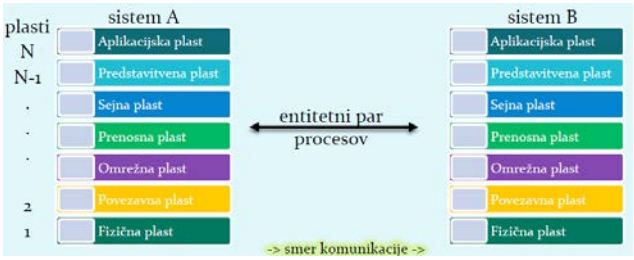
VSEBINA

OSNOVE RAČUNALNIŠKIH KOMUNIKACIJ

1. ISO/OSI model

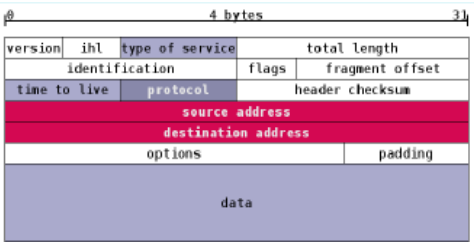
- Plast N nudi storitve (streže) plasti N+1
- Plast N zahteva storitve (odjema) od plasti N-1
- **Protokol:** pravila komuniciranja med istoležnima procesoma
- **Entitetni par:** par procesov, ki komunicira na isti plasti



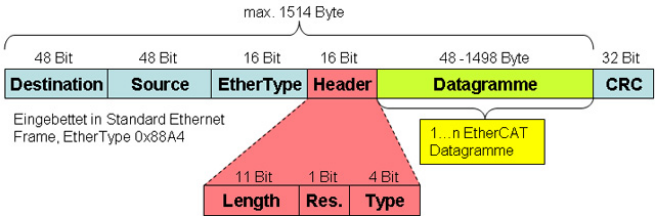
- Vsaka plast ima svoje protokole s katerimi se pogovarja istoležni par entitetnih procesov

OSI plasti: podrobneje

- **APLIKACIJSKA PLAST**
  - Najbližja uporabniku
  - Omogoča interakcijo aplikacije z omrežnimi storitvami
  - **Standardne storitve:** telenet, FTP, SMTP, SNMP, http
- **PREDSTAVITVENA PLAST**
  - Določa **pomen podatkov** med entitetnima paroma aplikacijske plasti
  - Sintaksa in semantika
  - Določa kodiranje, kompresijo podatkov, varnostne mehanizme
- **SEJNA PLAST**
  - Kontrola ''dialoga'' (množice povezav) med aplikacijama
  - **Logično povezovanje med aplikacijami**
  - Običajno vgrajena v aplikacije
- **TRANSPORTNA PLAST** (enota: SEGMENT)
  - **Učinkovit**, **zanesljiv** in **transparenten** prenos podatkov med uporabnikoma. Te storitve zagotavlja višjim plastem
  - **Mehanizmi:** kontrola pretoka, segmentacija, kontrola napak
  - Povezavni, nepovezavni prenosi
  - TCP, UDP, IPSec, GRE, L2TP, PPP
- **OMREŽNA PLAST** (enota: PAKET)
  - **Preklaplanje** (povezavne in nepovezavne storitve)
  - Prenos paketov od izvornega do ciljnega računalnika
  - **Lahko zagotavlja:** zagotovljeno **dostavo**, pravilno **zaporedje**, **fragmentacijo**, izogibanje **zamašitvam**
  - Usmerjanje, usmerjevalniki, usmerjevalni algoritmi
  - **Protokoli:** IP, ICMP, IPsec, IGMP, IPX



- **POVEZAVNA PLAST** (enota: OKVIR)
  - Asihrona/sihrona komunikacija
  - **Fizično naslavljanje:** npr MAC naslov
  - Zaznavanje in odpravljanje napak (pariteta, CRC, checksum)
  - Kontrola pretoka, okvirjanje
  - **Protokoli:** Ethernet, PPP, Frame Relay



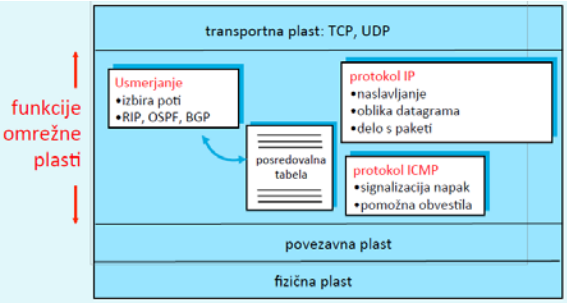
- **FIZIČNA PLAST**

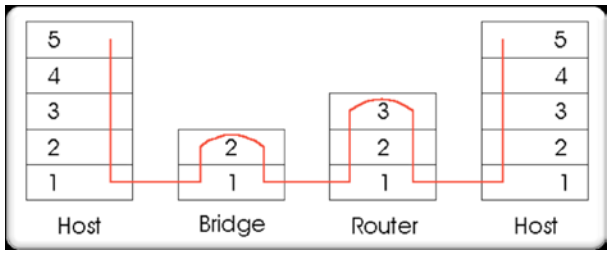
OSI model in model TCP/IP

PRIMERJAVA MODELOV

- **ISO OSI:** Teoretičen, sistematičen, pomankanje implementacij (izdelkov)
- **TCP/IP:** Prilagodljiv, nesistematičen, fleksibilen, veliko izdelkov

2. FUNKCIJE OMREŽNE PLASTI





- **Usmerjevalnik (router):**
  - Naprava, ki deluje na **OMREŽNI** plasti
  - Vzdržujejo **usmerjevalne tabele**, izvajajo **usmerjevalne algoritme**
- **Stikalo (switch)**
  - Naprava, ki deluje na **POVEZAVNI** plasti
  - Vzdržujejo table za preklapljanje, izvajajo filtriranje in odkrivanje omrežja
- **Hub:**
  - Naprava, ki deluje na **FIZIČNI** plasti, danes niso več v rabi

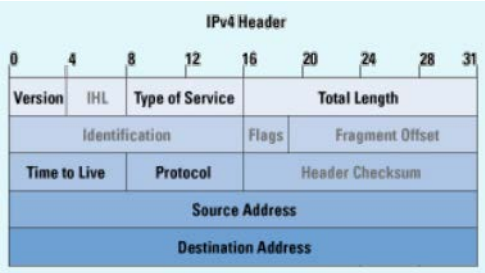
IPv4

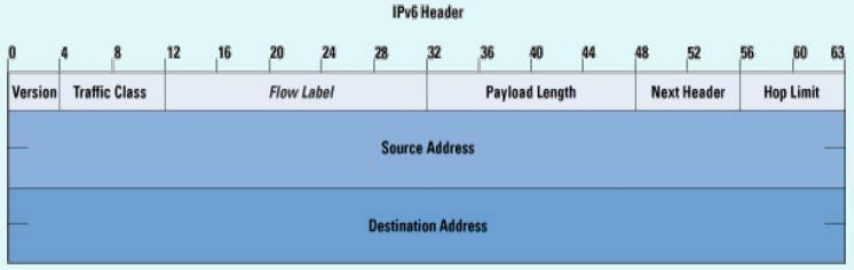
- **IPv4** naslov je **32 bitni** naslov vmesnika
- **Podomrežje:** Je množica IP naslovov, ki so med seboj dosegljivi brez posredovanja usmerjevalnika. Maska (32 bitov) določa del IP naslova, ki predstavlja naslov podomrežja.

IPv6

- Večji naslovni prostor: **128 bitov (Sestavljen iz 64b za ID podomrežja + 64b za ID vmesnika)**
- **Hitro usmerjanje** in **posredovanje** ter **QoS** omogoča že format glave, **fragmentacije ni**
- Implementacija **IPSec** znotraj IPv6 obvezna

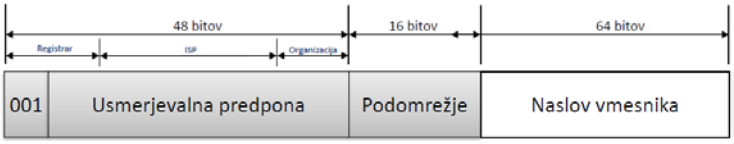
PRIMERJAVA IPv4 in IPv6





3. IPv6

- **NAČINI NASLAVLJANJA**
  - **Unicast:** Naslavljanje posameznega omrežnega vmesnika



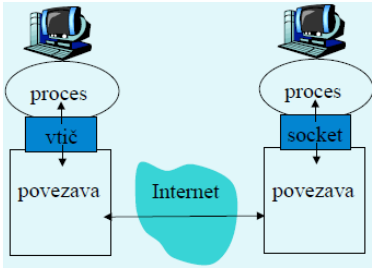
  - **Multicast:** Naslavljanje skupine omrežnih vmesnikov, dostava vsem vmesnikom v množici
  - **Anycast:** Je naslov množice vmesnikov, dostava se izvede enemu vmesniku iz te množice
  - **Broadcast:** V IPv6 ga ni več

IPv6 v omrežjih IPv4

- **Dual-stack:** usmerjevalniki poznajo IPv4 in IPv6. Z zmožnimi govori IPv6, z ostalimi pa IPv4
- **Tuneliranje:** IPv66 paket zapakiramo v enega ali več IPv4 paketov kot podatke

4. FUNKCIJE TRANSPORTNE PLASTI

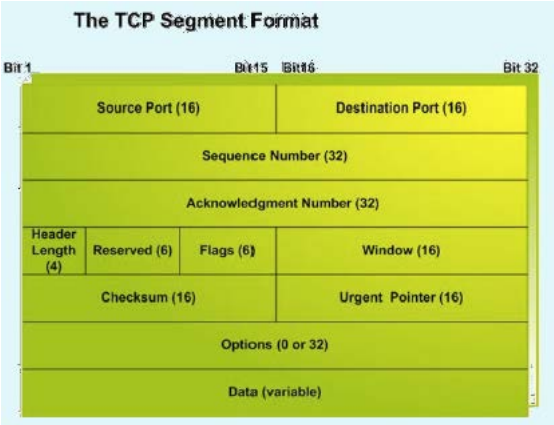
- **NALOGA:**
  - Sprejem sporočila od aplikacije
  - Sestavljanje segmentov v sporočilo za omrežno plast
  - Predaja aplikacijski plasti
- **VTIČ**
  - Vmesnik med **transportno in aplikacijsko plastjo**
  - Proces naslovimo z **IP številko** in **številko vrat**

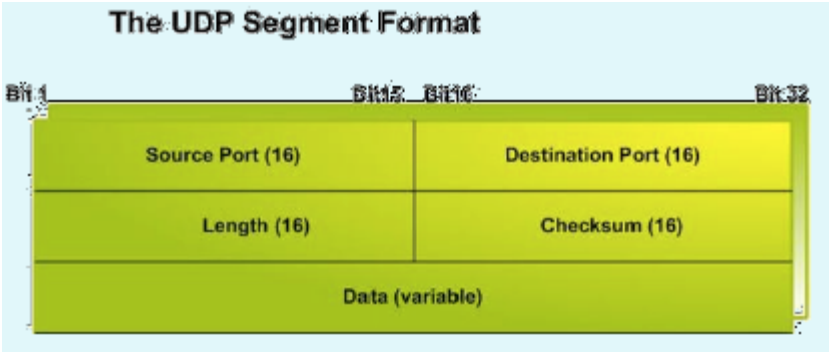


POVEZAVNA IN NEPOVEZAVNA KOMUNIKACIJA

- **Povezavna in nepovezavna komunikacija**
  - TCP in UDP ter ostali protokoli
  - Vzpostavitev, **prenos**, podiranje – povezave
- **Potrjevanje**
  - **V protokolu** (TCP)
  - **V aplikaciji** (UDP)
  - **Neposredno** (ACK in NACK)
  - **Posredno** (samo ACK, sklepamo na podlagi številk paketov)
  - **Tekoče pošiljanje:** ni potrditev

