#3 TCP/IP – sieťová vrstva

referenčné modely - TCP/IP a RM OSI

- určené na popis architektúry, funkcionality – vrstiev, služieb a protokolov
- tu preberieme nasledovné dva modely:

RM OSI
- na popisovanie sietí a funkcionality je vhodný RM OSI, ale jeho protokoly sú skoro nepoužiteľné
- popisuje 7 vrstiev podľa SNA (aby nebol štandard IBM – 5vrstvový)
- výsledkom je to, že Data-link a Network layer sú preplnené, ostatné skoro nič nerobia
- v každej vrstve hovorí o error-control
- data security, encryption sú vynechané, rovnako network management
- service primitives sú interrupt oriented – nevhodné pre vyššie jazyky
- bol zvolený klasický komunikačnícky prístup, nie computer oriented (telefóny zvonia, computre nie), t.j. circuit-switched (connection oriented) a nie packet-switched (connectionless)

TCP/IP
- nevýhodou je to, že nedefinuje jasne services, interface a protocol (komplikácie pre výrobcov)
- nehovorí o prvých dvoch vrstvách – Physical a Data-link
- protokoly použiteľné a používané, na popis vrstiev nevhodný
- výskumná sieť ARPANET – sponzorovaná DoD
- TCP/IP reference model – 197 Cerf&Kahn

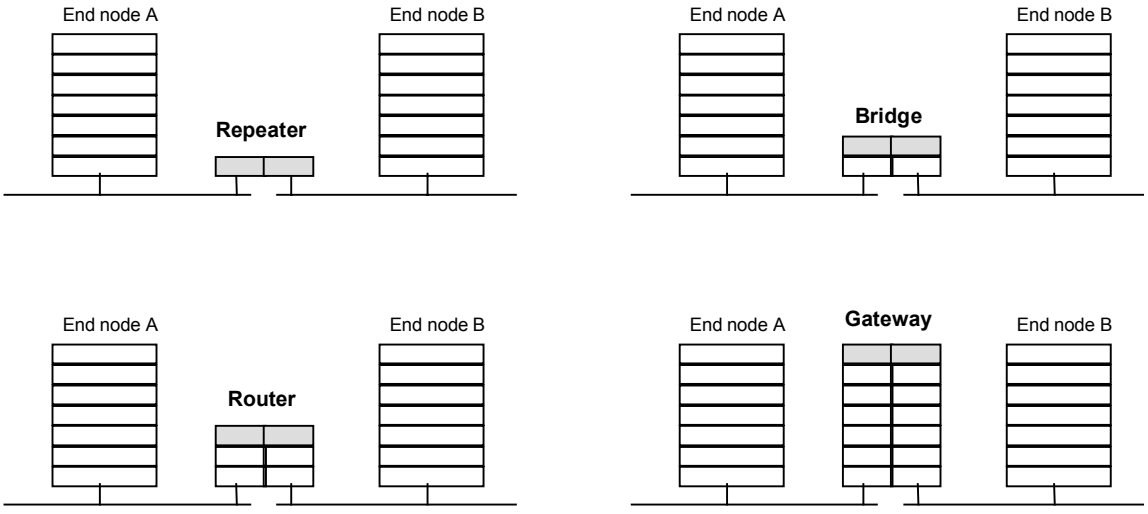
V ďalšom teda budú popísané reálne používané protokoly z TCP/IP stacku (z rodiny protokolov TCP/IP) a ako referenčný model použijeme RM OSI.

Potrebné pojmy:

RFC	- request for comment, spôsob vydávania dokumentácie k TCP/IP protokol suite
internet	- virtuálna sieť, vytvorená prepojením viacerých fyzických sietí
TCP/IP internet	- internet, ktorý na prepojenie využíva protokoly z rodiny TCP/IP
Internet	- konkrétna celosvetová sieť
Protocol suite	- rodina protokolov – pre každú vrsvu ich môže definovať niekoľko
protocol stack	- konkrétne vybraná zostava protokolov – pre každú vrstvu jeden

Internetworking

- prepájanie rôznych sietí
- môžeme prepájať na rôznych vrstvách

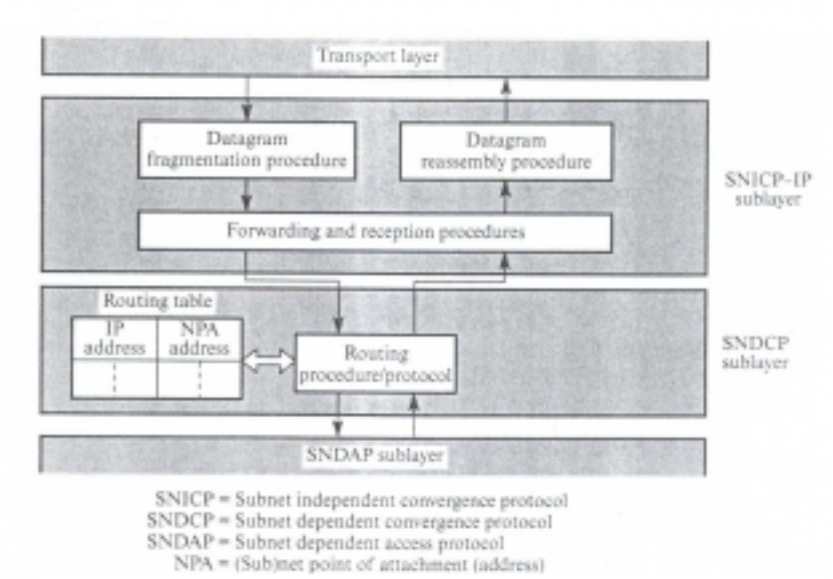


- teraz sa budeme zaoberať prepájaniu na tretej, teda sieťovej vrstve
- budeme teda hovoriť o routroch, IP protokole (Internet Protocol)

- ďalej budeme hovoriť o TCP/IP internete, teda virtuálnej sieti, ktorá vznikne prepojením viacerých fyzických sietí pomocou IP protokolu
- táto (tretia) IP vrstva je teda spoločná pre všetky siete, a z pohľadu nad ňou sa teda všetky siete tvária ako rovnocenné

