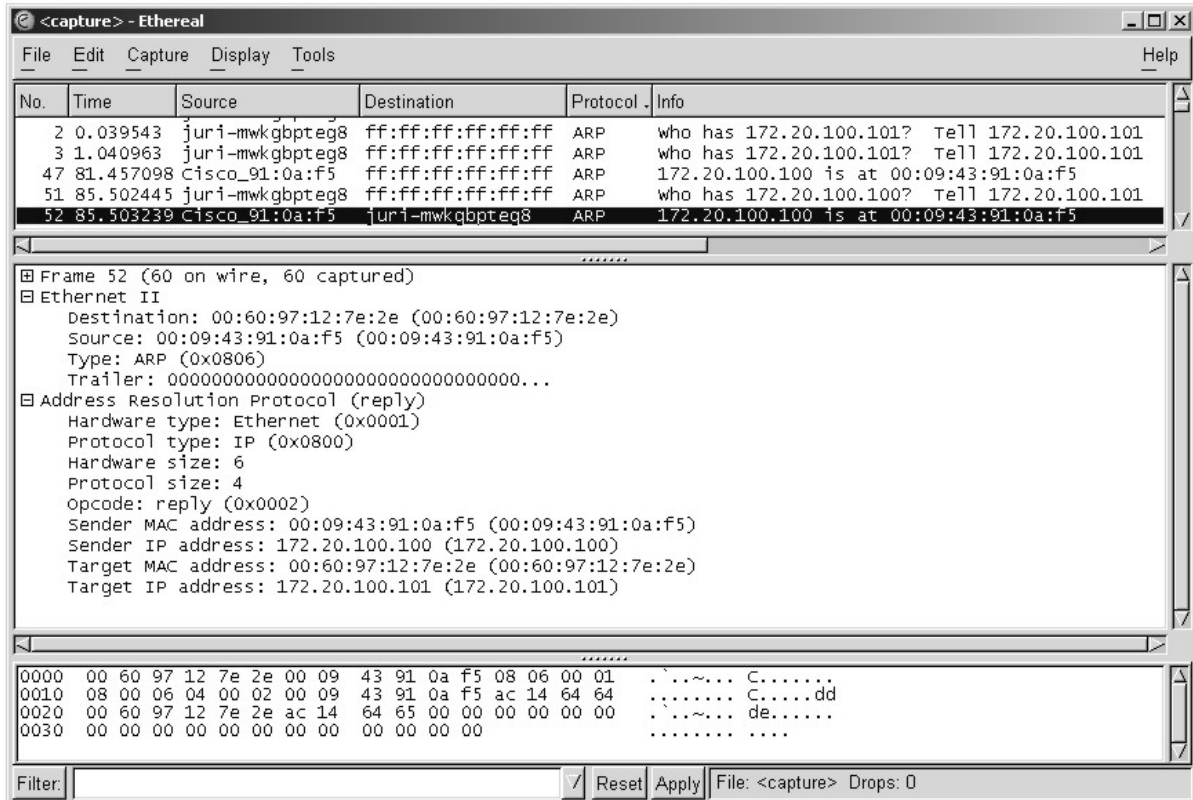
Frame 51 (42 on wire, 42 captured)

- Ethernet II
 - Destination: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
 - Source: 00:60:97:12:7e:2e (00:60:97:12:7e:2e)
 - Type: ARP (0x0806)
- Address Resolution Protocol (request)
 - Hardware type: Ethernet (0x0001)
 - Protocol type: IP (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - Opcode: request (0x0001)
 - Sender MAC address: 00:60:97:12:7e:2e (00:60:97:12:7e:2e)
 - Sender IP address: 172.20.100.101 (172.20.100.101)
 - Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
 - Target IP address: 172.20.100.100 (172.20.100.100)

```

0000  ff ff ff ff ff ff 00 60 97 12 7e 2e 08 06 00 01  ..... \.....
0010  08 00 06 04 00 01 00 60 97 12 7e 2e ac 14 64 65  ..... \.....de
0020  00 00 00 00 00 00 ac 14 64 64  ..... dd
  
```

ARP reply



RARP protokol

(Reverse Address Resolution Protocol)