TCP in UDP segmenta





5. PRIMER KOMUNIKACIJE: SPLETNO BRSKANJE

DHCP

- Računalnik ob priklopu na omrežje potrebuje **IP naslov** in podatke prehoda ter DN strežnika: uporabi torej **DHCP**
- Zahteva DHCP se **enkapsulira:** UDP -> IP -> 802.1 Ethernet
- **Ethernet okvir se razpošlje** (broadcast) na omrežje, prejme ga usmerjevalnik, ki opravlja nalogo **DHCP** strežnika
- DHCP strežnik **prebere vsebino DHCP zahteve**
- DHCP strežnik odgovori klientu s paketom **DHCP ACK**, ki vsebuje **njegov IP naslov ter naslove prehoda in DNS strežnika**
- Odgovor **enkapsulira** DHCP strežnik in ga posreduje klientu, ki ga **dekapsulira**
- **DHCP klient dobi odgovor DHCP ACK** (Rezultat: Klient je pripravljen na komunikacijo)

DNS

- Pred pošiljanjem zahtevka **HTTP**, potrebujemo **IP** naslov strežnika www.google.com: **uporabi DNS**
- Enkapsulacija zahtevka **DNS:** (UDP -> IP -> Ethernet). Potrebujemo MAC naslov usmerjevalnika: **uporabi ARP**
- Razpošljemo **zahtevek ARP**, usmerjevalnik odgovori z **ARP odgovorom,** ki hrani njegov MAC naslov
- Klient sedaj pozna MAC naslov prehoda, ki mu lahko **pošlje DNS zahtevek**
- **IP datagram z zahtevkom DNS** se posreduje usmerjevalniku
- IP datagram se posreduje **DNS strežniku**, ki je v omrežju ponudnika (z uporabo usmerjevalnih protokolov)
- DNS strežnik **dekapsulira** zahtevek in posreduje uporabniku IP naslov spletnega strežnika www.google.com

HTTP

- Za pošiljanje **HTTP zahtevka**, klient najprej naslovi **TCP vtič** spletnega strežnika
- **TCP SYN** segment se preko omrežja usmeri do spletnega strežnika
- Spletni strežnik odgovori s **TCP SYNACK** (potrditev rokovanja)
- Sedaj je **TCP povezava vzpostavljena**
- **HTTP zahtevek** se pošlje na **TCP vtič** spletnega strežnika
- **IP datagram**, ki vsebuje spletno zahtevo po strani www.google.com se usmeri k spletnemu strežniku
- Spletni strežnik odgovori s **HTTP REPLY,** ki vsebuje vsebino strani
- IP datagram s stranjo se usmeri h klientu
- **WWW stran je prikazana!**

6. OMREŽNA VARNOST

- **JE PODROČJE KI:**
    - Analizira možnosti vdorov v sisteme
    - Načrtuje tehnike obrambe pred napadi
    - Snuje varne arhitekture, ki so odporne pred vdori
- **INTERNET NI BIL SNOVAN OZIRAJOČ SE NA VARNOST**

KAKO LAHKO VDIRALEC ŠKODUJE SISTEMU

- **Prisluškovanje**: Prestrezanje sporočil
- Aktivno **ponarejanje** sporočil v neki komunikaciji
- **Kraja identitete (impersonacija):** Ponaredi lahko izvorni naslov ali poljubno drugo vsebino paketa
- **Prevzem povezave (hijacking):** Odstrani pravega pošiljatelja ali prejemnika iz komunikacije in prevzame njegovo vlogo
- **Zavrnitev storitve (denial of service):** Onemogoči uporabo regularne storitve (npr. lahko jo preobremeni)

ELEMENTI VARNE KOMUNIKACIJE

- **Zaupnost –** kdo sme prebrati? (enkripcija)
- **Avtentikacija –** dokaži, da si res ti (identifikacijija – povej, kdo si, brez dokaza)
- **Razpoložljivost in nadzor dostopa –** preprečevanje nelegitimne rabe virov
- **Integriteta sporočila –** je bilo med prenosom kaj spremenjeno?
- **Preprečevanje zanikanja** (nonrepodiation) – res si poslal / res si prejel

V PRAKSI:

- Požarni zidovi, sistemi za zaznavo vdorov (intrusion detection)
- Varnost na aplikacijski, transportni, omrežni in povezavni plasti

AVTENTIKACIJA

- Prepričamo se o dejanski identiteti osebe – sogovornika v komunikaciji
- **PRISTOPI**
  - Challenge-response (izziv-odgovor)
  - Zaupamo tretji strani
  - Avtentikacija s sistemom javnih ključev

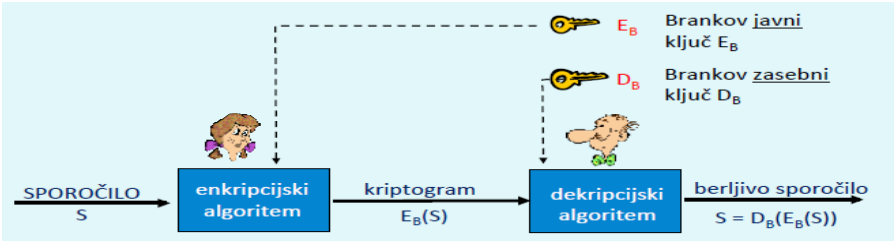
ZAUPNOST

- Je način obrambe pred **pasivnimi** vdiralci (prisluškovalci) in **aktivnimi** vdiralci (ponarejevalci).
- Sporočilo **P kriptiramo** s ključem **E()** – dobimo **kriptogram E(P).** Kriptogram **E(P)** predelamo v izvorno obliko s ključem **D()**, dobimo izvorno sporočilo **D(E(P))=P**
- **VRSTE METOD**
  - **Substitucijske** (menjava znakov) / **transpozicijske** (vrstni red znakov) **NEZANESLJIVE**
  - **Simetrične** (**E=D**, npr. DES, AES) / **asimetrične** (**E!=D,** npr. RSA, ECC)

VRSTE KRIPTOGRAFIJE

- **Kriptografija uporablja ključe**
  - **Kriptirani algoritem je običajno znan vsem**
  - **Tajni so le ključi**
  - **Kriptiranje:** skrivanje vsebine
  - Kriptoanalize (''razbijanje'' kode)
- **Asimetrična kriptografija**
  - Uporablja dva ključa: javnega in zasebnega
- **Simetrična kriptografija**
  - Uporablja samo en ključ

KRIPTOGRAFIJA Z JAVNIMI KLJUČI



INTEGRITETA

- **INTEGRITETA UPORABNIKOV:** Dokazuje, **kdo je sporočilo poslal** in da **sporočilo bere le pravi prejemnik**. Sporočilo S, ki ga uporabnik A pošlje uporabniku B kriptiramo

$$EB(DA(S)) = XXX$$
$$S = D_B(E_A(XXX))$$

- **INTEGRITETA SPOROČILA:** Dokazuj, da sporočilo ni bilo spremenjeno. Uporabljajo se zgoščevalne funkcije, ki izračunajo zgoščeno vrednost sporočila **Z(S)**. To vrednost podpišemo z mehanizmom elektronskega podpisa.

PRIKLOP IN ZAGON NAPRAVE

7. ZAGON

- CPE ob priklopu na napajanje nastavi vrednost **PŠ** (programskega števca) na točno določeno vrednost
- Za tem začne izvajati ukaze
  - Sledi običajno delovanje
- **POMEMBNO:** Kaj se nahaja v pomnilniku na mestu, kjer prične z delom CPE

8. BIOS (Basic I/O System)

- Sestavljen je iz dveh sklopov:
  - **Koda**, ki se prične izvajati ob zagonu (Začetek mora biti tam kamor je nastavljen PŠ)
  - **Gonilniki** za V/I enote
- Koda izkoristi gonilnike za dostop do zunanjih enot (trdi ali mehki disk, CD, …) in z njih **naloži** (poseben) **program**, ki mu rečemo **operacijski sistem**
- S tem je strojna oprema zbootana oz. ''Obuta''

9. OPERACIJSKI SISTEM – KLASIČNO

- Operacijski sistem (OS) je vmesnik med uporabniškimi programi in strojno opremo ter **skrbi za upravljanje z viri** (V/I enote, datoteke, procesorski čas, …)
- Prvotno je OS izkoriščal gonilnike iz BIOS-a za delo z V/I enotami
- Slednji so imeli dve pomanjkljivosti:
  - Niso bili prijazni
  - Niso bili učinkoviti
- OS je pričel uporabljati svoje gonilnike

10. NALAGANJE OS – SODOBNO

- **BIOS naloži nek program, ki ga nato začne izvajati**
- Vedno ga najde na **prvem bloku V/I enote** – master boot record, MBR
- Naloženi program ni nujno, da je OS, ampak lahko naloži nek program, ki šele potem naloži OS ali kaj podobnega
  - Tako lahko nalagamo **enega izmed večih nameščenih OS-ov** (Ni nujno da je na prvem bloku)

11. NALAGANJE PROGRAMA IZ OMREŽJA

- BIOS lahko naloži program namesto z diska, **s strežnika na omrežju**
- Potrebujemo **definicijo komunikacije** računalnika s strežnikom – **potrebujemo protokol**
- **PREDNOSTI:**
  - Ne potrebujemo diska na računalniku
  - OS preprosto zamenjamo za vse računalnike, saj ga zamenjamo samo na strežniku
- **SLABOSTI:**
  - Ranljivost
  - Počasnost
  - Varnost?

12. DNS – Domain Name Service

- www.fri.uni-lj.si = 212.235.188.25
- Storitev **DNS** **preslikuje med črkovnim nizom in številko**
  - Namesto DNS storitve lahko uporabimo preslikovalno tabelo v datoteki /etc/hosts
- **Kako najdemo strežnik DNS storitve?**
  - Lahko nam ga priskrbi DHCP
  - Ali pa moramo njegov IP podati sami
- **Kako strežnik DNS storitve najde druge strežnike DNS?**
  - Poznati mora njihove IP naslove
  - Datoteka: /etc/namedb/named.root
- DNS storitev uporablja vrata številka **53**
- Nimamo storitve, ki bi preslikovala med imenom **DNS in 53**
  - Imamo **preslikovalno tabelo** v datoteki /etc/services
- **Izziv: kako se v resnici imenuje DNS storitev v omenjeni tabeli?**
  - Imenuje se Domain
- **DNS** protokol upodablja UDP pakete
- V glavi paketa označimo, da gre za UDP paket s številko **17**
- **Nimamo storitve**, ki bi preslikovala med imenom **UDP in 17**
  - Zato imamo preslikovalno tabelo v datoteki /etc/protocols
- **Izziv: kateri protokol ima številko 50 in za kaj se uporablja? Kakšni so formati vseh treh etc datotek?**
  - **Encapsulating Security Payload**: priskrbi varnostne storitve pri IPv4 in IPv6
  - Vse 3 datoteke so C formata

13. IANA – The Internet Assigned Numbers Authority

- Svetovni dogovor o številkah
- Številke hrani in oglaša IANA
- **Korenski DNS strežniki:**
  - www.iana.org/domains/root/db/arpa.html
- **Vrata:**
  - www.iana.org/domains/root/db/arpa.html
- **Protokoli:**
  - www.iana.org/domains/root/db/arpa.html
- **izziv: kakšni podatki so na: www.iana.org/domains/root/db/si.html?**
  - Podatki o slovenski DNS strežnikih

14. NALAGANJE OS IZ OMREŽJA

- Ob zagonu računalnik **lahko** ali **ne pozna nekatere svoje podatke**: ime, IP, …
- Mora znati govoriti protokol, ki bo omogočal nalaganje OS
  - Rokovalnik protokola, ki mora biti jedrnat zaradi omejitve prostora v BIOS-u

KORAKI NALAGANJA

RAČUNALNIK MORA:

1. Znati poiskati strežnik, s katerega bo naložil OS
2. Znati se nastaviti, kot bo svetoval/zahteval strežnik
3. Prenesti OS k sebi
4. Namestiti OS in ga zagnati

- **Načrtovalska odločitev:**
  - Koraka 1. in 2. v enem protokolu (**bootp**) in korak 3. v drugem protokolu (**npr. tftp**)

