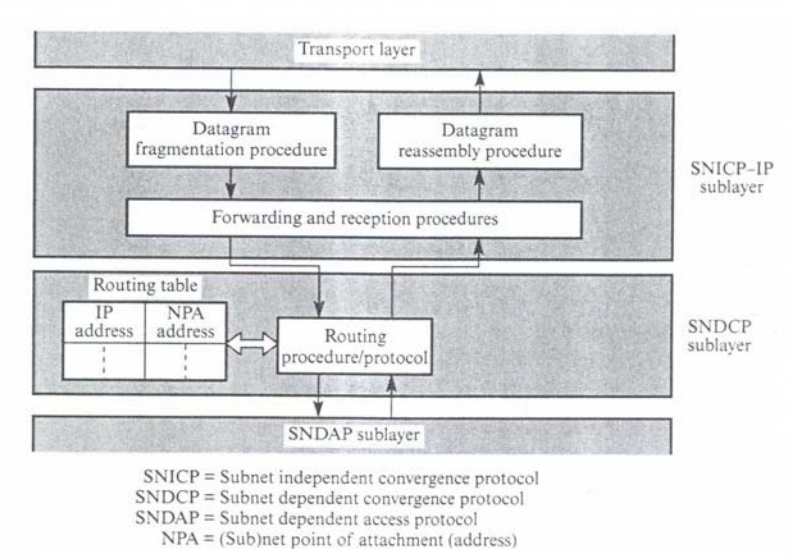
Smerovanie a prepájanie v sieťach

Dátové siete zabezpečujú prenos dát od zdoja 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 prideľované náhodne, alebo s určitou logikou. Napríklad adresy môžu byť prideľované hierarchicky (telefónne čísla), alebo prideľovaním 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.

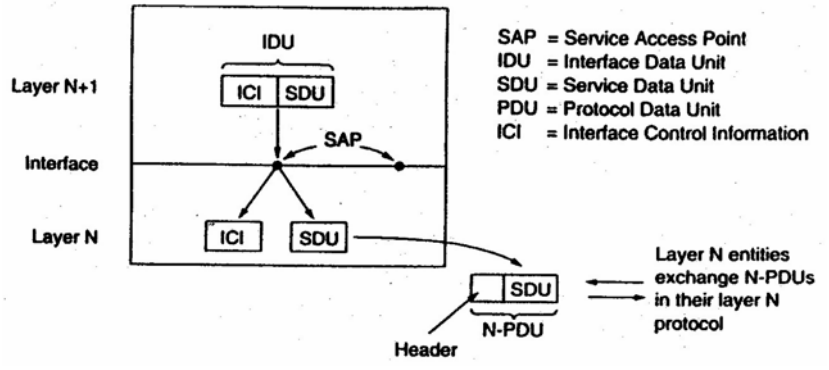
základné úlohy sieťovej vrstvy



adresovanie v sieťových vrstvách

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

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



TCP/IP a RM OSI

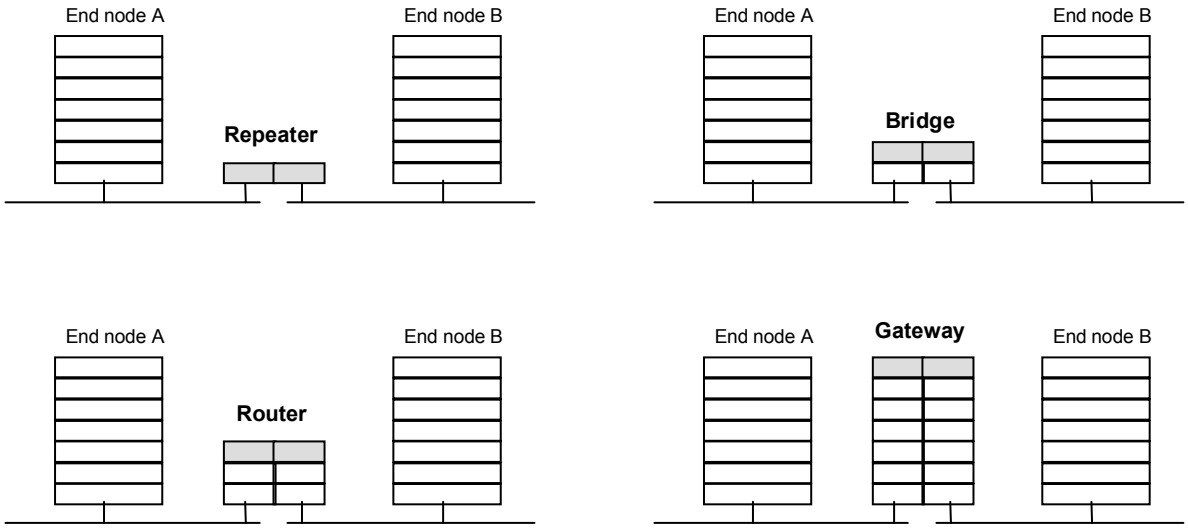
Vsúčasnosti je najrozšírenejšiou sústavou komunikačných protokolov rodina protokolov TCP/IP. Predstavuje de-fakto štandardy pre dátovú sieťovú komunikáciu. Tieto protokoly boli najskôr vytvorené, sú široko používané a až neskôr sú postupne štandardizované. Na druhej strane existujú aj protokoly, ktoré 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.