Služi pre bezdiskové stanice na zistenie ich IP adresy. Takáto stanica pri bootovaní pošle RARP request broadcastom s otázkou *“Mám takúto MAC adresu. Aká je moja IP adresa?”*. Na túto otázku odpovedajú tzv. RARP servery, ktoré majú prevodné tabuľky medzi MAC a IP adresami. Kvôli redundancii aj viacero ich funguje v sieti viacero. Aby však nebolo na jeden request veľa odpovedí, tak pre určitú skupinu staníc je iba jeden RARP server vo funkcii PRIMARY. Ostatné sú BACKUP a odpovedajú iba keď sa stanica pýta druhý krát (predpoklad, že Primary neodpovedal). Formát správ je rovnaký ako u ARP, pričom RARP request používa kódy *Operation=3, Response=4*.

Nevýhodou je, že stanice používajú limited broadcast (samé jednotky) na dosiahnutie RARP servera. Ten preto musí byť na lokálnej sieti (limitovaný broadcast sa neroutuje mimo sieť).

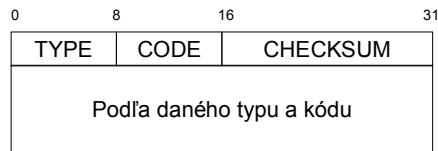
Preto existuje BOOTP protokol (bootstrap protocol), ktorý už používa UDP správy. Navyše vie pre bezdiskovú stanicu poslať tzv. image, IP adresu a subnet masku stanice, adresu default routa.

Na BootP protokole je založený protokol DHCP, ktorý navyše poskytuje automatickú alokáciu adresného IP priestoru a ďalšie voliteľné možnosti. Protokol DHCP je spätne kompatibilný s BootP a preto vie komunikovať aj so „starými BootP“ stanicami.

ICMP protokol (Internet Control Message Protocol)

Služi na posielanie riadiacich správ. ICMP správy sú enkapsulované do IP datagramov (hodnota v IP datagrame v poli *Protocol code=0x01*)

formát ICMP správy



Niektoré najpoužívanejšie ICMP správy:

- ping - Echo request (type=8), Echo reply (type=0)
- Destination unreachable (type=3)
 - network unreachable (code=0)
 - host unreachable (code=1)
 - port unreachable (code=3)
 - fragmentation needed and DF set (code=4)
 - source route failed (code=5).....
- Redirect (type=5)
- Time exceeded for a datagram (type=11)
- Address mask request (type=17), address mask reply (type=18)

ICMP echo request (ping)

The screenshot displays a network capture in Wireshark. The main pane shows a list of captured packets. Packet 7 is highlighted, showing an ICMP Echo (ping) request. The details pane for this packet is expanded, showing the following structure:

- Ethernet II**: Destination: 00:09:43:91:0a:f5 (00:09:43:91:0a:f5), Source: 00:60:97:12:7e:2e (00:60:97:12:7e:2e), Type: IP (0x0800)
- Internet Protocol**, Src Addr: 172.20.100.101 (172.20.100.101), Dst Addr: 172.20.100.100 (172.20.100.100)
 - Version: 4
 - Header length: 20 bytes
 - Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
 - 0000 00.. = Differentiated Services Codepoint: Default (0x00)
 -0. = ECN-Capable Transport (ECT): 0
 -0. = ECN-CE: 0
 - Total Length: 60
 - Identification: 0x004e
 - Flags: 0x00
 - ..0. = Don't fragment: Not set
 - ..0. = More fragments: Not set
 - Fragment offset: 0
 - Time to live: 128
 - Protocol: ICMP (0x01)
 - Header checksum: 0x1981 (correct)
 - Source: 172.20.100.101 (172.20.100.101)
 - Destination: 172.20.100.100 (172.20.100.100)
- Internet Control Message Protocol**
 - Type: 8 (Echo (ping) request)
 - Code: 0
 - Checksum: 0x3d5c (correct)
 - Identifier: 0x0200
 - Sequence number: 0e:00
 - Data (32 bytes)