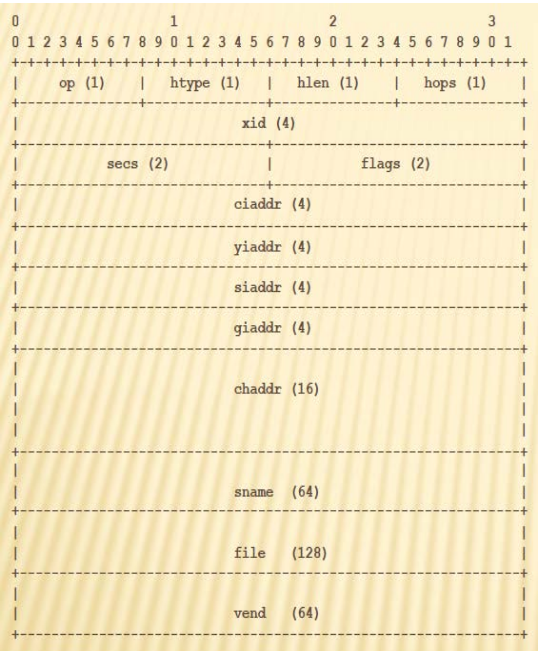
15. BOOTP – Bootstrap Protocol

- Koračni pogovor med odjemalcem in strežnikom: odjemalec vpraša in strežnik odgovori
- Lahko je hkrati prisotnih več strežnikov in lahko hkrati več odjemalcev želi naložiti OS

Nekaj podrobnosti:

- Odjemalec na začetku **ne pozna IP naslova strežnika**, zato razpošlje **(broadcast)** na **2. plasti na lokalni mreži** z željo po nalaganju OS
- Strežnik dodeli odjemalcu IP naslov (ali pa ne) ter **mu sporoči, kje se nahaja odjemalčev OS**
  - Ni nujno da, na lokalni mreži
- Bootp je aplikacija, ki na prenosni plasti uporablja nepovezavni način – **UDP protokol**
- **Izziv: kako je z varnostjo in trojanskimi konji?**
- **Zagotovljena ni nobena varnost!!** Saj uporablja UDP in IP ki sama po sebi ne zagotavljata nobene varnosti. Lahko bi uporabili **Encapsulating Security Payload**: priskrbi varnostne storitve pri IPv4 in IPv6. Vendar bi moral sam BIOS biti večji.

PAKET BOOTP



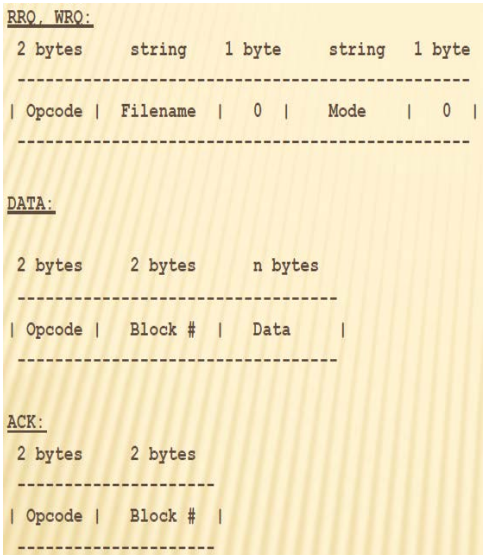
- **op**: zahteva ali odgovor
- **htype**: vrsta medija
- **hlen**: dolžina naslova
- **chaddr**: odjemalčev naslov plasti 2
- **hops**: število skokov
- **xid**: id zahteve (random št.), da poveže zahtevo z odzivom
- **secs**: koliko časa je minilo od prvega pošiljanja
- **flags**: zastavice – samo razpošiljanje ali ne
- **ciaddr**: odjemalčev naslov
- **yiaddr**: nastavljen naslov
- **siaddr**: strežnikov naslov
- **giaddr**: naslov prehoda
- **sname**: ime strežnika z OS
- **file**: datoteka z OS
- **vend**: možne razširitve

16. PROTOKOL TFTP

- **Zelo poenostavljena funkcionalnost FTP protokola** – ohranjena predvsem možnost prenosa podatkov
- Ni izpisa imenika, avtentikacije in kriptiranja, dovoljuje zelo velike pakete, ne more naložiti datoteke večje od 1 TB
- Odjemalec na začetku **pozna IP naslov strežnika**, saj ga dobi preko **bootp protokola**
- **TFTP** je aplikacija, ki na **prenosni plasti** uporablja **nepovezavni način** – UDP protokol
- **Izziv: tako bootp kot tftp uporabljata UDP protokol. Zakaj?**
  - Zato, da je nujna implementacija lažja saj ju uporablja BIOS, katerega pomnilnik je majhen. Zato ne implementiramo TCP saj bi rabili več prostora.

PRIMER POGOVORA OB BRANJU

1. Odjemalec pošlje zahtevo po branju (RRQ)
2. Strežnik **odgovori z DATA paketom in podatki**, ki jih je zahteval odjemalec. Poslani so z novih vrat in vsa komunikacija z odjemalcem mora odslej potekati preko teh vrat (NAT prehod?)
   - Pri NAT prehodu moramo ročno konfigurirati usmerjevalnik (port forwarding)
3. Na vsak paket podatkov odjemalec **odgovori z ACK paketom**, nakar strežnik pošlje naslednji paket (prejšnja točka) – če potrditve ni v določenem času, strežnik ponovno pošlje paket
4. Posebnost je zadnji paket, ki je manjši od največje dovoljene velikosti **(Tako vemo kdaj se pošiljanje konča)**



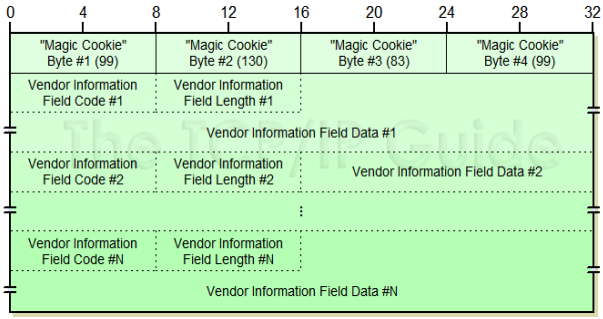
- **Opcode:** zahteva
- **Filename 0:** ime datoteke
- **Mode 0:** oblika zapisa podatkov
- **Block #:** številčenje poslanih paketov

17. PRIKLOP NA OMREŽJE

- Nekateri računalniki imajo svoj disk in si sami naložijo OS, vendar se želijo priključiti v omrežje:
  - Stalna IP številka deluje **samo pri stacionarnih** računalnikih
  - Mobilni računalniki potrebujejo vsakič drugo številko
  - Ponudniki želijo poslužiti **več strank**, kot imajo IP naslovov
- Uporabimo lahko **BOOTP**. V prvem koraku odjemalcu pošlje tudi podatke za nastavitev IP naslova in nastavitev IP naslova prehoda
- Poleg IP naslova, potrebujemo še **naslov prehoda**, **naslov DNS strežnika**, **naslov vmesnega (proxy) strežnika**, …
  - Uporabimo / spremenimo namen polja **vend** v BOOTP protokolu, da v njem posredujemo naslove

RAZŠIRITVE VEND

- Prve 4 bajte imenujemo **Magic Cookie** z vrednostjo 99.130.83.99, ki prejemniku pove kako naj interpretira naslednje podatke



- **Tag:** 1 – 127 predefinirani, povejo kakšni so naslednji podatki:
  - DNS IP, time, gateway IP,.. ipd.
- **Tag:** 128 – 254 nedefinirani, lahko dodamo lastno razširitev
- **Length:** pove dolžino podatkov
- **VIFData:** podatki npr. cas, IP..

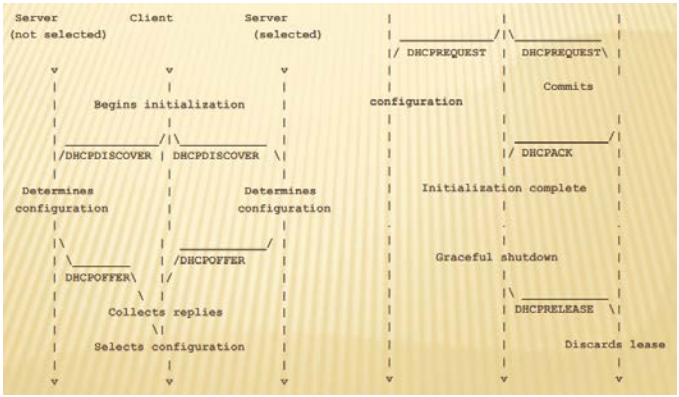
18. PROTOKOL DHCP

- Bazira na podlagi **BOOTP** protokola, doda možnost avtomatskega alociranja IP naslovov
- Odjemalec na začetku **ne pozna IP naslova strežnika**
- DHCP je aplikacija, ki na prenosni plasti uporablja **nepovezavni način** – UDP protokol
- **VARNOST:**
  - Ker je zgrajen na IP in UDP sam po sebi ne zagotavlja nobene varnosti
  - **Napad:** Napadalec lahko postavi **zlonameren DHCP strežnik**, ki na zahteve odgovori z napačnimi ali dupliciranimi IP naslovi
  - **Napad:** Neavtorizirani odjemalci pridobijo dostop do virov, do katerih nebi smeli
  - **Napad:** Izpraznjenje virov s strani neavtoriziranih odjemalcev
  - **Varnost bi lahko zagotovili z ključi (avtentikacijo)..** Vendar tega ne delamo, ker je nepraktično

DHCP JEDRO

- **Osnovna ideja:** odjemalec dobi na uporabo IP naslov za določen čas
- **MOŽNE ZAHTEVE**:
  - **DHCPDISCOVER:** iskanje strežnika
  - **DHCPOFFER:** ponudba odjemalcu
  - **DHCPREQUEST:** odjemalec potrjuje prejete nastavitve; tudi želja po podaljšanju sposoje IP naslova
  - **DHCPACK, DHCPNAK**: strežnikova potrditev/zanikanje odjemalcu
  - **DHCPDECLINE:** odjemalec strežniku, da je IP naslov že v uporabi
  - **DHCPRELEASE:** odjemalec vrača naslov pred potekom
  - **DHCPINFORM:** odjemalec želi samo ostale podatke, naslov že ima

DHCP ŽIVLJENSKI CIKEL



19. DHCPv6 PROTOKOL

- Povsem drugačen protokol za IPv6
- **Dva načina** konfiguracije računalnika:
  - **Brezstanjsko (stateless),** kjer se računalnik lahko sam nastavi
  - **Stanjsko (statefull)**, kjer računalnik nastavi s pomočjo drugih enot
- Odjemalec na začetku **ne pozna** IP naslova strežnika
- DHCP je aplikacija, ki na prenosni plasti uporablja **nepovezavni način – UDP protokol**

JEDRO PROTOKOLA

- **Možne zahteve (msg-type):**
  - **SOLICIT:** prošnja za nastavitev
  - **ADVERTISE:** oglašanje naslova
  - **REQUEST**: zahteva za nastavitvene parametre
  - **CONFIRM:** preverjanje, ali je naslov, ki ga je dobil odjemalec, še vedno v redu
  - **RENEW:** zahteva za obnovitev
  - **REBIND:** zahteva za ohranitev
  - **REPLY:** odgovor odjemalcu
  - **RELEASE:** sprostitev naslova
  - **DECLINE:** zavrnitev dodeljenega naslova
  - **RECONFIGURE**: strežnik odjemalcu sporoča, naj obnovi nastavitve
  - **INFORMATION-REQUEST:** zahteva za nastavitve brez IP naslova
  - **RELAY-FORW**: prepošiljanje
  - **RELAY-REPL**: potrdilo prepošiljatelju, ki vsebuje odgovor odjemalcu

NADZOR IN UPRAVLJANJE Z OMREŽJI

1. UPRAVLJANJE Z OMREŽJEM

- Z rastjo interneta in lokalnih omrežij so se majhna omrežja povezala v **VELIKO** infrastrukturo. Zato je s tem narasla tudi potreba po **SISTEMATIČNEM** upravljanju strojnih in programskih komponent tega sistema.
- **Pogosta vprašanja:**
  - Kateri viri so na razpolago v omrežju?
  - Koliko prometa gre skozi določeno omrežno opremo?
  - Kdo uporablja omrežne povezave, zaradi katerih direktor prepočasi dobiva elektronsko pošto?
  - Zakaj ne morem pošiljati podatkov določenemu računalniku?
- **Definicija:** Upravljanje z omrežjem vključuje **vpeljavo**, **integracijo** in **koordinacijo** s **strojno opremo**, **programsko opremo** in **človeškimi viri** z namenom **opazovanja**, **testiranja**, **konfiguriranja**, **analiziranja** in **nadzorovanja** omrežnih virov, pri katerih želimo zagotoviti delovanje v realnem času (ali delovanje z ustrezno kakovostjo - QoS) za **sprejemljivo ceno**.