Sieťová vrstva

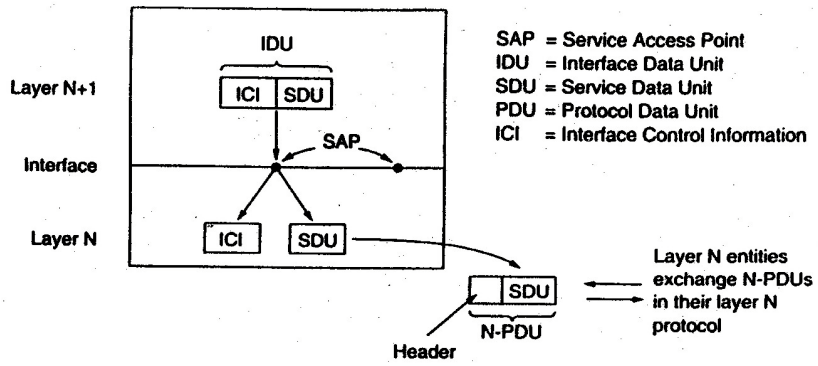
základné úlohy sieťovej vrstvy



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



IP sieťová vrstva

- základom je IP protokol – FRC 791, rok 1981
- požiadavky na sieťovú vrstvu – packet switching network based on connection less internetwork layer
- jej úlohou je vkladať pakety do ľubovoľnej siete a nechať ich nezávisle putovať do ich cieľa
- táto úloha je veľmi podobná pošte
  - rozsekám veľkú správu na menšie – zabalím do obálok
  - pošlem viacero obálok
    - – tieto nemusia dôjsť
    - môžu prísť rôzne neskoro
    - môžu ísť rôznymi cestami

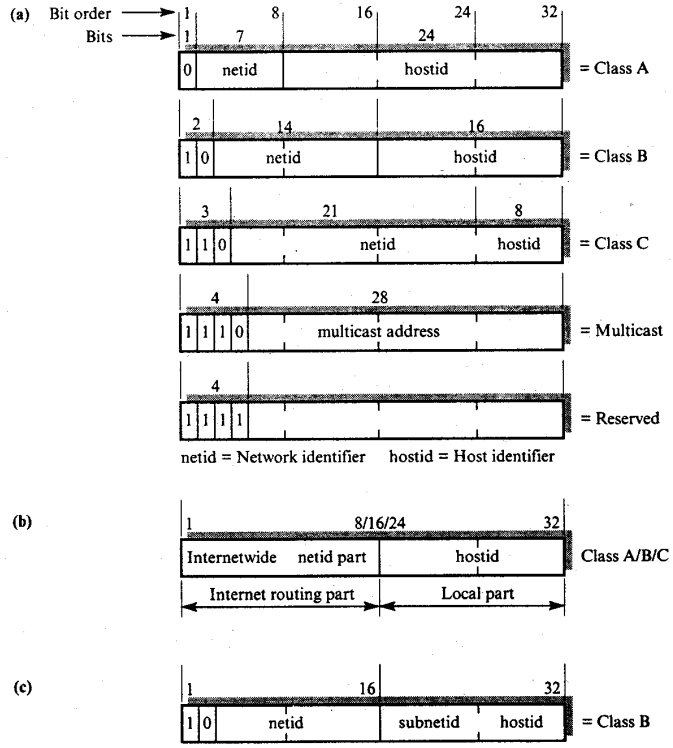
<u>adresovanie v IP</u>

- je potrebné, aby sa všetky druhy sietí prepájaných na 3. vrstve pomocou IP protokolu tvárili rovnocenne

- je teda potrebné použiť jednotné adresovanie, všeobecne unikátne adresy nezávislé od typy konkrétnej fyzickej siete. Toto univerzálne adresovanie je v IP protokole realizované tzv. IP adresami.
- následne bude teda potrebné vykonať previazanie adries fyzickej siete s univerzálnimi (IP) adresami

IP adresovanie

- v IPv4 sú definované 32 bitové adresy, tzv. IP adresy



- každý host v TCP/IP má pridelenú 32-bitovú IP adresu
- niektoré stanice môžu mať pridelených aj viac IP adries – multihomed hosts alebo routre
- IP adresa v sebe zahŕňa adresu hosta, ale zároveň aj adresu siete (prvá časť adresy)

adresa siete

- IP adresa môže označovať ako hosta, tak aj konkrétnu sieť
- podľa dohody sieť adresujeme tak, že všetky bity na konci (v hostid časti) sú rovné 0
  príklad: 01010101 00000000 00000000 00000000 (85-a sieť triedy A)

smerovaný (directed) broadcast

- všetky bity na konci – v hostid časti – sú rovné 1
  príklad: 01010101 11111111 11111111 11111111

limitovaný (limited) broadcast

(local network bradcast address)

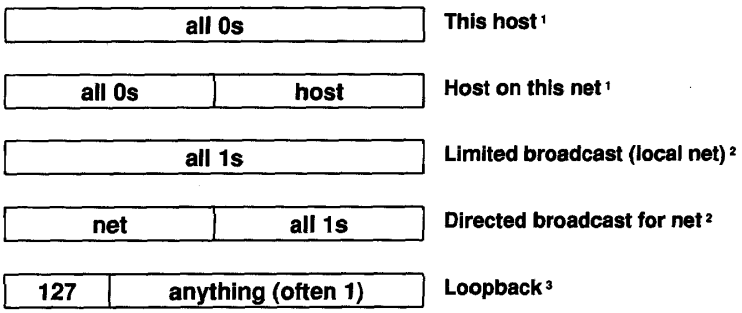
- je to broadcast na lokálnej sieti, bez potreby znalosti samotnej adresy siete (rovnako dobre funguje na všetkých sieťach)
- sú to samé 1-ky, t.j. *11111111 11111111 11111111 11111111*
- možné využiť pri start-up procedúre, kedy ešte nie je známa adresa siete (ani hosta samotného)
- tento broadcast sa samozrejme nešíri mimo sieť (to by bol dobrý kolaps)

Platí teda:
1-ky znamenajú „Všetky“
0-znamená „tento jeden konkrétny“

Dotted decimal notation

pri bežnej komunikácia sa nepoužíva zápis v dvojkovej sústave, používa sa zápis vo forme 4-roch desiatkových čísel oddelených bodkou,
napr. 192.168.123.12 (11000000 10101000 01111011 00001100)

špeciálne adresy



Notes: 1 Allowed only at system startup and is never a valid destination address.
2 Never a valid source address.
3 Should never appear on a network.