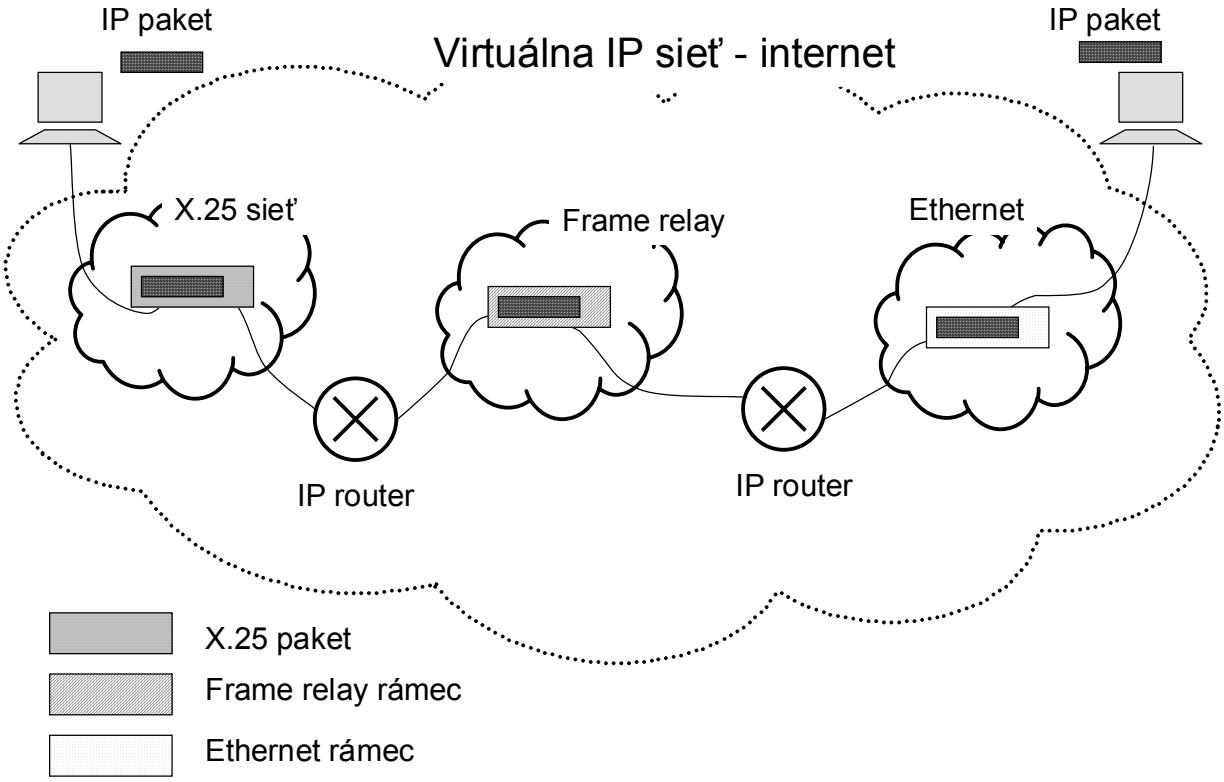
Protokoly TCP/IP sú široko používané, sú funkčné a podporované takmer všetkými výrobcami. Vznikli počas budovania výskumnej siete ARPANET, vybudovanej v rámcia projektu sponzorovaného DoD (ministertsvo obrany USA) počas studenej vojny. Hlavnou požiadavkou bolo vybudovať komunikačnú sieť, ktorá bude odolná aj voči jadrovej vojne. Špecifikácie protokolov sú vydávané vo forme tzv. **RFC** (**R**equest **F**or **C**omment), 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 routre. 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 **nespoľahlivú**, „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 univerzálnimi (IP) adresami