2. PRIMERI AKTIVNOSTI UPRAVLJANJA

1. **Zaznavanje napake na vmesniku računalnika ali usmerjevalnika:**
   - Programska oprema lahko sporoči administratorju, da je na vmesniku prišlo do težave (celo preden odpove!)
2. **Nadzorovanje delovanja računalnikov in analiza omrežja**
3. **Nadzorovanje omrežnega prometa:**
   - Administrator lahko opazuje pogoste smeri komunikacij in najde **ozka grla**
4. **Zaznavanje hitrih sprememb v usmerjevalnih tabelah:**
   - Ta pojav lahko opozarja na težave z usmerjanjem ali napako v usmerjevalniku
5. **Nadzorovanje nivoja zagotavljanja storitev:**
   - Ponudniki omrežnih storitev nam lahko jamčijo **razpoložljivost**, **zakasnitev** in **določeno prepustnost storitev**. Administrator lahko meri in preverja.
6. **Zaznavanje vdorov:**
   - Administrator je lahko obveščen, **če določen promet prispe iz sumljivih virov**. Zaznava lahko tudi določen **tip prometa** (npr. množica SYN paketov, namenjena enem samem vmesniku)

PRIMERI:

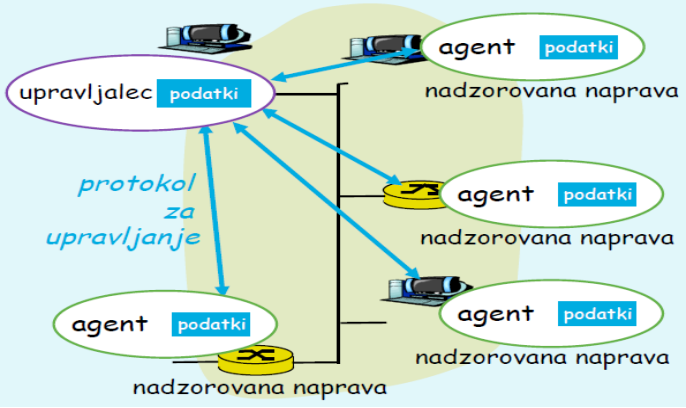
- Nadzorovanje delovanja računalnikov in analiza omrežja **(odkrivanje topologije omrežja)**
- Nadzorovanje omrežnega prometa **(profiliranje)**
- Nadzorovanje nivoja zagotavljanja storitev **(pretok podatkov)**
- Nadzorovanje delovanja računalnikov in analiza omrežja **(popis IP naslovov, diagnostika in odkrivanje napak)**
- **TOREJ:** kako dobra je povezava med računalniki, če je prepočasna, **kje je problem**, **katera povezava** je problematična? Težave na požarnem zidu? Če je prišlo do vdora, stvar seveda **popraviti** … Vse te stvari bi želeli **opazovati** in **nadzorovati** (opazujemo, kaj se je dogajalo in kaj potrebujemo, da se to odpravi? Potrebujemo **ZAPISE** (log-e), ki podatke zberejo, podatke pregledamo in nato potrebujemo različna orodja za reševanje težav.

PROGRAMSKA OPREMA ZA UPRAVLJANJE

- **CLI (Comand Line Interface)**
  - Natančno upravljanje
  - Možnost rabe ukaznih datotek (batch)
  - Problem poznavanja sintakse, težavnost shranjevanja konfiguracije, manj splošno – specifično za posamezno omrežno opremo
- **GUI (Graphical User Interface) aplikacije:**
- Vizualno lepše
- Omogoča pregled delovanja cele naprave/omrežja
- Uporablja lahko **svoj (zgoščen) protokol** za komunikacijo z napravo **(hitrost)**
- Izgubimo možnost shranjevanja berljive konfiguracije (binarni zapis), lahko maskira vse konfiguracijske možnosti

INFRASTRUKTURA ZA UPRAVLJANJE

- **Komponente sistema za upravljanje:**
  - **Upravljalec** = entiteta (aplikacija + človek), BOSS
  - **Nadzorovana naprava** (vsebuje agenta **NMA** in nadzorovane **OBJEKTE**, ki vsebujejo nadzorovane **PARAMETRE**)
  - **Protokol za upravljanje** (npr. SNMP – Simple Network Management Protocol)



ZGODOVINA: PROTOKOLI ZA UPRAVLJANJE

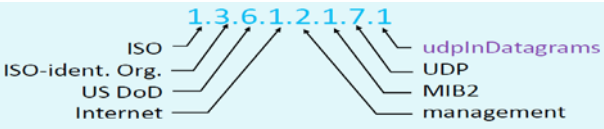
- **OSI CMIP**
  - Common Management Information Protocol
  - Nastal 1980: prvi standard za upravljanje
  - Prepočasi standariziran, ni zaživel v praksi
- **SNMP**
  - Simple Network Management Protocol
  - IETF standard
  - Prva verzija zelo preprosta
  - Hitra uvedba in razširitev v praksi
  - Trenutno: SNMP V3 **(Dodana varnost!)**
  - **De facto** (glavni) standard za upravljanje z omrežji

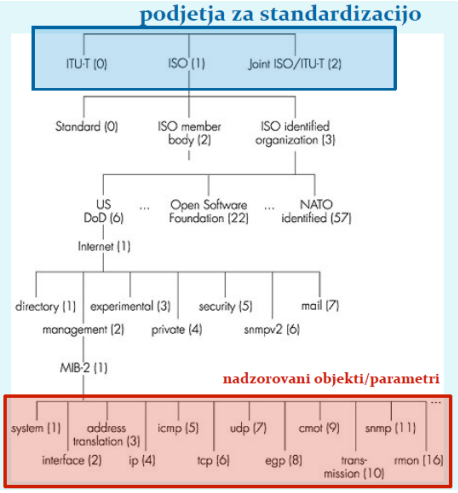
PODATKI ZA UPRAVLJANJE

- Za vsako vrsto nadzorovane naprave imamo svoj **MIB** **(Management Information Base)**, kjer so podatki o upravljanih **OBJEKTIH** in njihovih **PARAMETRIH**.
- Upravljalec ima svoj **MDB (Management Database)**, kjer za vsako upravljano napravo hrani konkretne vrednosti za njihove **MIB objekte/parametre**
- Potreben je jezik, ki definira zapis **OBJEKTOV** in **PARAMETROV**: **SMI (Structure of Management Information)**

3. MIB MODULI: STANDARIZACIJA

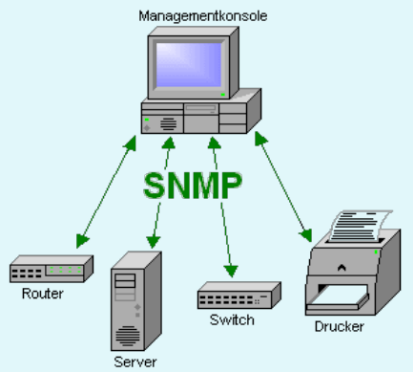
- MODULI:
  - ''Standardizirani''
  - Lastni proizvajalci opreme (vendor-specific)
- **IETF (Internet Engineering Task Force)** zadolžena za standardizacijo MIB modulov za usmerjevalnike, vmesnike in drugo omrežno opremo
  - **Potrebno poimenovanje** (označitev) standardnih komponent)
  - Uporabi se poimenovanje **ISO ASN.1**
- **Hierarhična urejenost objektov z drevesom identifikatorjev**
- Vsak objekt ima ime, **sestavljeno iz zaporedja številčnih identifikatorjev** (ločenih z piko) od korena drevesa do lista
- **PRIMER:**
  - 1.3.6.1.2.1.7 določa protokol UDP
  - 1.3.6.1.2.1.7 določa protokol UDP* določa opazovane parametre UDP protokola

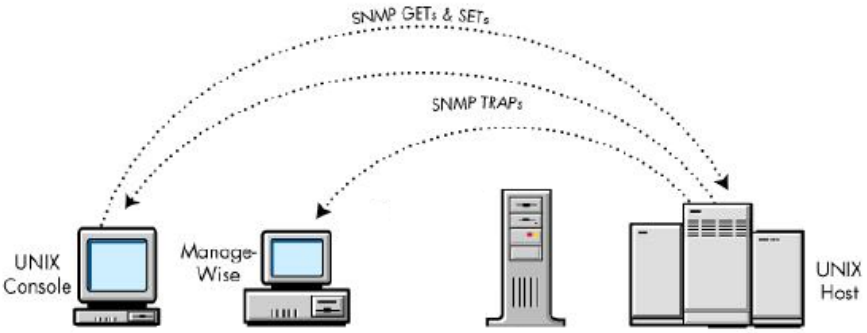




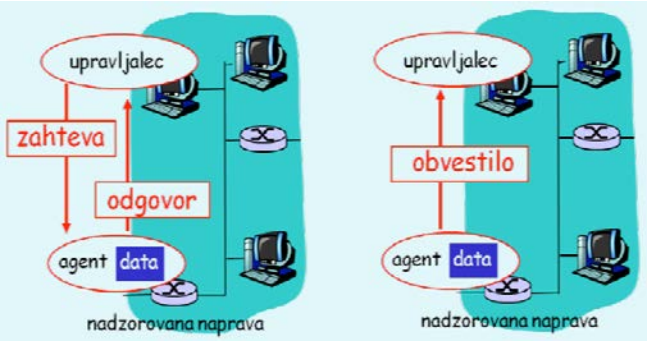
Object ID	Name	Type	Comments
1.3.6.1.2.1.7.1	UDPInDatagrams	Counter32	total # datagrams delivered at this node
1.3.6.1.2.1.7.2	UDPNoPorts	Counter32	# underliverable datagrams no app at portl
1.3.6.1.2.1.7.3	UDInErrors	Counter32	# undeliverable datagrams all other reasons
1.3.6.1.2.1.7.4	UDPOutDatagrams	Counter32	# datagrams sent
1.3.6.1.2.1.7.5	udpTable	SEQUENCE	one entry for each port in use by app, gives port # and IP address

4. PROTOKOL SNMP





- **Simple Network Management Protocol**
- **Protokol za izmenjavo nadzornih informacij** med upravljalcem in nadzorovanimi objekti
- Podatki o nadzorovanih objektih se prenašajo **med nadzorovano opremo in upravljalcem skladno z definicijo MIB**
- **DVA NAČINA DELOVANJA:**
  - **Zahteva – odgovor (request – response):** bere in nastavlja vrednosti (upravljalec zahteva)
  - **Obvestilo (trap message):** naprava obvesti upravljalca o nekem dogodku



Sporočilo	Smer	Pomen
GetRequest *GetNextRequest* *GetBulkRequest*	upravljalec -> agent	"daj mi podatke" (vrednost, naslednja v seznamu, blok podatkov-tabela)
InformRequest	upravljalec -> upravljalec	medsebojno posredovanje vrednosti iz MIB
SetRequest	upravljalec -> agent	nastavi vrednost v MIB
Response	agent -> upravljalec	"tukaj je vrednost", odgovor na Request
Trap	agent -> upravljalec	obvestilo upravljalcu o izrednem dogodku

Izziv: poiščite RFC dokumente o SNMP in ugotovite razlike med njimi

- **SNMPv1:** prešibek za implementacijo vseh potrebnih zahtev (ni bilo možne povratne komunikacije)
- **SNMPv2:** izboljšan SNMPv1 na področjih **hitrosti**, **komunikacij med upravljalci**. Ni bilo varnosti (skupino naprav si lahko dal v isto skupnost -> prek tega si nadzoroval naprave)
- **SNMPv3:** dodatni varnostni mehanizmi (DES kriptografija(simetrična) redko uporabljena v praksi, problem: izmenjava ključev). Omogoča kritpografijo, avtentikacijo, integriteto, zagotavlja zaupnost