Nevýhody IP adresovania
- adresa označuje jednoznačne hosta, ale aj sieť – teda príslušnosť hosta k sieti. Pre routovanie je to dobrá vlastnosť, lebo routre nemusia držať vo svojich tabuľkách všetky adresy osobitne, iba adresu siete (ostatné si odmaskuje). Z hľadiska mobility je to však nevýhoda, lebo ked sa host premiestni do inej siete, tak si musí zmeniť adresu (alebo to obísť inak, nie je to však bez roboty:)
- keď napr. v sieti s adresou typu C narastie počet hostov cez 254, tak všetkým treba zmeniť adresy – resp. minimálne masku pri CIDR – čo spravidla vyžaduje zhodenie celej siete

Aby boli adresy na Internete jednoznačné, prideľuje ich IANA (Internet Assigned Number Authority)
Zápis adresy v binárnom tvare je Big endian style (most significant byte first) – t.j. niektoré počítače musia konvertovať (napr. Pentium:)

Rozšírenie pôvodného použitia adresácie

- pôvodne sa zdalo, že IP adries, resp. priestorov bude dosť – ale rýchlo sa „minuli“ (hlavne A, a už aj B triedy)
- triedy, a teda aj masky boli napevno dané prefixom
- adresa obsahuje sieťovú časť (jej veľkosť je daná typom) a potom časť pre označenie hosta (adresu hosta v danej sieti). Routre držia u seba tabuľku s adresami sietí. Cieľovú adresu, pre ktorú chcú nájsť odchodzí interfes jednoducho odmaskujú (vynásobia ju binárne s maskou, t.j. z časti pre hosty urobí nuly, a máme adresu siete) a následne len porovnajú s tabuľkou
- každý systém, ktorý má pridelený určitý adresný priestor, si môže robiť s adresami čo chce, pokiaľ to z vonku zostane neviditeľné.

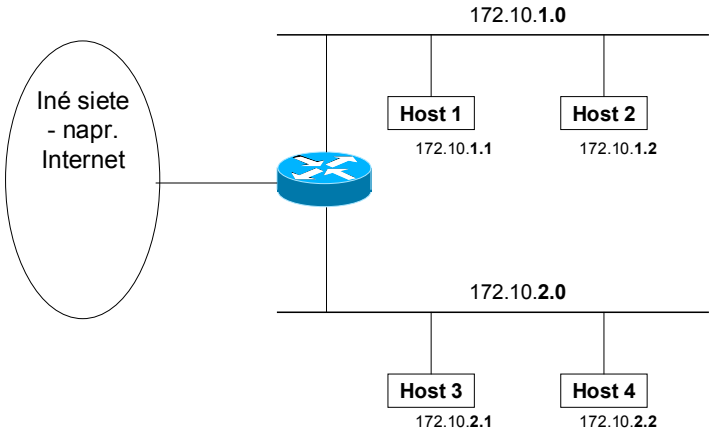
Ako teda možno zdieľať adresy z rovnakého priestoru medzi viacerými sieťami?

Transparent router
- jeden router multiplexuje viacero sietí do jednej tak, že tieto o ňom ani nevedia
- táto technika má veľa nevýhod, funguje iba na veľkých priestoroch, na samotný transp. router nefunguje ping, snmp...

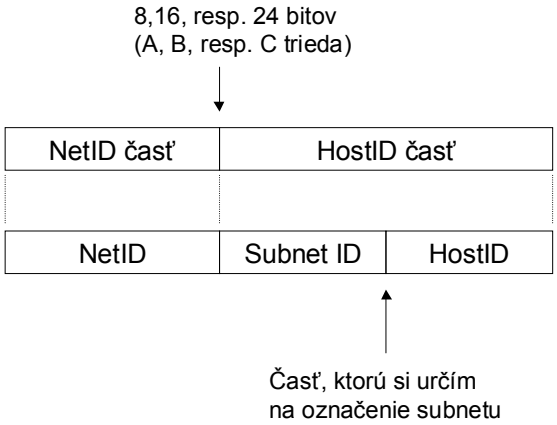
Proxy ARP
- určené na prepojenie dvoch fyzických sietí routrom, pričom obe siete používajú adresy z toho istého adresného priestoru
- má to viacero nevýhod – funguje iba pri ARP, nezahŕňa bezpečnostné pravidlá, adresy hostom aj tak treba prideľovať korektne a spravidla ručne, nedá sa použiť pre zložitejšie topológie
- proxy ARP klame stanice v jednej sieti, že on má IP adresu hosta (ktorý je v druhej sieti) a tak na ARP request odpovedá on – potom je v ARP tabuľke viacero IP adries previazaných s tou istou fyzickou adresou (adresou proxy ARP) – teoreticky všetky IP adresy z druhej siete.
- keby bola na hostoch ochrana proti spoofingu, tak to nefunguje

Subnetting
- technika na rozdelenie jedného adresného priestoru medzi viacero fyzických sietí
- rozšírená technika, lebo bola štandardizovaná

napr. organizácii sa pridelí jeden B adresný priestor, ktorý si môže organizácia vo vnútri rozdeliť na tzv. subnetty, povedzme na 254 adresných priestorov veľkosti C (po 254 hostov)
nasledujúci obrázok znázorňuje takéto rozdelenie jedej siete:



s adresami to vo všeobecnosti vyzerá nasledovne:



Inými slovami povedané, klasické A, B a C priestory prideľuje IANA, s prideleným priestorom si môžem robiť čo chcem, sám si ho rozdeľujem na menšie – ľubovoľne veľké, dokonca môžem vytvoriť aj hierarchické delenie, t.j. v predošlom obrázku by som napr. ešte mohol rozdeliť sieť 172.10.1.0 na viacero podsietí s jemnejšie delenými maskami.