IP adresovanie

V protokole IPv4 sú definované tzv. IP adresy. IP adresa je 32-bitové číslo (v dvojkovej sústave zapísateľné 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 osmiciach.
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 prideliť 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 prideľovať 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 dresy v jednej skupine majú počiatočnú časť (**prefix**) spoločnú, adresy sú prideľované v celých skupinách pre jednotlivé siete. Táto časť označuje samotnú sieť, koniec identifikuje konktrétnu stanice v danej sieti. Otázne samozrejme je, aká veľká časť adresy má byť vyhradená ako spoločná. Je zrejmé, 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. Počiatočné bity adresy určujú, o ktorú triedu adresy ide (do ktorej triedy daná adresy patrí). Princíp znázorňuje nasledovná tabuľka:

IP adresa	trieda
0xxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx	**A**
10xxxxxx xxxxxxxx xxxxxxxx xxxxxxxx	**B**
110xxxxx xxxxxxxx xxxxxxxx xxxxxxxx	**C**
1110xxxx xxxxxxxx xxxxxxxx xxxxxxxx	**D**
1111xxxx xxxxxxxx xxxxxxxx xxxxxxxx	**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 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 prideľované pre veľké siete a prefix bude dlhý 8 bitov.
- Adresy z triedy B budú prideľované pre stredne veľké siete a prefix bude dlhý 16 bitov
- Adresy z triedy C budú prideľované 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-ej 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*

konrétny 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. Routre majú tabuľku s adresami sietí

a na základe týchto tabuliek vedia, 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 adries vo svojej routovacej tabuľke vedia príslušný datagram nasmerovať na správny interfejs.
Samotné porovnanie je realizované pomocou tzv. **maskovania**, kedy sa prijatá IP adresa binárne vynásobí s tzv. **maskou**. Maska je 32 bitové binárne čislo, ktoré určuje, koľko bitov z IP adresy určuje sieť (*netid)* a koľko bitov určuje konkrétne zariadenie (*hostid).*

Pre triedy adries 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	**11111111 11111111 11111111 00000000**
výsledok po odmaskovaní	*11000000 10101000 01111011 00000000*

Z príkladu vidno, že odmaskovaním ľubovoľnej adresy dostaneme adresu siete. Routrom 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 routre 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
- triedy A, B, C napevno určujú veľkosť prideľovaných adresných priestorov pre jednotlivé siete (organizácie) a preto sú prideľované spravidla väčšie priestory, ako reálne treba. Zvyšné, nepoužité adresy sú natvrdo alokované pre danú sieť

Poznámka:
Aby boli adresy na Internete jednoznačné, prideľuje ich IANA (Internet Assigned Number Authority).

Spôsoby riešenia nedostatku IP adries

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

Proxy ARP

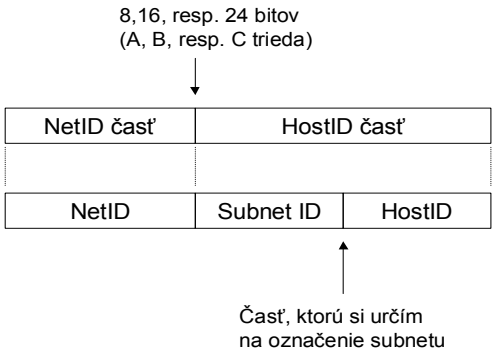
Je metóda vhodná na prepojenie dvoch fyzických sietí routrom, pričom obe siete používajú adresy z toho istého adresného priestoru (A, B resp. C triedy). Má viacero nevýhod – funguje iba keď je použitý protokol ARP (pozri neskôr), nezahŕňa bezpečnostné pravidlá, adresy treba zariadeniam prideľovať korektne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

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 adries previazaných s tou istou fyzickou adresou (adresou zariadenia proxy ARP) – teoreticky všetky IP adresy z druhej siete. Preto v prípade, že je na staniciach implementovaná ochrana (napr. proti spoofingu), tak táto metóda nefunguje.

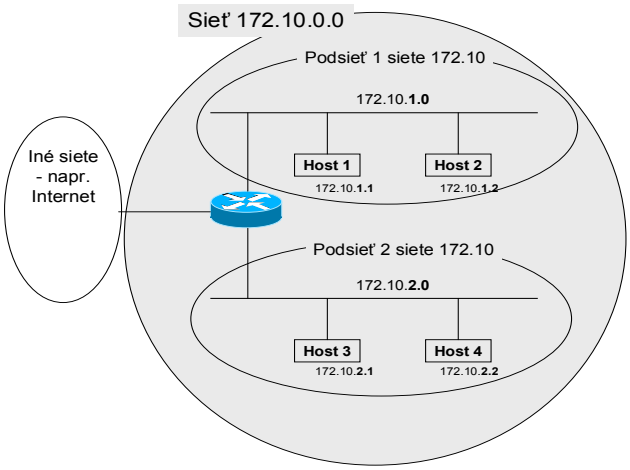
Subnetting

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 prideľuje 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:

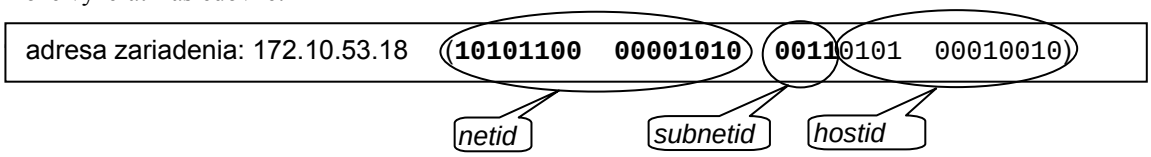


Príklad 1:
Sieti (organizácii) sa pridelí jeden adresný priestor triedy B, ktorý si môže organizácia vo vnútri rozdeliť na tzv. subnetty, napr. na 254 adresných priestorov veľkosti C (každý pre 254 staníc/hostov). Nasledujúci obrázok znázorňuje takéto rozdelenie jedej siete:



Subnetting možno implementovaťpomocou 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 jendotlivé subnety - 0000, 0001, 0010 .... 1110, 1111. Na konkrétnom príklade to môže vyzerať 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 0000`0000 `00000000` označuje nultú podsieť alebo celú sieť triedy B, a pri `10101100 00001010 1111`1111 `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 podisete, ale dohoda a prax je taká, že sa tieto adresné priestory nepoužívajú (adresy z týchto častí zostanú nepoužité).

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 väčsieho bloku. Samozrejme, adresné priestory triedy C musia ísť za sebou (v binárnom zápise).

Princíp:
IANA mi pridelí 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 nepozerá na pôvodné „maskovanie napevno“ podľa tried A, B a C - jednoducho zvyšný priestor prideľuje jemnejšie, presne podľa potreby

Príklad:
Potrebujeme priestor pre 1000 staníc. Jeden adresný priestor triedy C nepostačuje pretože obsahuje iba 254 adries. Adresný priestor triedy B je na druhej strane prílišným plytvaním, 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 zapísateľné 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 prideľovali 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. 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. Je ich 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).