- **SNMP** uporablja transportni protokol **UDP**
  - **Vrata 161:** SNMP vrata na katerih poslušajo naprave po zahtevah
  - **Vrata 161:** Za obvestila (traps), na teh običajno poslušajo sistemi za nadzorovanje in upravljanje z omrežjem

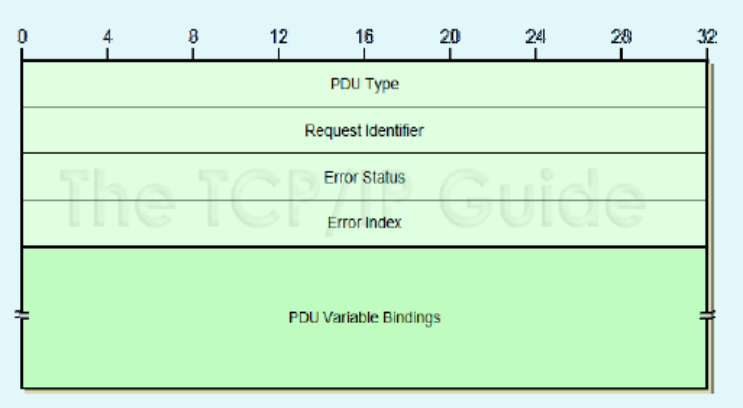
IMPLEMENTACIJA SNMP

- **VELIKOST PAKETOV:** SNMP paketi lahko vsebujejo obsežne informacije o objektih v MIB, UPD pa ima zgornjo mejo velikosti segmenta (TCP nima)
- **PONOVNO POŠILJANJE:** Ker se uporablja UDP, nimamo zagotovljene dostave in potrjevanja. Nadzor dostave je potrebno reševati na višjem nivoju
- **PROBLEM Z IZGUBLJENIMI OBVESTILI:** Če se obvestilo pri prenosu izgubi, pošiljatelj o tem nič ne ve, prejemnik pa ga tudi ne dobi

Izziv: kako SNMPv3 rešuje navedene težave?

- **VELIKOST PAKETKOV:** V paketu je dodano polje **Maximum Message Size**, ki pove največjo velikost paketa, ki ga pošiljatelj še lahko sprejme
- **PONOVNO POŠILJANJE:** SNMPv3 ima **dodatno obvestilo inform** (poleg trap). Ko pošiljatelj pošlje paket, zahteva obvestilo **inform**. Če to ne pride v določenem času, ponovno pošlje paket dokler ne prejme informa oz. **preseže mejo** kolikokrat lahko ponovno pošlje.
- **PROBLEM Z IZGUBLJENIMI OBVESTILI:** Enako kot za ponovno pošiljanje.

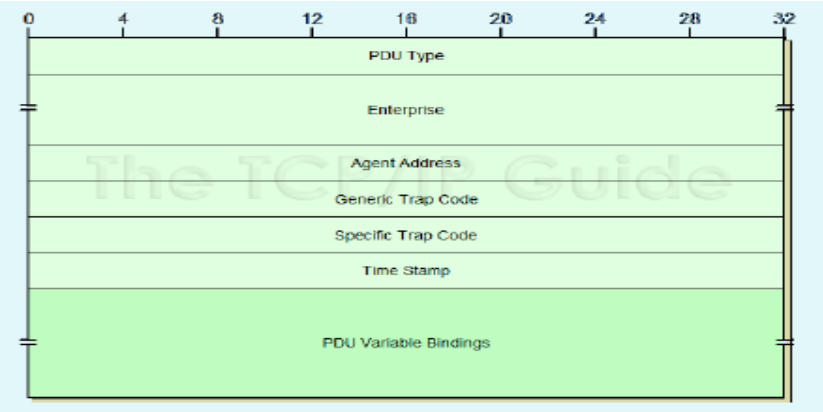
SNMP: SPOROČILO TIPA ZAHTEVA-ODGOVOR



PDU Type Value	PDU TYPE
0	GetRequest-PDU
1	GetNextRequest-PDU
2	Response-PDU
3	SetRequest-PDU
4	Old Trap-PDU (Obsolete)
5	GetBulkRequest-PDU
6	InformRequest-PDU
7	Trapv2-PDU
8	Report-PDU

- **Request ID:** Številka, ki povezuje zahteve z odgovori. Naprava, ki odgovori, ko shrani v paket tipa Response. Uporablja se tudi za umetno kontrolo prejetih paketov (SNMP namreč uporablja UDP transportni protokol, ki tega ne zagotavlja!)
- **Error status:** Koda napake, ki ga agent posreduje v paketu tipa Response. Vrednost 0 pomeni, da do napake ni prišlo, ostale vrednosti definirajo točno napako.
- **Error index:** Če je prišlo do napake, je ta vrednost indeks objekta, ki je povzročil napako
- **Variable Bindings:** Pari ime-vrednost (name-value), ki definirajo objekte in njihove vrednosti.

SNMP: SPOROČILO TIPA OBVESTILA



- **PDU Type:** Vrednost, ki definira tip sporočila. Vrednost 4/7 pomeni obvestilo (trap message)
- **Enterprise:** Identifikator skupine.
- **Agent Address:** IP naslov agenta, ki je generiral obvestilo.
- **Generic Trap:** Splošna koda napake - iz predefiniranega šifranta.
- **Specific Trap:** Specifična koda napake (odvisna od proizvajalce opreme)
- **Time Stamp:** Čas, odkar se je naprava nazadnje inicializirala. Uporablja se za beleženje.
- **Variable Bindings:** Pari ime-vrednost (name-value), ki definirajo objekte in njihove vrednosti.

VERZIJE SNMP

- **SNMPv1:** **prešibek** za implementacijo vseh potrebnih zahtev (ni bilo možne povratne komunikacije)
- **SNMPv2:** izboljšan SNMPv1 na področjih **hitrosti**, **komunikacij med upravljalci**. Ni bilo varnosti (skupino naprav si lahko dal v isto skupnost -> prek tega si nadzoroval naprave)
- **SNMPv3:** **dodatni varnostni mehanizmi** (DES kriptografija(simetrična) redko uporabljena v praksi, problem: izmenjava ključev). Omogoča kritpografijo, avtentikacijo, integriteto, zagotavlja zaupnost

5. VARNOST

- **Zakaj je pomembna?**
  - SetRequest nastavlja nadzorovane naprave. **Zahtevo lahko pošlje kdorkoli!?**
  - Nekdo lahko prestreže pakete in jih spremeni. **MIMT**
  - Kdor koli lahko prestreže pakete in jih prebere, ter tako pride do informacij za katere ni avtoriziran
  - **Replay napad:** Nekdo prestreže paket in ga kasneje pošlje in tako spremeni konfiguracijo
- Varnostni elementi so vpeljani šele v **SNMPv3**, prejšnji dve različici jih **nista imeli**. SNMPv3 ima vgrajeno varnost na osnovi **uporabniških imen**.

izziv: preberi RFC 3414 in poišči informacijo, proti kakšnim vdorom omogoča SNMPv3 zaščito? Kako je z napadi Denial of Service in prisluškovanjem prometa?

- Zagotavlja **integriteto**, zagotovljen je **izvor** paketkov, paketi ki so **zgenerirani izven časovne sekvence** so zavrnjeni DOS napad je še vedno mogoč

SNMP VARNOSTNI MEHANIZMI

1. **KRIPTIRANJE VSEBINE PAKETOV (PDU):** uporablja se DES (ključa je predhodno potrebno izmenjati)
2. **INTEGRITETA:** Uporablja se **zgoščanje sporočila s ključem**, ki ga poznata pošiljatelj in prejemnik. S preverjanjem poslane zgoščene vrednosti imamo zaščito pred **aktivnim ponarejanjem sporočil**
3. **ZAŠČITA PRED PONOVITVIJO ŽE OPRAVLJENE KOMUNIKACIJE (REPLAY ATTACK):** Uporaba enkratnih žetonov (angl. Nonce): pošiljatelj, mora sporočilo kodirati glede na **žeton**, ki ga določa sprejemnik (Običajno **število vseh zagonov sistema** pošiljatelja in **čas ki je minil od zadnjega zagona**)
4. **KONTROLA DOSTOPA:** kontrola dostopa na osnovi **uporabniških imen**. Pravice določajo, kateri uporabniki lahko berejo/naslavljajo katere informacije. Podatki o uporabnikih se hranijo v bazi **Local Configuration DataStore**, ki ima ravno tko nadzorovane objekte s **SNMP!**

6. KODIRANJE VSEBINE PDU

- Potrebujemo enotni način kodiranja ali nek **predstavitveni nivo teh podatkov**
  - **ASN.1** standard poleg podatkovnih tipov **definira tudi standarde kodiranja**
  - Uporablja se **TLV notacija** (Type, Length, Value – tip, dolžina, vrednost)

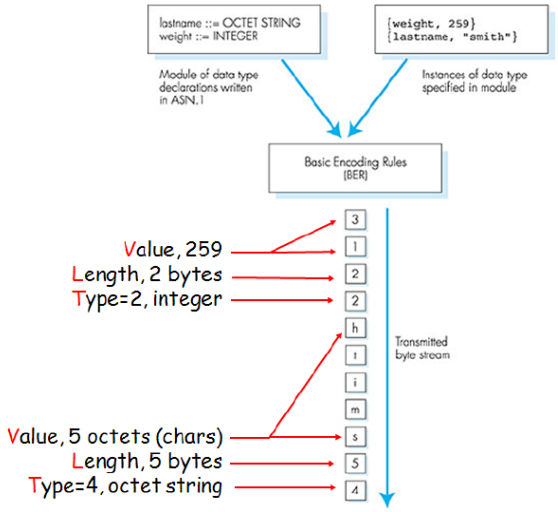
PREZENTACIJSKA STORITEV: MOŽNE REŠITVE

1. **Pošiljatelj upošteva** obliko podatkov, ki jo uporablja prejemnik: podatke pretvarja v njegovo obliko nato šele pošlje
2. Pošiljatelj pošlje podatke v svoji obliki, **prejemnik pretvori** v lastno obliko
3. Pošiljatelj pretvori v **neodvisno obliko** in nato pošlje. Prejemnik neodvisno obliko pretvori v svojo lastno obliko

- **ASN.1** uporablja 3. rešitev zgoraj (**neodvisno obliko**)
- Pri zapisovanju tipov se uporablja **pravila BER** (Binary Encoding Rules). Ta definirajo zapis **podatkov po principu TLV** (Type, Length, Value = tip, dolžina, vrednost)

PRIMER BER KODIRANJA PO PRINCIPU TLV

Osnovni ASN.1 podatkovni tip	Št. tipa	Uporaba (angl.)
BOOLEAN	1	Model logical, two-state variable values
INTEGER	2	Model integer variable values
BIT STRING	3	Model binary data of arbitrary length
OCTET STRING	4	Model binary data whose length is a multiple of eight
NULL	5	Indicate effective absence of a sequence element
OBJECT IDENTIFIER	6	Name information objects
REAL	9	Model real variable values
ENUMERATED	10	Model values of variables with at least three states
CHARACTER STRING	*	Models values that are strings of characters from a specified character set



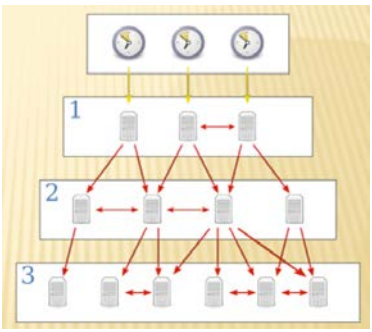
1 * 256^1 + 3 * 256^0

PROMET V REALNEM ČASU

1. OMREŽNI ČAS

- Včasih moramo uskladiti čas med večimi oddaljenimi sistemi
- Problem zakasnitve prenosa podatka
- Uporabimo lahko več sistemov hkrati
- **Zakaj je pomemben:**
  - Veliko storitev potrebuje sinhroniziran čas med napravami za pravilno delovanje

PROTOKOL NTP



- Je protokol za **sinhroniziranje časa** čez omrežje
- Uporablja UDP
- Pod nivojem 16 je dobljeni čas obravnavan kot napačen