Ako teda implementovať Subnetting?
- pomocou subnet masiek
- je to binárne čislo, ktoré určuje, koľko bitov z IP adresy určuje sieť a podsieť a koľko bitov určuje konkrétny host
- pre typy A, B, C sú imlicitné masky 8, 16 a 24 bitové
- pre C typ je to 11111111 11111111 11111111 00000000, resp. 255.255.255.0
-
- príklad, B priestor je rozdelený na 16 subnetov, t.j. v 3-ťom Bajte vyhradím 4 bity na subnet
- pomocou tohto viem definovať tieto subnety (0000, 0001, 0010 .... 1110, 1111)
- maska je teda 11111111 11111111 11110000 00000000, alebo inými slovami 20-bitová
- adresa hosta môže byť 172.10.53.18 (**10101100 00001010 0011**0101 00010010)
  - hrubé číslice označujú adresu siete (**10101100 00001010 0011**0000 00000000)
  - ostatné určujú číslo hosta 0101 00010010 (1298-my host) v **0011** (tretej) podsieti B priestoru

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**0101 00010010

označuje nultú podsieť alebo celý B priestor, a pri **10101100 00001010 1111**0101 00010010 by nebolo jasné, či broadcast má ísť iba do poslednej podsiete (16-tej v poradí) alebo do celého B priestoru (teda do všetkých podsietí). Samozrejme, v spojení s maskou by to logicky fungovalo, ale dohoda a prax je taká, že sa to **nepoužíva**. (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) host: **10101100 00001010 00110101 00010010**
maska: 11111111 11111111 11110000 00000000

b) host: 172.10.53.18 maska: 255.255.240.0

c) host/maska: 172.10.53.18/20

Classless Inter-Domain Routin (CIDR)

- niekedy označované ako Supernetting
- je zase hra s maskami
- tu ide o spájanie viacerých C priestorov do jedného väčsieho bloku
- samozrejme, C priestory musia ísť za sebou (v binárnom zápise)
- inak povedaná, IANA mi pridelí priestor, povedzme s maskou 20 bitov – tento priestor je ale podľa pôvodnej konvencie z časti začínajúcej 110xxxxx ...., t.j. C-čkové priestory
- v podstate CIDR už nepozerá na pôvodné „maskovanie natvrdo“ podľa A, B a C typu, jednoducho ešte zvyšný priestor prideľuje jemnejšie – presne podľa potreby.
- napr. potrebujem priestor pre 1000 hostov, C-čo mi nestačí (max254), a B-čko je prílišné plytvanie (65 tisíc). 1000 hostov binárne zapíšem do 10-tich bitov (1024), t.j. potrebujem 32-10=**22 bitovú masku.**
- podľa starej konvencie musím spojiť 4 priestory typu C

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, t.j. IP priestory, ktoré sa nevyskytujú v Internete
  - 1 priestor typu A **10.**x.x.x
  - 16 B-čkových priestorov **172.16.**xx až **172.31.**x.x
  - C-čkové priestory **192.168.0.**x až **192.168.255.**x
- s tymito adresami si môžem robiť čo chcem, Internet provideri by ich vôbec nemali routovať von

NAT – network address translation

- statický (one-to-one mapping), dynamický (množina na množinu) a s overloadom (väčšia množina na menšiu, resp. iba na jedinú adresu)

NAT s overloadom (Port address translation)

- to znamená, že napr. pri použití FW alebo HTTP proxy, sa všetky vnútorné adresy prekladajú na jednu, resp. iba niekoľko vonkajších, verejných adries
- takto sa celá moja sieť môže javiť ako jeden host (dosť aktívny:)
- samotná logika rozdeľovania viacerých spojení, napr. na http proxy funguje na rozlišovaní TCP portov – t.j. proxy server vie, ktoré porty sú priradené danej vnútornej IP adrese, a teda hostovi

Multicast

- doteraz sme mali unicast a brodcast
- multicast je určený na označenie určitej skupiny príjemcov
- je to D trieda IP priestorov (začína 1110) – t.j. 28 bitový priestor
- sú definované well-know multicast adresy
  - adresa 224.0.0.1 je určená ako „all hosts“ multicast address
  - 224.0.0.2 – All routers
  - 224.0.0.5 – All OSPF routers