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```

0000 00 09 43 91 0a f5 00 60 97 12 7e 2e 08 00 45 00 ..C....`...E.
0010 00 3c 00 4e 00 00 80 01 19 81 ac 14 64 65 ac 14 .<.N....de..
0020 64 64 08 00 3d 5c 02 00 0e 00 61 62 63 64 65 66 dd.=\..abcdef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmnopqrstuv
0040 77 61 62 63 64 65 66 67 68 69 wabcdefgh
  
```

ICMP echo reply (ping odpoveď)

The screenshot shows a network capture in Wireshark. The packet list pane displays four packets:

No.	Time	Source	Destination	Protocol	Info
6	2.003747	172.20.100.100	172.20.100.101	ICMP	Echo (ping) reply
7	3.004328	172.20.100.101	172.20.100.100	ICMP	Echo (ping) request
8	3.005183	172.20.100.100	172.20.100.101	ICMP	Echo (ping) reply
9	4.005755	172.20.100.101	172.20.100.100	ICMP	Echo (ping) request

The packet details pane for packet 8 shows the following structure:

- Ethernet II
 - Destination: 00:60:97:12:7e:2e (00:60:97:12:7e:2e)
 - Source: 00:09:43:91:0a:f5 (00:09:43:91:0a:f5)
 - Type: IP (0x0800)
- Internet Protocol, Src Addr: 172.20.100.100 (172.20.100.100), Dst Addr: 172.20.100.101 (172.20.100.101)
 - Version: 4
 - Header length: 20 bytes
 - Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
 - 0000 00.. = Differentiated Services Codepoint: Default (0x00)
 -0. = ECN-Capable Transport (ECT): 0
 -0 = ECN-CE: 0
 - Total Length: 60
 - Identification: 0x004e
 - Flags: 0x00
 - .0.. = Don't fragment: Not set
 - ..0. = More fragments: Not set
 - Fragment offset: 0
 - Time to live: 255
 - Protocol: ICMP (0x01)
 - Header checksum: 0x9a80 (correct)
 - Source: 172.20.100.100 (172.20.100.100)
 - Destination: 172.20.100.101 (172.20.100.101)
- Internet Control Message Protocol
 - Type: 0 (Echo (ping) reply)
 - Code: 0
 - Checksum: 0x455c (correct)
 - Identifier: 0x0200
 - Sequence number: 0e:00
 - Data (32 bytes)

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```

0010  00 3c 00 4e 00 00 ff 01  9a 80 ac 14 64 64 ac 14  .<.N... ..dd..
0020  64 65 00 00 45 5c 02 00  0e 00 61 62 63 64 65 66  da..E... ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76  ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                                wabcdefg hi
  
```

- rozdiel je v ICMP type (request má 8, reply 0)
- majú rovnaké sequence number

ICMP – destination unreachable

The screenshot shows a network capture in Wireshark. The packet list pane displays two packets:

No.	Time	Source	Destination	Protocol	Info
9	1.218758	172.20.100.100	juri-mwkgbpteg8	ICMP	Destination unreachable
10	2.716767	juri-mwkbotea8	172.20.100.100	NBNS	Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00>

The packet details pane for packet 9 shows the following structure:

- Ethernet II
 - Destination: 172.20.100.100 (172.20.100.100)
 - Source: juri-mwkgbpteg8 (172.20.100.101)
- Internet Protocol, Src Addr: 172.20.100.100 (172.20.100.100), Dst Addr: juri-mwkgbpteg8 (172.20.100.101)
 - Version: 4
 - Header length: 20 bytes
 - Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
 - 0000 00.. = Differentiated Services Codepoint: default (0x00)
 -0. = ECN-Capable Transport (ECT): 0
 -0 = ECN-CE: 0
 - Total Length: 78
 - Identification: 0x005a
 - Flags: 0x00
 - .0.. = Don't fragment: Not set
 - ..0. = More fragments: Not set
 - Fragment offset: 0
 - Time to live: 127
 - Protocol: UDP (0x11)
 - Header checksum: 0x1a53 (correct)
 - Source: juri-mwkgbpteg8 (172.20.100.101)
 - Destination: 172.20.100.100 (172.20.100.100)
- User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137)
 - Source port: netbios-ns (137)
 - Destination port: netbios-ns (137)
 - Length: 58
 - Checksum: 0x1edf