PRENOS OD A DO B

- Osnovni vir težav je omrežje
  - Vsak paket lahko potuje po drugi poti
  - Vsak paket lahko potuje različno dolgo
    - Problem latence – ni tako velik pri enosmernem prometu
  - Nekateri paketi se izgubijo
- **Zamujene pakete se obravnava kot izgubljene**
- Sam protokol **poskrbi za časovni zamik** glede na nivo na katerem smo
- Aplikacija poskrbi za izgubljene pakete (saj uporabljamo UDP)

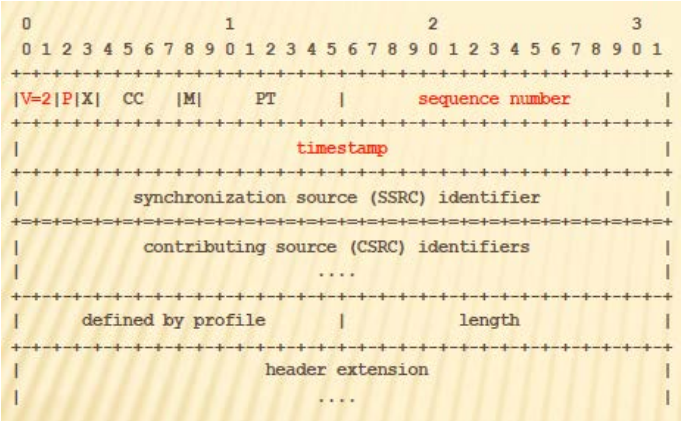
2. PROTOKOL RTP

- Postavljen na protokolih **IP in UDP** (nepovezavni način)
- **UPORABA:**
  - IP telefonija
  - **VOD** – Video On Demand **(Uporablja RTSP)**
- **Osnovne funkcionalnosti**
  - Skrbi za pravo zaporedje paketov
  - Skrbi za časovne značke dogodkov
- **Dodatne funkcionalnosti:**
  - Ena povezava lahko prenaša **več podatkovnih tokov** (virov dogodkov): zvok levi, zvok desni,… slika desnega očesa, slika levega očesa; podnapisi ,..
  - **Identifikator vira/seje** in njegov sinhronizacijski vir
  - Poseben element – **mešalec (mixer),** ki lahko združuje več sej v eno sejo
  - V združeni seji, komu v resnici pripada poslani paket

PODROBNOSTI

- Je prenosni protokol, ki služi **prenosu podatkov**
  - **Ne vključuje ukazov** za **začetek** povezave in **vzdrževanje** povezave
- Omogoča aplikacijam prenos posebnih podatkov (za predvajanje zvoka, filma, ..) – profil
- Za nadzor delovanja uporablja **RTCP protokol**

OBLIKA PAKETA



OSNOVA
- **V** – Verzija 2
- **P** – Zapolnitev (padding)
- **Sequence number** – Številčenje paketov poslanih v toku
- **Timestamp** – Časovna značka dogodka

DODATNE FUNKCIONALNOSTI:
- **SSRC** – Identifikator vira (Sync source)
- **CC** – Število mešanih virov
- **CSRC** – Identifikatorji mešanih virov (Countributing source)

VIŠJI PROTOKOL/APLIKACIJA:

- **PT** – Identifikacija protokola
- **M** – Poseben bit za potrebe protokola
- **X** – Ali je prisotna razširitev glave (Zadnji del je razširitev glave)

3. NADZORNI PROTOKOL RTCP

1. Sporoča o **kakovosti prenašanega prometa** (**RR**: Reciver Report in **SR**: Sender Report)
2. Dodaten opis **vira toka dogodkov** (**SDES**: Source Description Items)
3. Skrbi za **pravilno gostoto** pošiljanja sporočil o kakovosti prenosa
4. Prenaša lahko še dodatne podatke za potrebe aplikacije **(APP**: Application-specific functions)

- Za potrebe **RTCP** je uporabljena stalna pasovna širina
- Če je veliko sodelujočih strank (multicast), potem je **gostota poročanja manjša**

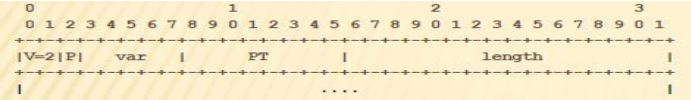
Izziv: kakšne vse podatke lahko prenaša RTCP o viru dogodkov? Kaj je to CNAME?

- **CNAME:** Formata ''user@host'' edino obvezno in različno za vsakega userja
- **NAME:** Ime, ki opisuje source npr. Webcam,..
- **EMAIL:** Email userja, ki streama
- **PHONE:** Telefonsko številko userja, ki streama
- **LOC:** Geografsko lokacijo sourca
- **TOOL:** Ime aplikacije, ki generira stream podatkov
- **NOTE:** Ki ga nastavi user, ki streama, npr. afk, dnd..

Izziv: kako izgleda poročilo o kakovosti prometa? Kakšne podatke vključuje?

- **Procent izgubljenih paketkov** poslanih po zadnjem reportu
- **Število izgubljenih paketkov**
- **Jitter** (približni odklon v času potovanja paketka)
- **Timestamp** zadnjega Source report paketa
- **Delay** glede na zadnji Source report paket

RTCP – OBLIKA PAKETA



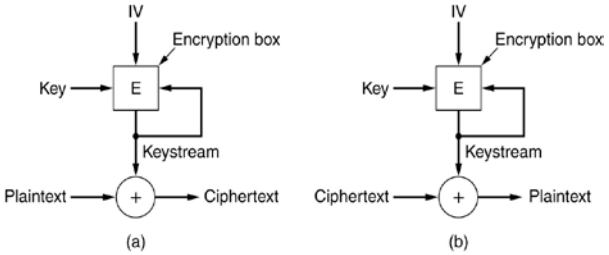
- **V** – Verzija 2
- **P** – Zapolnitev (padding)
- **PT** – Ukaz: SR, RR, SDES, BYE, APP
- **VAR** – različne vrednosti v odvisnosti od ukaza

4. VARNI RTP

- **RTP** protokol uporablja **UDP prenos**, ki nima SSL plasti
- Zato moramo varnost za RTP dograditi sami
- Nekako **izmenjamo ključe**, toda **paketi se izgubljajo**
- Drugačen pristop: **Kriptiranje s tokom šifer**

KRIPTIRANJE S TOKOM ŠIFER

- **Začetna vrednost** (IV) je poznana obema stranema
- Obema stranema je poznan tudi **ključ**
- **Vsak paket se ločeno zakriptira**
- + je preprost **XOR** ali kakšen podoben algoritem
- Če se paket izgubi, samo v prazno zavrtimo E



THE SECURE REAL-TIME TRANSPORT PROTOCOL (SRTP)

- Zasnovan na RTP
- Varnost dodana z **kriptiranjem s tokom šifer**

izziv: kako si obe strani izmenjata ključe?

izziv: v RFC je omenjena HMAC funkcija (tudi RFC 2104); kako deluje in kako se uporablja? Kaj je to f8, ki je omenjena v standardu?

REAL-TIME STREAMING PROTOCOL (RTSP)

- **OSNOVNI UKAZI:**
  - Setup, play, record, pause, teardown
- Ima še dodatne ukaze za nastavljanje in branje parametrov
- Primer uporabe na spletnih straneh:
  - **<a href**=“rtsp://tainta.isp.ponudnik/Dolina_miru”**>**prelep slovenski film ****

Izziv: eno od polj, ki jih odjemalec nastavi v zahtevi strežniku je transport. Kako izgleda, kaj pomeni in čemu služi?

Izziv: kje se vidi povezava med RTSP in RTP – na primer pri RTP smo imeli v glavi SSRC polje; ali obstaja tudi pri RTSP in če da, kje ter kako izgleda?

RAZPOŠILJANJE (MULTICAST)

NAČINI NASLAVLANJA

- **UNICAST** (Tradicionalno): pošiljanje enemu ciljnemu **IP naslovu** (unikaten v omrežju)
- **BROADCAST:** naslavljanje ''vseh prejemnikov'' v podomrežju (npr. iskanje usmerjevalnika ali strežnika, nujno sporočilo). **Ne dostavlja paketov izven omrežja**

Kako poslati samo izbrani skupini naslovov, tudi izven lokalnega omrežja?

- **Multicast** naslavljanje (razpošiljanje) omogoča dostavo skupinam ne glede na meje podomrežij
- **IGMP** (**I**nternet **G**roup **M**anagement **P**rotocol) se uporablja za **upravljanje s skupinami**

PRIMER: Poslati želimo 4-im od 6-ih računalnikov v omrežju. Kako?

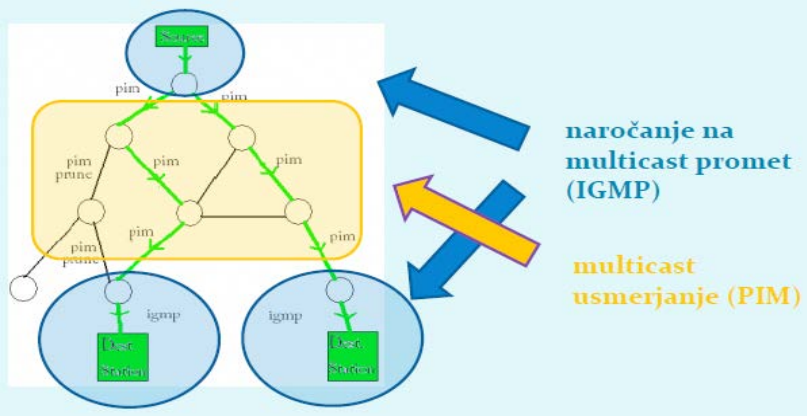
1. **UNICAST:** Potrebujemo 6 kopij istega paketa, večkratno pošiljanje **obremenjuje medij**
2. **BROADCAST:** Naslovi vse računalnike, filtriranje pravih prejemnikov prepustim protokolom na višjih plasteh
3. **MULTICAST:** Pošljemo **''Posebnemu naslovu''**, ki predstavlja **SKUPINO prejemnikov**, ki posluša pakete, naslovljene na ta naslov
    - Podobno kot broadcast: **paket dobijo vsi**
    - Vendar: **Filtriranje se izvede na omrežnem nivoju** – IP (včasih tudi na povezavnem)

USMERJANJE PAKETOV

- Broadcast paketov usmerjevalniki **NE posredujejo**, torej ostajajo znotraj lokalnega omrežja
- **USMERJANJE PRI RAZPOŠILJANJU** je praktično: en sam paket **USMERJEVALNIKI** razmnožijo in posredujejo samo preko tistih vmesnikov, **kjer so poslušatelji paketa**. Ime skupine je 32 bitno število (skoraj)

IZZIVI PROTOKOLA:

- **Odkrivanje**, kje so prejemniki paketa
- **Razpošiljanje zahteva dodatno delo:** usmerjevalni protokoli, posredovane informacije o poslušateljih
- **Razpošiljalni naslovi ne oblikujejo (pod)mrež** -> maska ima 32 bitov. V usmerjevalnih tabelah zato zahtevajo **posebne vnose**
- **Varnost:** prisluškovalec se lahko naroči na poslušanje paketov in postane legitimni prejemnik
- Kaj narediti, če samo en prejemnik javi, da ni dobil paketa?