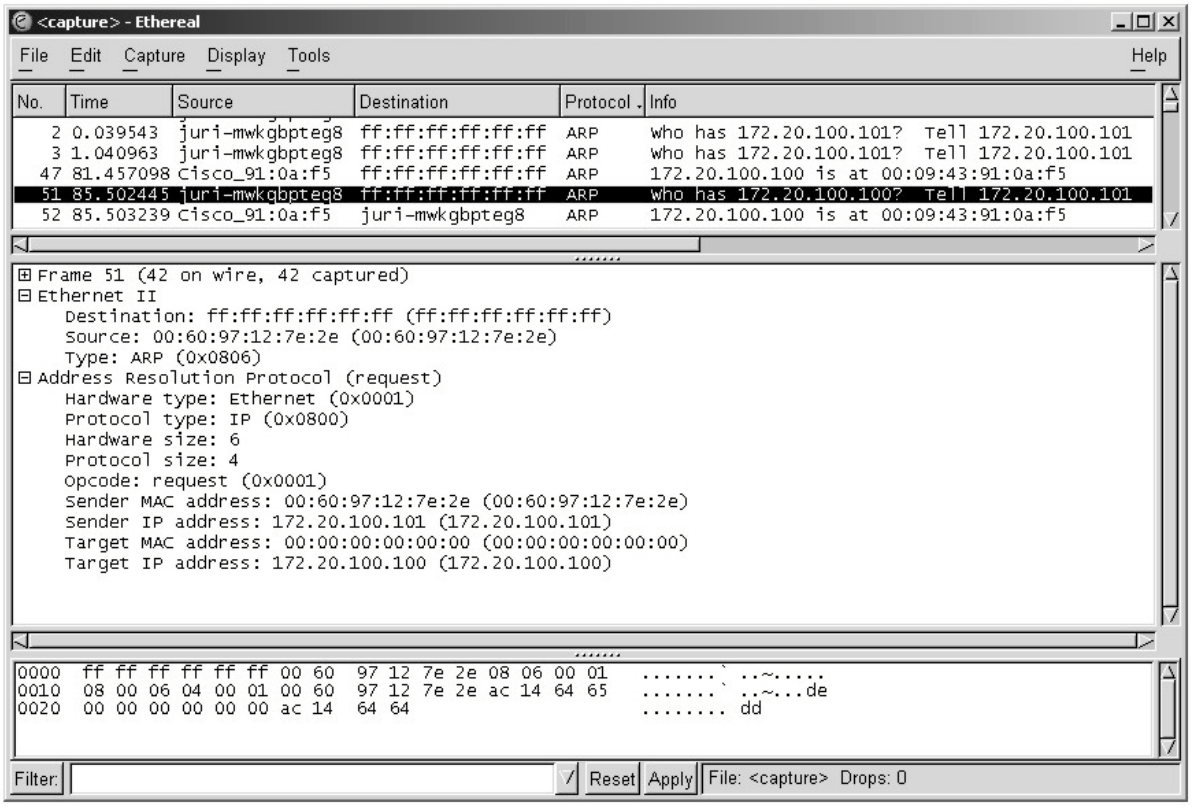
  - 224.0.0.6 -all designated OSPF routers on LAN
  -
- dá sa previazať s Ethernetovým multicastom (posledných 23 bitov z IP adresy sa prilepí k Ethernetovje multicast adrese **01.00.5E**.00.00.00)
  - napr. z 224.0.0.2 sa stane 01.00.5E.00.00.02 (IP multicast adries je dosť, dajú sa vyberať tak, aby mali spodných 23 bitov rôznych)
- v rámci jednej siete teda multicast nie je problém
- 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
- cca každú minútu posielajú na 224.0.0.1 požiadavku všetkým hostom aby oznámili, o aké skupiny (D adresy) majú záujem
- každá stanica si udržuje tabuľku s príslušnosťou ku grupám
- keď router nedostane žiadne odpovede na doteraz používanú grupu, tak ju vytimeoutuje a s ostatnými routrami sa dohodne o zrušení jeho členstva
- 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

ARP protokol

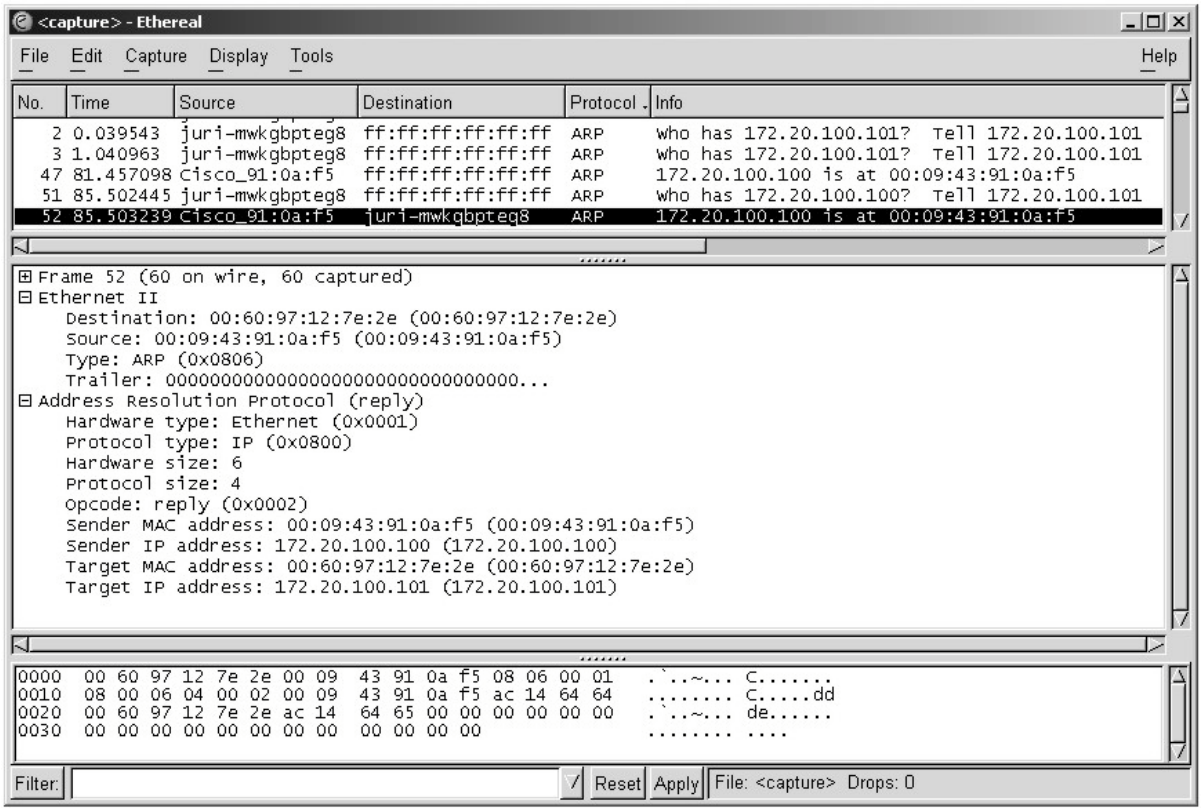
(address resolution protocol)