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```

0000  00 60 97 12 7e 2e 00 09  43 91 0a f5 08 00 45 c0  .N... E.
0010  00 38 05 d8 00 00 ff 01  9a 3a ac 14 64 64 ac 14  .8... ..dd..
0020  64 65 03 03 dc d1 00 00  00 00 45 00 00 4e 00 5a  de... ..E.N.Z
0030  00 00 7f 11 1a 53 ac 14  64 65 ac 14 64 64 00 89  .N.S. de..dd..
0040  00 89 00 3a 1e df                                .....
  
```

Multicast

There are several parameters that the network layer must define in order to support multicast communications:

Addressing. There must be a network-layer address that is used to communicate with a group of receivers rather than a single receiver. In addition, there must be a mechanism for mapping this address onto data-link layer multicast addresses where they exist.

Dynamic registration. There must be a mechanism for the computer to communicate to the network that it is a member of a particular group. Without this ability, the network cannot know which networks need to receive traffic for each group.

Multicast routing. The network must be able to build packet distribution trees that allow sources to send packets to all receivers. A primary goal of these packet distribution trees is to ensure that each packet exists only one time on any given network (that is, if there are multiple receivers on a given branch, there should only be one copy of the packets on that branch).

The Internet Engineering Task Force has been developing standards that address each of the issues described above.

Addressing. The IP address space is divided into four pieces: Class A, Class B, Class C, and Class D. Classes A, B, and C are used for unicast traffic. Class D is reserved for multicast traffic. Class D addresses are allocated dynamically.

Dynamic registration. RFC 1112 defines the Internet Group Membership Protocol (IGMP). IGMP specifies how the host should inform the network that it is a member of a particular multicast group.

Multicast routing. There are several standards available for routing IP Multicast traffic:

- RFC 1075 defines the Distance Vector Multicast Routing Protocol (DVMRP).
- RFC 1584 defines the Multicast Open Shortest Path First (MOSPF) protocol - an extension to OSPF that allows it to support IP Multicast.
- Internet standards-track drafts describe PIM (Protocol-Independent Multicast) - a multicast protocol that can be used in conjunction with all unicast IP routing protocols. These documents are entitled.

IGMP

(Internet Group Membership Protocol)

Princíp:

Aby stanica mohla prijímať a vysielat' multicast pakety aj so stanicami v iných sieťach, musí sa stať členom skupiny – tak, že informuje svoj multicast router. Routers musia podporovať multicast, aby vedeli takéto datagramy šíriť. Približne každú minútu posielajú multicast routre všetkým stanicám požiadavku na adresu 224.0.0.1 aby oznámili, o aké skupiny (multicast adresy) majú záujem. Každá stanica si udržuje tabuľku s príslušnosťou ku skupinám. Keď router nedostane žiadne odpovede na doteraz používanú skupinu, tak po vypršaní timeoutu sa s ostatnými routrami dohodne o zrušení jeho členstva v danej skupine. Na výmenu query/response informácií sa používa IGMP – Internet Group Management Protocol (podobný protokol ako ICMP - ale má iba query a response). Multicast routing používa na routovanie modifikovaný routing algoritmus.

IGMP is the protocol through which hosts exchange this information with their local routers. There is 3 versions of this protocol, defined first in RFC 1112, in 1985.

Routers use periodically the IGMP to check whether the known group members are active or not. In case there is more than one multicast router on a given subnetwork (LAN), one of them is chosen as the "querier". It has the

responsibility of keeping track of the membership state of the multicast groups which have active members on its subnetwork. After receiving a multicast packet, the router will check if there is at least one member of that group on its subnetwork. If there is, the router will forward the message to that subnetwork. Otherwise, the multicast packet will be discarded.

IGMP messages are used by routers to keep track of group memberships in their immediately connected subnetwork. The following rules apply :

1. Host sends an IGMP "report" for joining a group
2. Host will never send a report when it wants to leave a group.
3. Multicast routers send IGMP queries to the all-hosts group periodically to see whether any group members exists on their subnetworks. If no response is received after a number of queries, the router assumes that there is not any group member on the network.