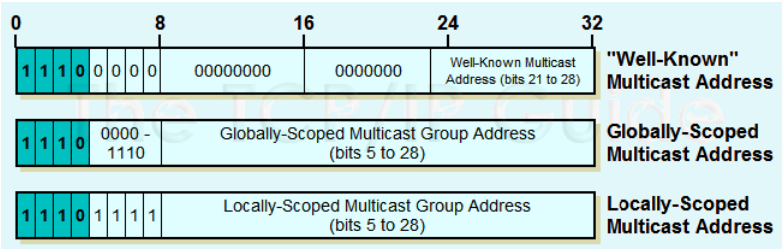
APLIKACIJE RAZPOŠILJANJA

- **Pošiljanje velikih datotek preko omrežja** (glavni urad podružnicam) – zanesljivi prenos
- Nadgradnja programske opreme v velikem omrežju
- **Data streaming** (npr. pošiljanje podatkov o delnicah vsem finančnim družbam)
- **Avdio/video streaming**
- **Video na zahtevo** (spremljanje TV programa)
- **Izvedba konferenc** (pomislek: boljša uporaba konferenčnega centra, ki odloča, kdo lahko govori in čigave pakete posredovati drugim)
- **Aplikacije v realnem času z RTP,** ki se uporablja za zagotavljanje tekoče in kakovostne dostave v okoljih kjer se uporablja razpošiljanje

1. NASLAVLJANJE IPv4 in IPv6

- Imena razpošiljevalnih skupin so dejansko **posebej rezervirani IPv4 naslovi:**
  - **224.0.0.0 – 239.255.255.255 (razred D)**
- Posebni naslovi znotraj tega obsega

Razpon naslovov	Opis
224.0.0.0 - 224.0.0.255	Rezervirano za **znane** ("well-known") multicast naslove
224.0.0.1	Vsi **sistemi** (vmesniki in usmerjevalniki)
224.0.0.2	Vsi **usmerjevalniki**
224.0.1.0 - 238.255.255.255	**Globalni** multicast naslovi (dosegljivi v internetu)
239.0.0.0 - 239.255.255.255	**Lokalni** multicast naslovi (lokalno omrežje)

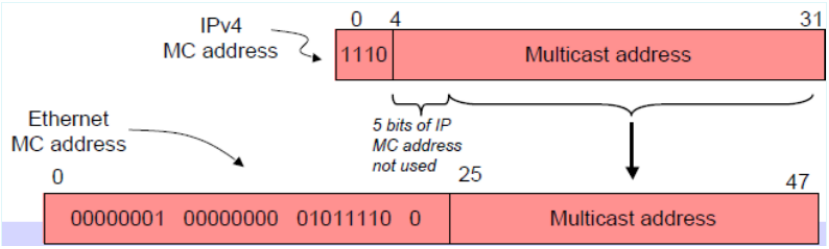


2. NASLAVLJANJE IPv6

- Ime razpošiljevalne skupine je 128-bitno število – IPv6 naslov, ki se prične z **FF**
- **FF02::1** (link local: **vsi VMESNIKI**)
- **FF02::2** (link local: **vsi USMERJEVALNIKI**)
- Struktura IPv6 naslova:

3. PRESLIKAVA V POVEZAVNE NASLOVE

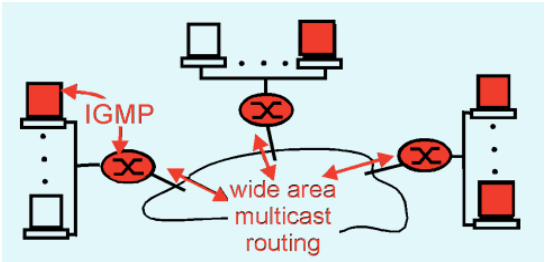
- Ethernet in FDDI okvirji uporabljajo 48 bitne naslove. Naslovi **01-00-5e-00-00-00 do**
- **01-00-5e-ff-ff-ff** predstavljajo **naslove razpošiljevalnih skupin**.
- Predpona **01-00-5e** pomeni **razpošiljevalni okvir**, naslednji bit je 0, ostalih 23 bitov tvori ime razpošiljevalne skupine.
- Ker so IP razpošiljevlani naslovi dolgi 28 spremenljivih bitov, preslikava ni enolična! V okvir se vstavi samo 23 manj pomembnih bitov. To pomeni, da se po 32 (25) naslovov združuje v isti naslov na drugi plasti
- **Omrežna plast odloča, ali so datagrami pomembni za sprejem ali ne.**



4. PROTOKOL IGMP

- Mrežni protokol je IPv4 in številka protokola je 2
- **IGMP** skrbi za upravljanje s tem, kdo so prejemniki razpošiljanih sporočil.
- **Omogoča:**
  - Pridružitev skupini
  - Izstop iz skupine
  - Zaznavanje drugih vmesnikov v skupini

- **IGMP** komunikacija poteka **med odjemalcem in najbližjem razpošiljevalnim usmerjevalnikom**
- Na podlagi protokola IGMP usmerjevalniki dobijo nalogo povezati se v strukturo razpošiljevalnega drevesa



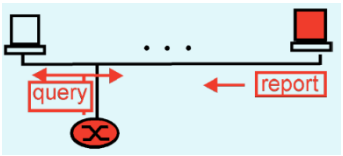
VERZIJE IGMP

- **IGMPv1:** Vmesniki se lahko pridružijo skupinam. **Sporočila za izstop ne obstajajo,** usmerjevalniki uporabljajo mehanizem s pretekom časa, da odkrivajo skupine, ki za vmesnik niso zanimive.
- **IGMPv2: Dodana sporočila za izstop iz skupine.** S tem omogočitev hitrejšega sporočanja usmerjevalniku o prekinitvi dostave nepotrebnega prometa.
- **IGMPv3:** Večje spremembe v protokolu. Vmesniki lahko določijo **SEZNAM drugih vmesnikov, od koder želijo prejemati promet.** Promet od ostalih vmesnikov omrežje blokira.

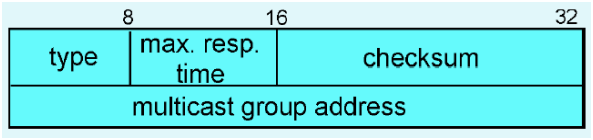
5. IGMP SPOROČILA

- Kako z IGMP udejaniti upravljanje s skupinami?

Dejanje	IGMP sporočilo	IP Destination Address	IGMP Group Address
pridružiti se želim skupini	Group Membership Report	naslov skupine	naslov skupine
Kdo vse je član določene skupine?	Group Membership Query	naslov skupine	naslov skupine
katere skupine obstajajo?	Group Membership Query	vsi vmesniki (224.0.0.1)	0.0.0.0
sem član skupine, o kateri se poizveduje, želim se odzvati, da sem član	Group Membership Report	naslov skupine	naslov skupine
zapustiti želim skupino	Group Leave Report	vsi usmerjevalniki (224.0.0.2)	naslov skupine

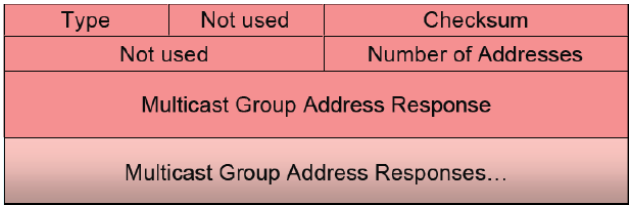


- **IGMP** sporočilo je dolgo 8 zlogov



- **TYPE** – tip sporočila
  - 17 (0x11): **Group Membership Query** (odkrivanje članov skupine)
  - 18 (0x12): **Group Membership Report** IGMP v1 (objava prejemnika)
  - 22 (0x16): **Group Membership Report** IGMP v2 (objava prejemnika)
  - 34 (0x22): **Group Membership Report** IGMP v3 (objava prejemnika)
  - 23 (0x17): **Leave Group Report IGMP v2** (objava, da je prejemnik zapustil skupino)
- **CHECKSUM** - kontrolna vsota (ne pokriva IP glave)
- **MULTICAST GORUP ADDRESS** – IPv4 naslov razpošiljevalne skupine

- **Posebno sporočilo: IGMPv3** Group Membership Report

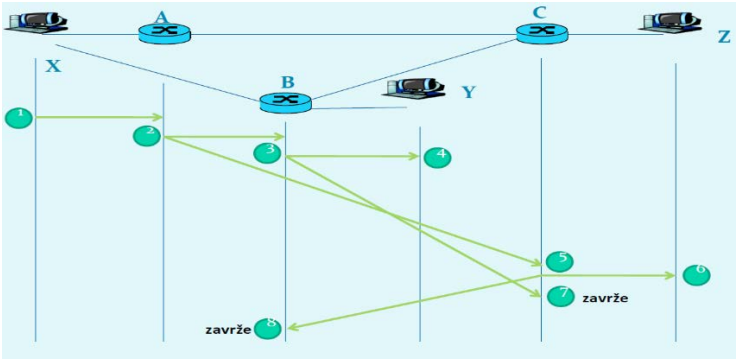


- **TYPE =** 0x22
- Odgovori vseh vmesnokov v skupini so **zbrani v istem paketu**
- Vmesnik **čaka na odgovore drugih prejemnikov v skupini**, preden odgovori sam
  - Posebna oblika paketa torej omogoči izogibanje podvojenemu multicast prometu

PRIJAVA NA VIR

- Za pridružitev skupine, se pošlje **GMR** sporočilo z vrednostjo TTL=1 **(dostava samo najbližjemu usmerjevalniku)**
- Usmerjevalnik evidentira, da mora skupinske pakete posredovati novemu naročniku (kako? povezavni razpošiljevalni naslov / kopije datagramov na IP naslov)

- **Usmerjevalnik sporoči sosednjim usmerjevalnikom, da ima novega naročnika.** Če bi vsak usmerjevalnik sporočil enako naprej, **pride do problema** – paketi bi se posredovali navzkrižno preko vseh povezav v omrežju.
    - **Uporablja se RLP algoritem** (Reverse Path Lookup): Zavržemo vse multicast pakete, ki pridejo od usmerjevalnikov, ki ne povezujejo z izvorom paketa po najbližji poti
    - **Usmerjevalniki imajo posebne usmerjevalne protokole:** za multicast promet: npr. protokol **PIM-SM** (Protocol Independent Multicast – Sparse Mode)

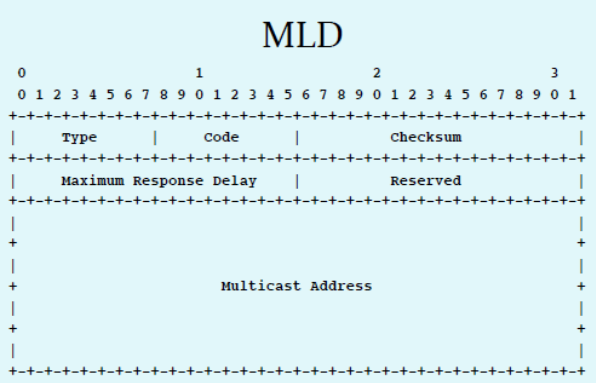


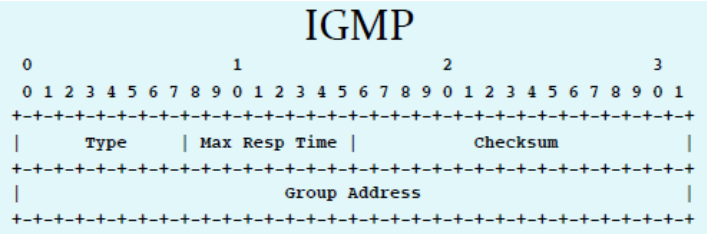
6. PROTOKOL MLD

- Dejansko je protokol za IPv6 razpošiljanje in ima enako funkcionalnost kot IGMP

Izziv: poiščite razlike med MLD in IGMP

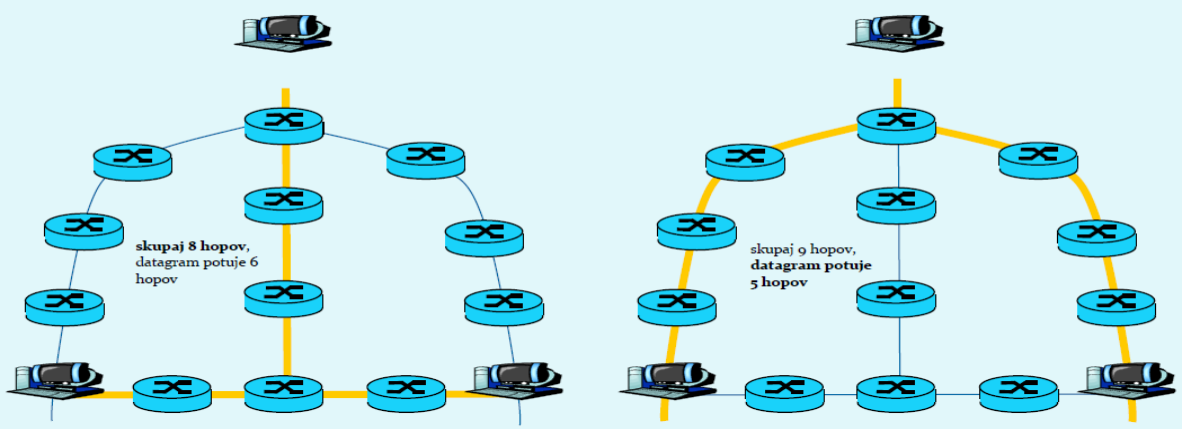
Izziv: kaj pa sobivanje IGMP (IPv4) in MLD (IPv6)?