- mapovanie fyzický adries s adresami vyšších vrstiev môže byť aj priame, t.j. IP adresy môžu byť previazané s fyzickými nejakou funkciou (napr. IP adresa bude vždy podčasťou fyzickej adresy). Toto je ale dosť problematické – napr. pri výmene HW sieťovej karty by bolo potrebné zmeniť aj IP adresu) – tak sa používa dynamické previazanie (dynamic binding) – príkladom je ARP.
- ARP slúži na previazanie fyzických adries a IP (t.j. virtuálnych) adries
- pri LANoch je to previazanie MAC adries s IP adresami
- **hostovi 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)
-
- princíp:
  - broadcastom pošle request na všetkých s otázkou “Kto má IP adresu XY?”
  - následne dostane odpoveď – “Ja mám IP adresu XY”
  - bavia sa samozrejme na základe MAC adries – pozri obrázok:

ARP request



ARP reply



- samozrejme, toto sa nerobí pred každým vysielaním paketu na danú IP
- každý host si drží cashe – jednotlivé položky po určitý čas (timeouts)
- keď niekomu odpovie, tak si aj jeho pridá do cashe (lebo s ním asi niečo bude chcieť)
- rovnako si ostatní môžu updatovať tabuľky na základe info z broadcast requestov
- aj z obrázkov vidno, že ARP je implementované priamo nad linkovou vrstvou, v Ethernet vII rámci má type 0806 hexa, ostatné polia sú tiež zrejmé z obrázku
-

RARP protokol
(reverse address resolution protocol)

- pre diskless stanice na zistenie ich IP adresy
- takáto stanica pri bootovaní pošle RARP request broadcastom s otázkou “mám takúto MAC adresu. Akú je moja IP adresa”
- v sieti fungujú RARP servery (kvôli redundancii aj viacero), ktoré to vedia (majú tabuľky MAC-IP) a aj odpovedia danej stanici
- aby zas ale nebolo veľa odpovedí na jeden request, tak RARP server je PRIMARY iba pre určitú skupinu staníc, - ostatné sú BACKUP a odpovedajú iba keď sa stanica pýta druhý krát (asi jej primárny neodpovedal)
- formát správ je rovnaký ako u ARP, RARP request má Operation =3, response=4.

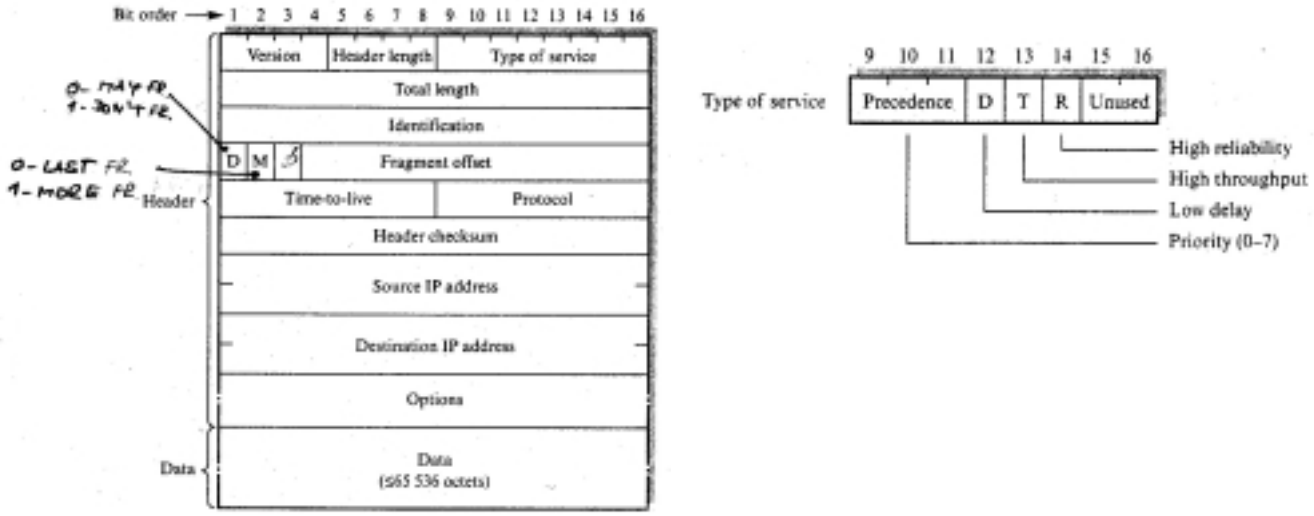
nevýhodou je, že sa používa limited broadcast (samé 1-ky) na dosiahnutie RARP servera – ten teda musí byť na lokálnej sieti (lebo 1-kový broadcast sa neroutuje mimo sieť).
Preto existujem BOOTP protokol (bootstrap protocol), ktorý už používa UDP správy, vie navyše poslať image pre diskless machine, IP default routra a subnet mask.
Na BootP protokole je založený DHCP, ktorý navyše poskytuje automatickú alokáciu adresného priestoru a ďalšie options. Je spätne kompatibilný s BootP clientmi.