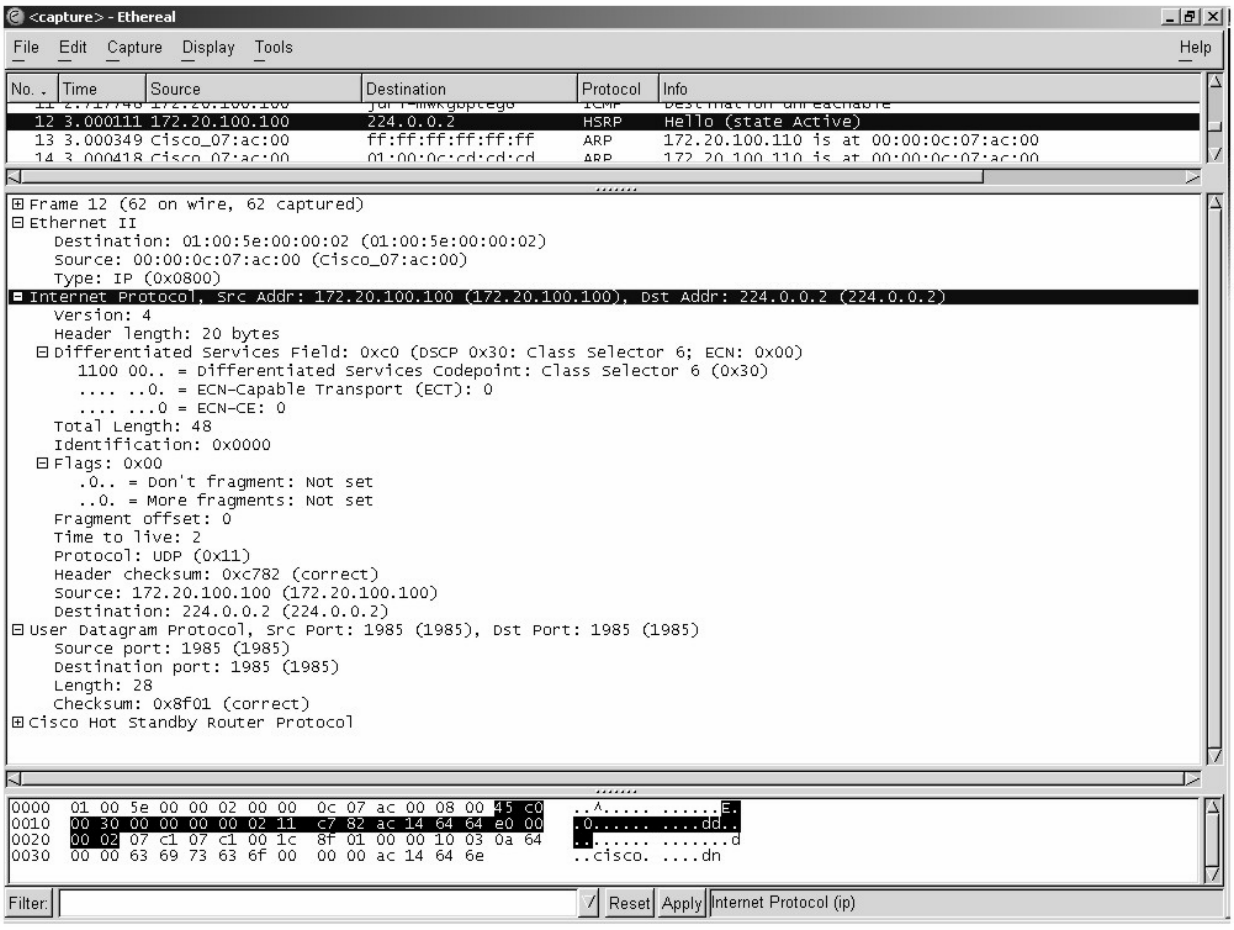
Multicast

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.

Princíp:
Aby stanica mohla prijímať a vysielať multicast pakety aj so stanicami v iných sieťach, musí sa stať členom skupiny – tak, že informuje svoj multicast router. Routre musia podporovať multicast, aby vedeli takéto datagramy šíriť. Približne každú minútu posielajú multicast routre všetkým staniciam 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.

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.

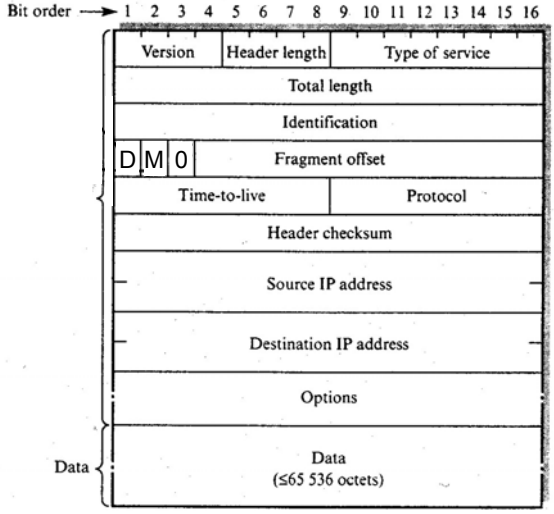
Pr.: Z IP adresy 224.0.0.2 sa stane 01.00.5E.00.00.02 (IP multicast adries je dosť, dajú sa vyberať tak, aby mali spodných 23 bitov rôznych). Že to tak aj funguje, dokazuje nasledovný obrázok:



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



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 **T**ype **o**f **S**ervice

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 Differenciated 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 - xxx**000** class selector – plná spätná kompatibilita s IP precedence
      - 1 trieda - **101**110 – expedited forwarding – na úrovni IP precedence 5
      - 12 tried - **001**dd0 až **100**dd0 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 oktetov
  - (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 oktetov
- *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 oktetov (v 32 bitových jednotkách)

- *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 checksumm*
  - 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