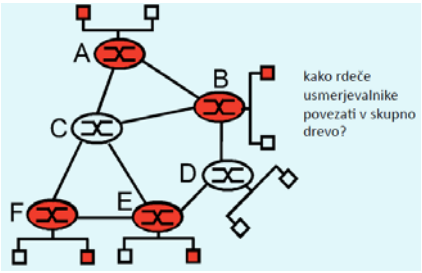


7. RAZPOŠILJEVALNA DREVESA

- Paketi se gibljejo v obliki razpošiljevalnega drevesa
- **Drevo lahko optimizira različne kriterije:**
    - **Slika 1:** skupna dolžina poti (število hopov) vseh datagramov
    - **Slika 2:** najkrajša pot za vsak datagram posebej (minimalno vpeto drevo)

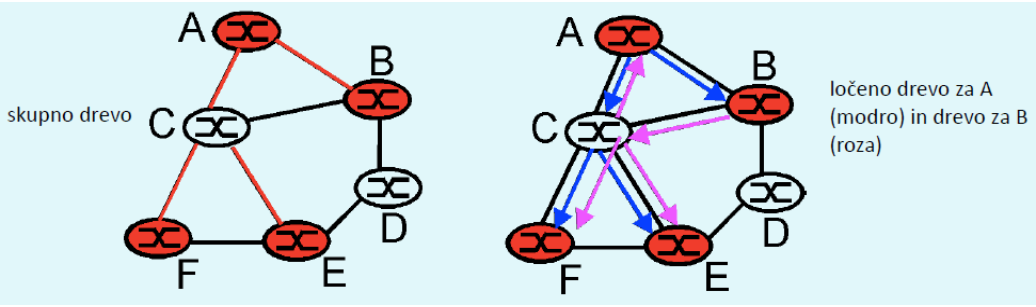


- **Naloga usmerjanja:** najti drevo povezav, ki povezuje vse usmerjevalnike v isti razpošiljevalni skupini
- Za komunikacijo med usmerjevalniki **potrebujemo usmerjevalne algoritme** (delujejo na omrežni plasti), npr. : PIM, BGP, DVMRP..



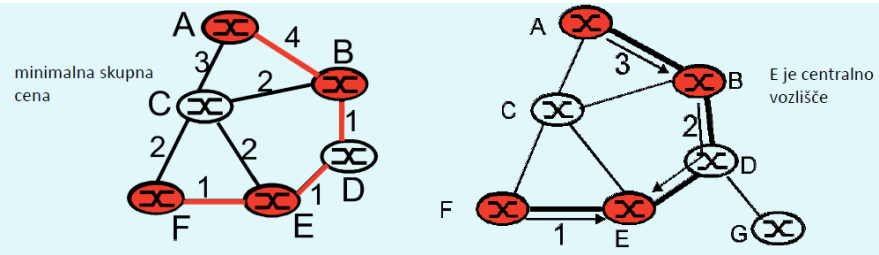
DVE REŠITVI ISKANJA RAZPOŠILJEVALNEGA DREVESA

- Za usmerjanje razpošiljevalnega prometa se določi eno samo drevo (group-shared tree) **- slika levo**
- Določitev **ločenega drevesa za vsakega udeleženc**a v skupini (source-based tree); za N članov skupine imamo torej N dreves (za vsako razpošiljevalno skupino) - **slika desno**



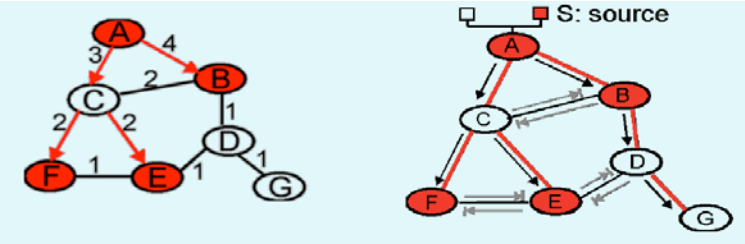
Določanje skupnega drevesa (group-shared)

- Iskanje drevesa z **minimalno skupno ceno** (Steinerjev algoritem za vpeta drevesa), **slika levo**
- **Določitev centralnega vozlišča** "rendez-vouz point". Usmerjanje do njega po unicast pravilih. (**Ves promet skozi centralno vozlišče**) Usmerjevalnik se pridruži drevesu, ko na poti do centralnega vozlišča naleti na prvo vozlišče, ki je že v drevesu, **slika desno**



Določanje dreves posameznih pošiljateljev (source-based)

- **Iskanje drevesa najkrajših poti v grafu** (uporaba algoritma Dijkstra, ki išče drevo najkrajših povezav glede na podano začetno vozlišče), slika levo
  - Usmerjevalniki morajo poznati stanja vseh povezav (link-state)
- **Uporaba RPL (Reverse Path Lookup):** ne sprejmemo sporočil od usmerjevalnikov, ki niso na najbližji poti do izvora sporočila, slika desno



8. USMERJANJE RAZPOŠILJANJA

USMERJEVALNI PROTOKOLI

- Skrbijo za **oglaševanje skupin** v omrežju
- Delimo jih glede po **2 kriterijih (2x2=4 skupine)**
- **Razpršeno/Gosto** (sparse-mode / dense-mode)
  - **Sparse-mode:** posamezna vozlišča zahtevajo vključitev v drevo (pull princip)
  - **Dense-mode:** razpošiljane pakete razpošljemo po vsem omrežju, usmerjevalniki se odjavljajo, če so nepotrebni (push princip).
- **Intra** (znotraj domene) / **Interdomain** (med domenami)
- Obstaja povezava med načinom delovanja in vrsto drevesa, ki ga protokol gradi

Protokol	Način delovanja	vrsta drevesa	Vrsta
PIM-SM	sparse	skupno	znotraj in med domenami
PIM-DM	dense	posamezno	znotraj domen
CBT	sparse	skupno	znotraj in med domenami
MOSPF	dense	posamezno	znotraj domen
BGMP	dense	posamezno	znotraj domen
DVMRP	dense	posamezno	znotraj in med domenami

9. PIM-SM (Protocol Independent Multicast - Sparse Mode)

- **PIM-DM:** dense-mode, posamezno drevo
- **PIM-SM:** sparse-mode, skupno drevo, včasih posamezno
- Protokola **PIM-SM** in **PIM-DM** sta primerna za usmerjevalnike, ki že izvajajo unicast usmerjanje. **Sta neodvisna od unicast protokola**
- Sporočila uporabljajo IP mrežni protokol s številko protokola 103
- Sporočila med usmerjevalniki so **unicast ali multicast na naslov 224.0.0.13** (vsi PIM usmerjevalniki)

DELOVANJE PIM-SM

vzpostavitev arhitekture

- kandidatni zagonski (*bootstrap*) usmerjevalniki (c-BSR) oznanijo svojo prisotnost (tip sporočila BOOTSTRAP) in izberejo glavni zagonski usmerjevalnik BSR
- kandidatni centralni (*rendezvouz*) usmerjevalniki (c-RP) oznanijo prisotnost BSR usmerjevalniku (tip sporočila CANDIDATE-RP-ADVERTISEMENT)
- BSR izbere dokončni centralni usmerjevalnik (RP) za vsako skupino in jih oznani s sporočili tipa BOOTSTRAP

pošiljanje podatkov

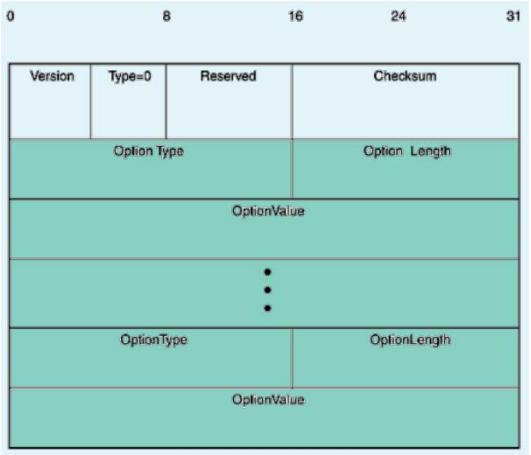
- PIM-SM usmerjevalniki se zaznajo in vzdržujejo komunikacijo s sporočili HELLO
- vmesnik, ki pošilja podatke na skupinski naslov, razpošlje datagram na lokalni segment mreže
- izbrani usmerjevalnik na mreži (*designated router*) enkapsulira datagram v tip sporočila REGISTER in ga pošlje RP
- RP dekapsulira datagram in ga razpošlje po razpošiljevalnem drevesu

vzdrževanje naročnine

- kadar RP zazna, da v skupini ni več prejemnikov, pošlje sporočilo REGISTER-STOP vsem izbranim usmerjevalnikom
- kadar se novi uporabnik želi pridružiti skupini, pošlje sporočilo JOIN/PRUNE s seznamom vseh želenih skupin in dovoljenih prejemnikov

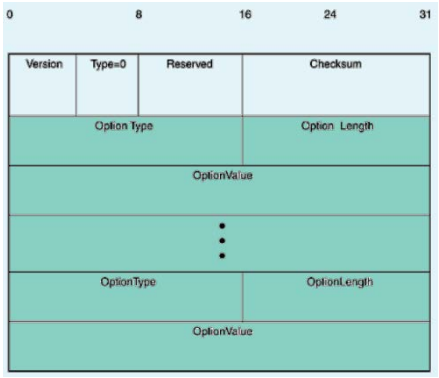
OBLIKA PAKETA – VSEBINA GLAVE

vrednost	pomen
0	hello
1	register
2	register stop
3	join/prune
4	bootstrap
5	assert
6	candidate-rp-advertisement



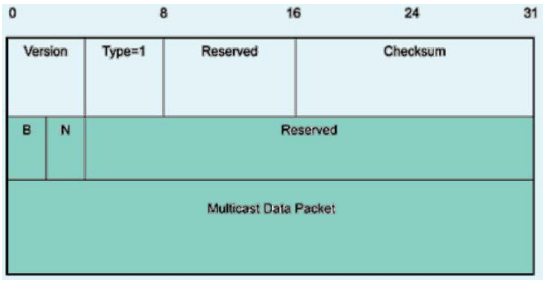
PIM-SM - PAKET HELLO

- **Namenjen vzdrževanju povezav med usmerjevalniki**
- V primeru, da se izbrani usmerjevalnik za pošiljanje multicast prometa ne odzove, se izbere drugi
- Paket vsebuje množico TLV vrednosti, kot so npr. potek časa, v katerem je pričakovan odgovor

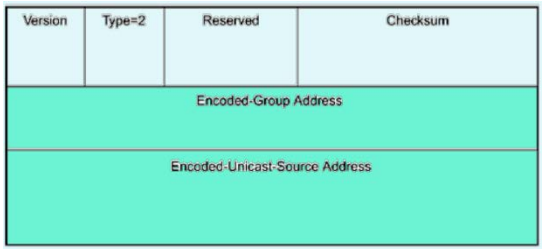


PIM-SM - paket REGISTER in REGISTER-STOP

- Sporočilo **REGISTER** nosi vsebino multicast sporočila do centralnega usmerjevalnika (unicast)
  - **B (border router)** - sporočilo prišlo usmerjevalniku, ki je neposredno povezan z vmesnikom,
  - **N (null)** - paket je prazen, za vzpostavitev povezanosti



- Sporočilo **REGISTER STOP** pošlje centralni usmerjevalnik izbranemu usmerjevalniku, z njim **sporoči naj ne pošilja sporočil** (prejemnikov ni / sporočila dobiva že od drugje)



PIM-SM - JOIN/PRUNE

- Omogoča prejemniku, da se **prijavi/odjavi** od prejemanja multicast prometa
- **PIM-SM** ima **Number of Pruned sources enak 0** (ker uporablja skupno drevo)
- Polja za **prijavo/odjavo**:
  - Encoded Join Source Address
  - Encoded Pruned Source Address