IP protokol

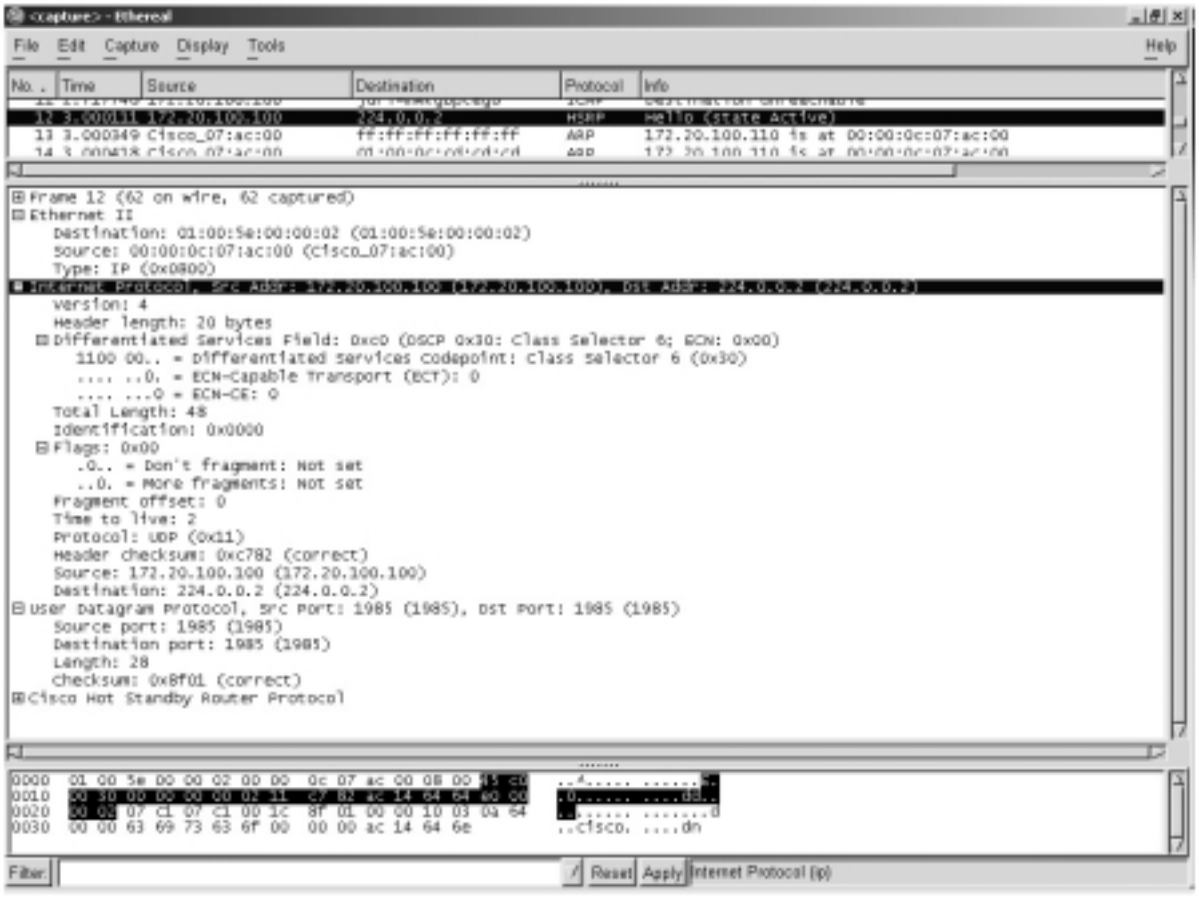
Inernet protocol

- jeden z dvoch hlavných protokolov pri internetworkingu (podľa neho aj názov pre celú rodinu TCP/IP)
- do RM OSI ho možno zaradiť do tretej vrstvy - sieťovej
- tento protokol spája rôzne druhy fyzických sietí a vytvára jednu virtuálnu sieť
- sám o sebe poskytuje iba nespoľahlivú, nespojovanú – connectionless službu s prepájaním paketov
- funčne sa to podobá na poštu – pošle veľa samostatných obálok, ale nezaručí kadiaľ ich doručí, kedy ich doručí, v akom poradí ich doruči a či ich vôbec doručí
- PDU pri IP je IP datagram (alebo IP paket)

formát IP datagramu



príklad IP datagramu



jednotlivé polia:
- verzia – teraz IPv4
- udáva dĺžku hlavičky v 32 bitových slovách (bežný header má 20Byteov – t.j. je tam 5 (na obrázku vyčiernené 45 – verzia 4, a dĺžka hlavičky 5x32bitov = 5x4Bajty = 20Bajtov)
- 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 Differenciated Service Codepoint **DSCP** (používa 6 bitov)

DSCP pole (6 bitov)	Not used

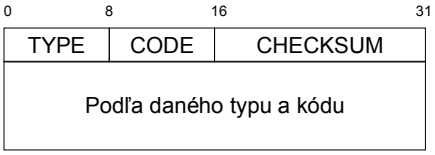
    - xxxxx0 – standard action
    - xxxx11, xxxx01 experimental, local use
    - default hodnota – 000000
    - určuje rôzne triedy kvality služieb – používané hlavne pri traffic shapingu, spätne kompatibilné s IP precedence
    - na cisco
      - 8 tried - xxx**000** class selector – plná spätná kompatibilita 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 padding
    - hlavička môže byť dlhá max. 2^4x32=16x8B=64B, t.j. na options zostáva 59bajtov –rel.málo
    - dá sa definovať, či pri fragmentácii sa options majú kopírovať do každého fragmentu, alebo iba do prvého
    - **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