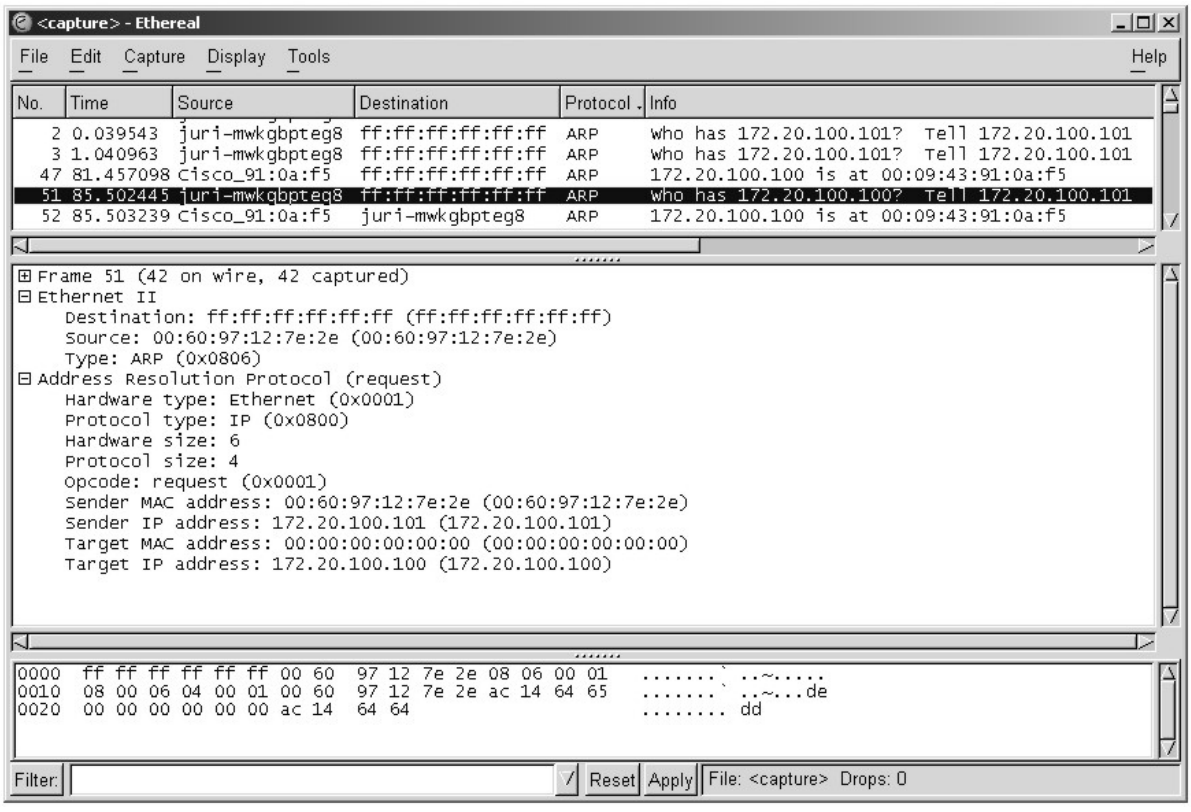
  Niektoré *options*:
  - **record route options**
    - každý router pridá do hlavičky svoju IP adresu, pokiaľ tam ešte je miesto – ak nie, tak už datagram len forwardne ďalej
  - **source route options**
    - paket si nesie info, cez ktoré routre má ísť
    - dve formy – striktnú (musí ísť iba cez definované 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)

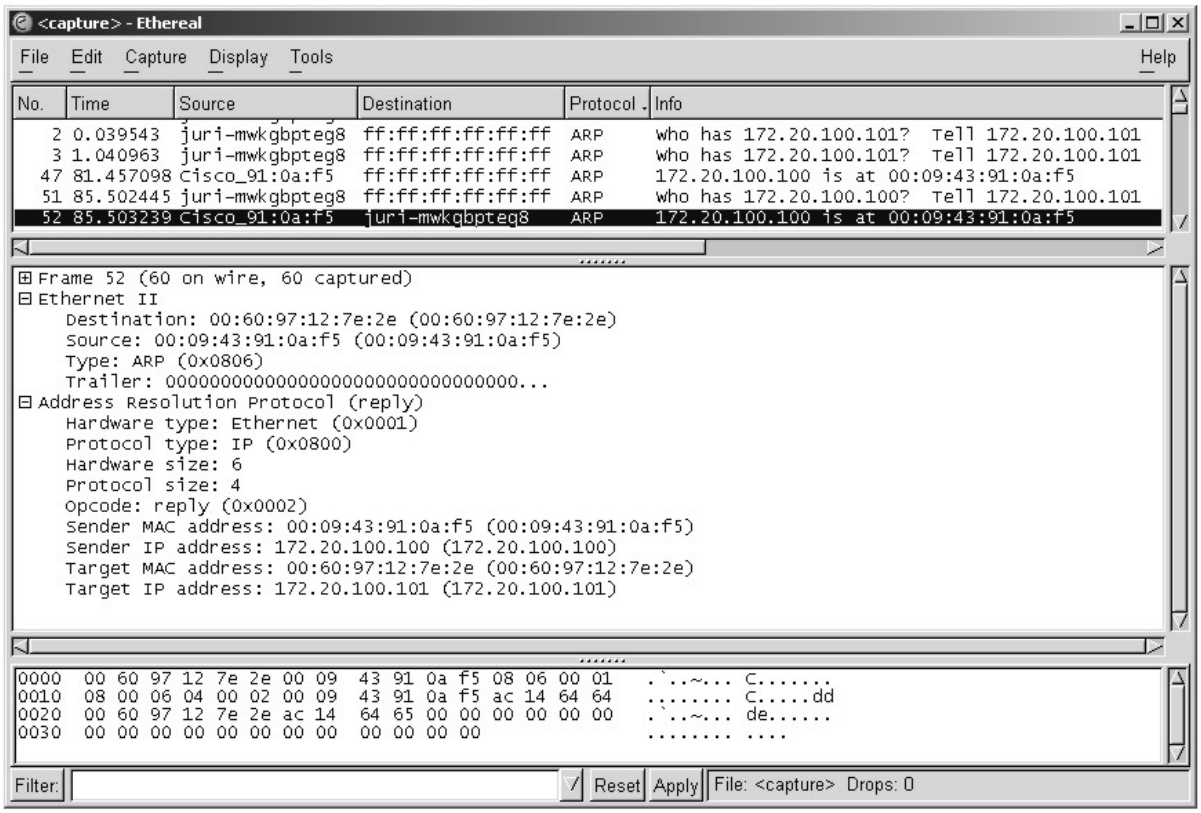
Mapovanie fyzický adries s adresami vyšších vrstiev môže byť napr. priame, kedy sú IP adresy previazané s fyzickými adresami pomocu určitej funkcie (napr. IP adresa je podčasťou fyzickej adresy). Toto môže byť dosť problematické – napr. pri výmene HW sieťovej karty by bolo potrebné zmeniť aj IP adresu. Preto sa používa dynamické previazanie (dynamic binding). Príkladom je protokol ARP, ktorý slúži na previazanie fyzických adries a IP adriesami. 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 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