ICMP protokol

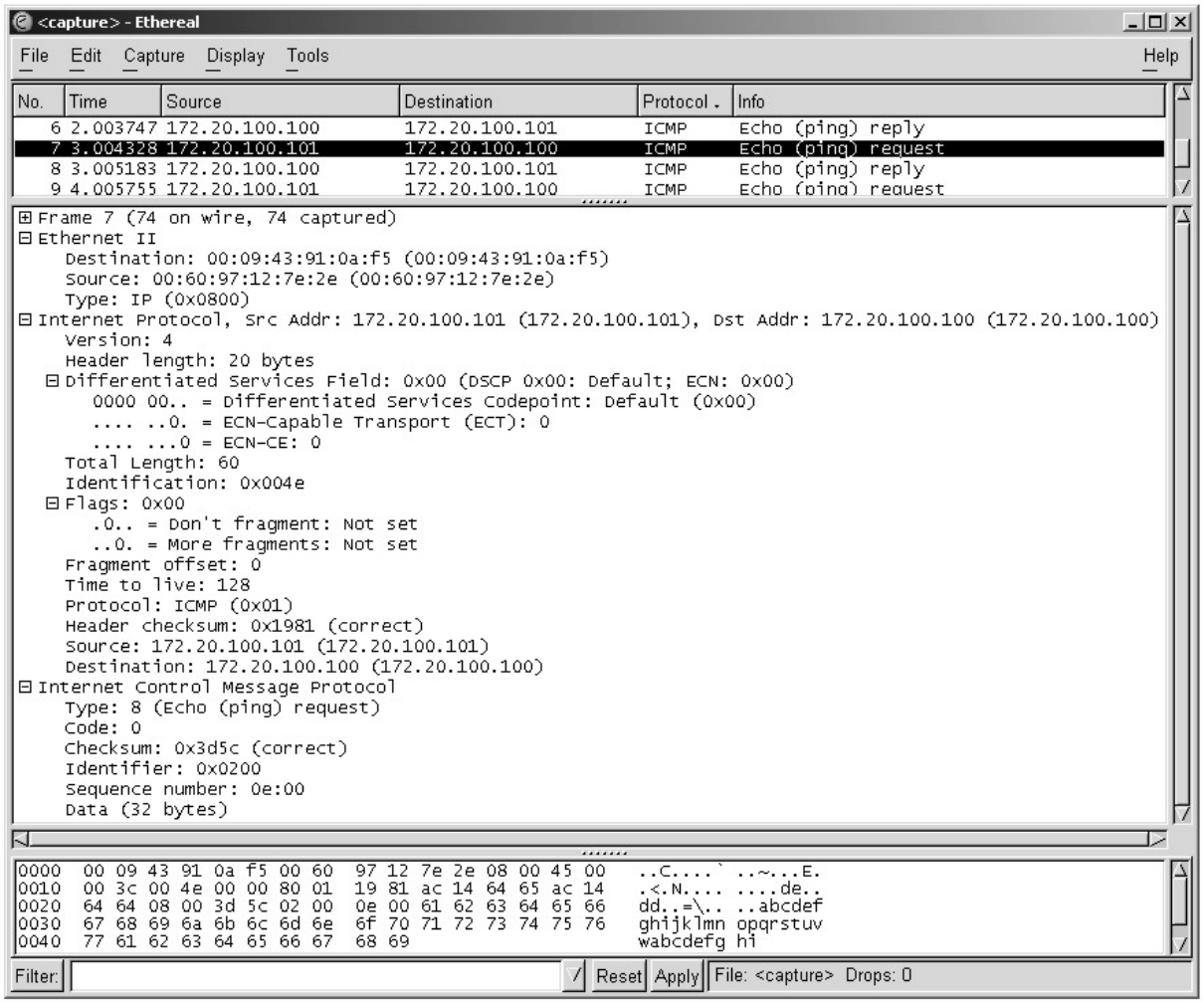
- internet control message protocol
- slúži na posielanie riadiacich správ
- ICMP je encapsulovaný do IP datagramu (Protocol code 0x01)

formát ICMP správy

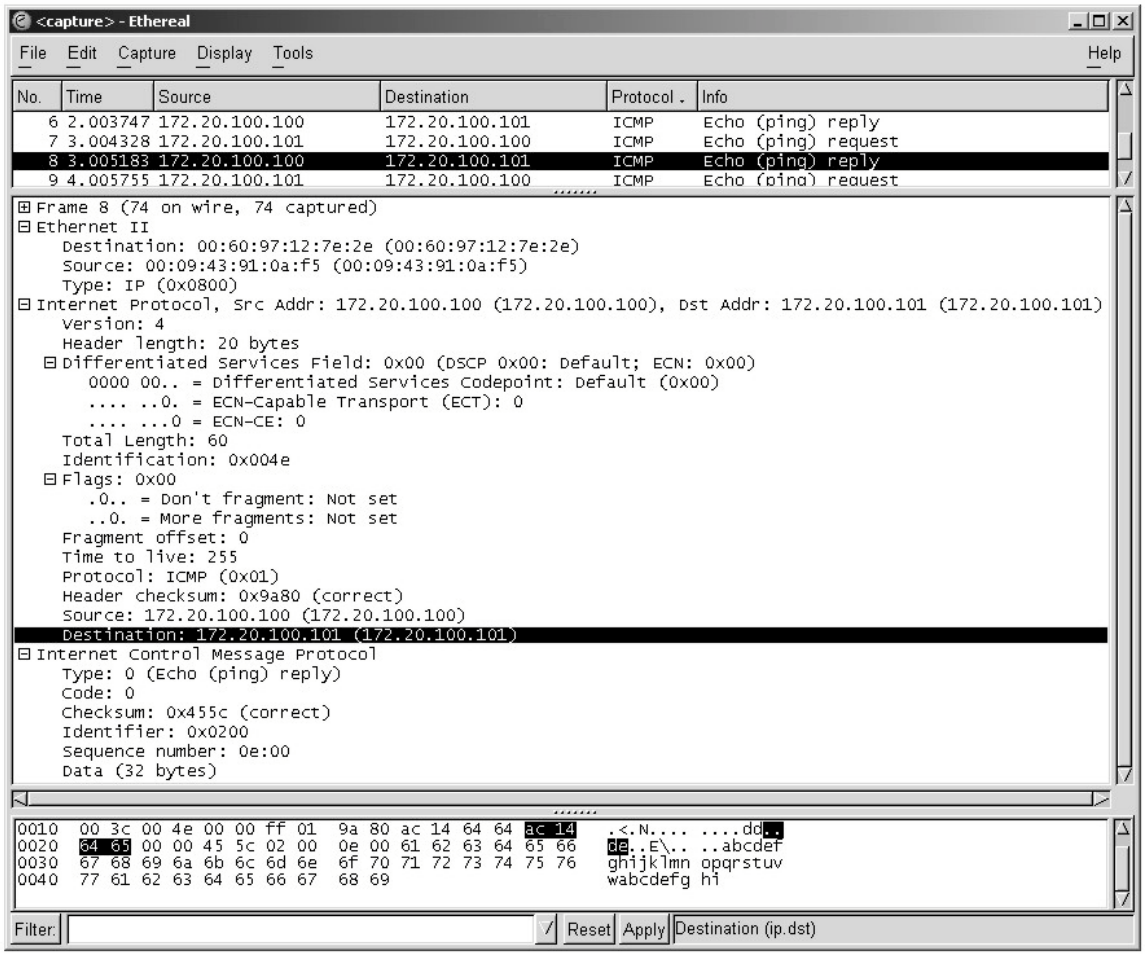


- základné
  - ping - Echo request (type=8), Echo reply (type=0)
  - Destination unreachable (type=3)
    - network unreachable (code=0)
    - host unreachable (code=1)
    - port unreachable (code=3)
    - fragmentation needed and DF set (code=4)
    - source route failed (code=5).....
  - Redirect (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