ARP reply



RARP protokol
(Reverse Address Resolution Protocol)

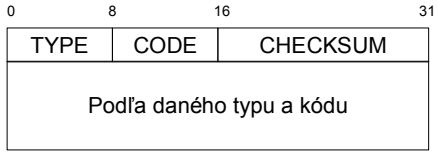
Slúži pre bezdiskové stanice na zistenie ich IP adresy. Takáto stanica pri bootovaní pošle RARP request broadcastom s otázkou *"Mám takúto MAC adresu. Aká je moja IP adresa?"*. Na túto otázku odpovedajú tzv. RARP servery, ktoré majú prevodné tabuľky mezdi MAC a IP adresami. Kvôli redundancii aj viacero 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 routra.
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)

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

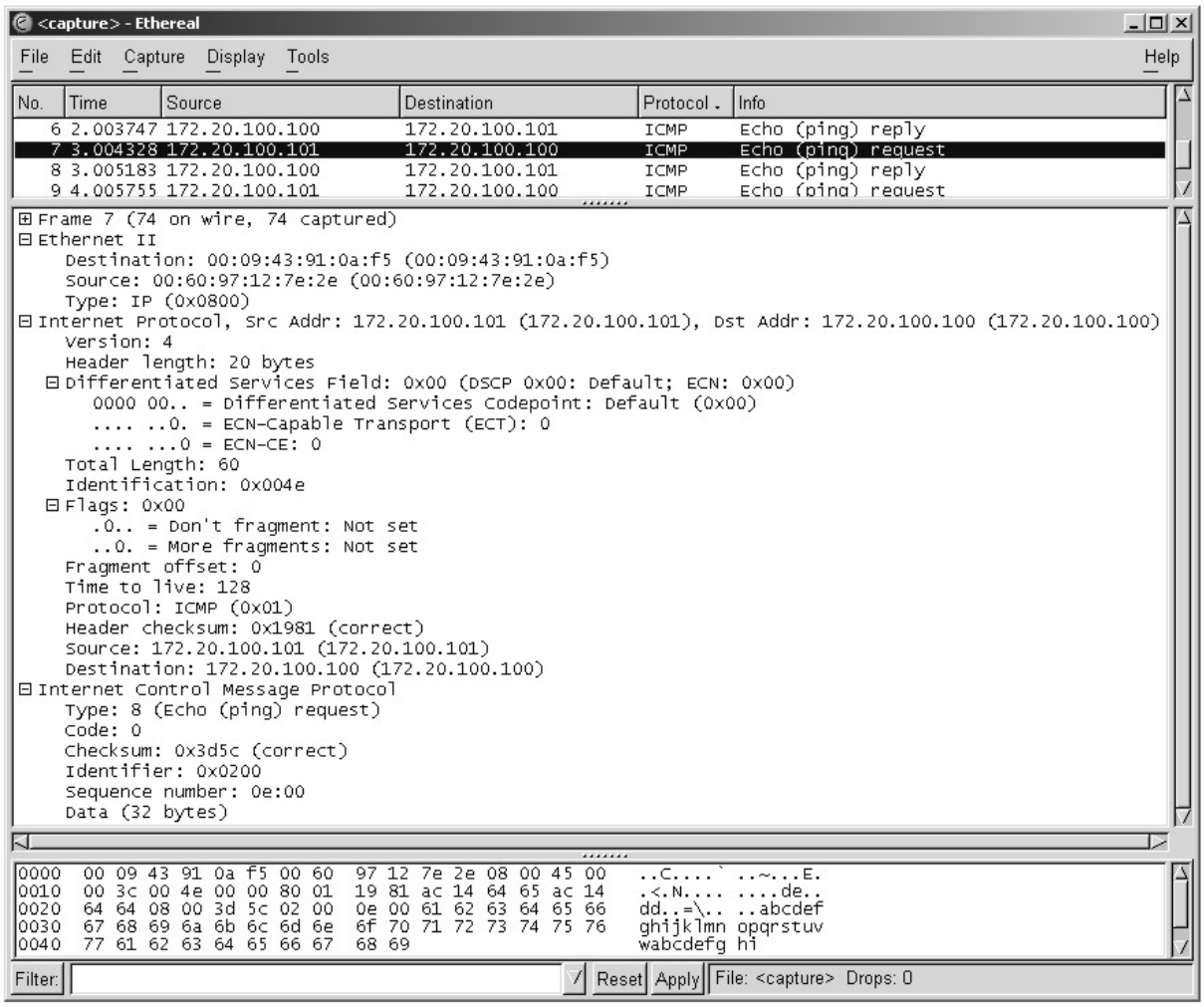
formát ICMP správy



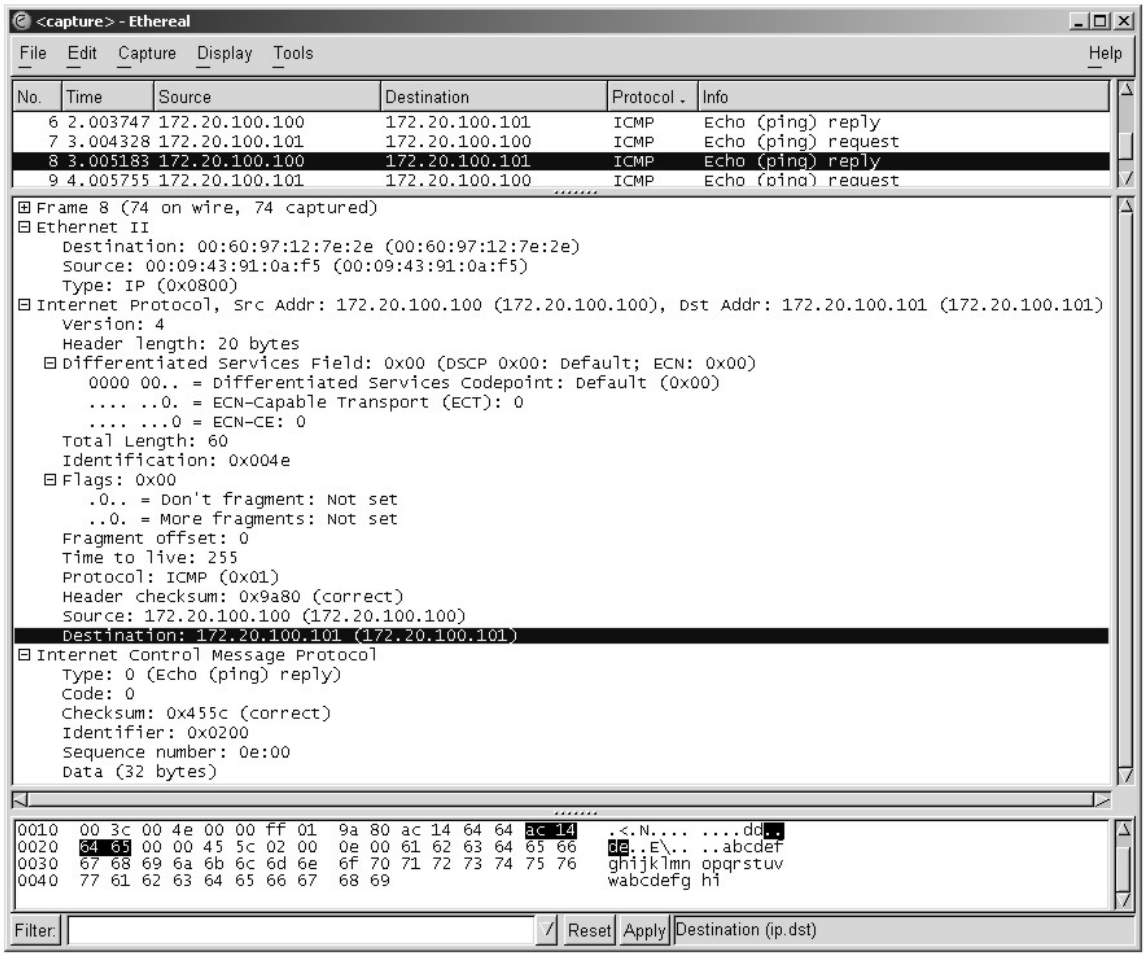
Niektoré najpoužívanejšie ICMP správy:

- 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)



ICMP echo reply (ping odpoveď)



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

ICMP – destination unreachable