

Priklop in zagon naprave

# **KOMUNIKACIJSKI PROTOKOLI IN OMREŽNA VARNOST**

# VSEBINA

---

- ✘ zagon računalnika
- ✘ zagon preko omrežja – bootp
- ✘ priklop na omrežje

# ZAGON RAČUNALNIKA

---

- ✘ CPE ob priklopu na napajanje nastavi vrednost PŠ (programskega števca) na točno določeno vrednost
  - ✧ izziv: na katero vrednost se nastavi pri intel procesorju? Na katero pri powerpc? Na katero pri arm?
- ✘ za tem začne izvajati ukaze
  - + običajno delovanje
- ✘ pomembno: kaj se nahaja v pomnilniku na mestu, kjer prične z delom CPE

# BIOS

---

- ✘ *Basic I/O System – firmware*
- ✘ Sestoji iz dveh sklopov:
  - + koda, ki se prične izvajati ob zagonu
  - + 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 „obuta” – ima škornje, *boot*

# 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 za delo z V/I enotami gonilnike iz BIOS
- ✘ slednji so imeli dve pomanjkljivosti: i) niso bili „prijazni“; ii) niso bili učinkoviti
- ✘ OS je pričel uporabljati svoje gonilnike

# NALAGANJE OS – SODOBNO

---

- ✘ BIOS v resnici **naloži nek program, ki ga nato prične izvajati**
- ✘ najde ga na prvem bloku V/I enote – *master boot record, MBR*
- ✘ naloženi program ni nujno, da je OS, ampak lahko naloži naslednji (ali enega od naslednjih) program, ki je še OS
  - + možnost nalaganja enega od večih OS
    - ✘ izziv: kako se imenuje ta novi program? poiščite vsaj dva njegova primera.

# NALAGANJE PROGRAMA – DRUGAČE

- ✘ BIOS v resnici i) naloži nek program, ki ga ii) nato prične izvajati.
- ✘ Kaj, če bi BIOS naložil program namesto z diska, s strežnika na omrežju (zamenjava i) vendar ohranimo drugi del).
- ✘ Potrebujemo definicijo načina pogovora našega računalnika s strežnikom – potrebujemo protokol.
- ✘ Kaj, če naenkrat pomnilnik ni več omejitev?
  - + UEFI, *Unified Extensible Firmware Interface* (199x)

# NALAGANJE PROGRAMA Z OMREŽJA

## × Prednosti:

- + ne potrebujemo diska na računalniku
- + OS preprosto zamenjamo za vse računalnike, saj ga zamenjamo samo na strežniku

## × Slabosti:

- + ranljivost
- + počasnost
- + varnost?



# VSE JE V ŠTEVILKAH

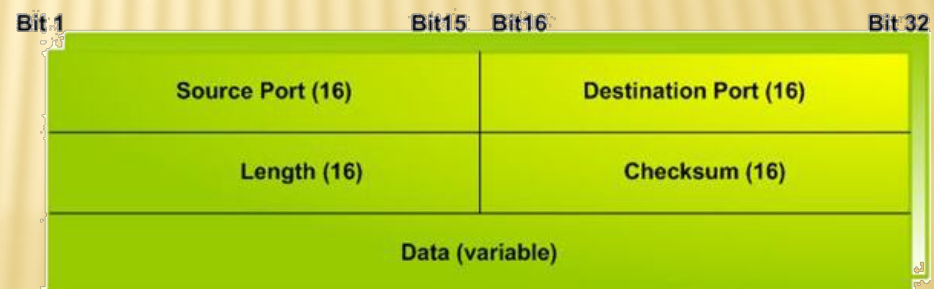
---

- × [www.fri.uni-lj.si](http://www.fri.uni-lj.si) = 212.235.188.25
- × Storitve DNS preslikuje med črkovnim nizom in številko.
  - + namesto DNS storitve lahko uporabimo preslikovalno tabelo v datoteki /etc/hosts
- × Kako najdemo strežnik DNS storitve?
- × Kako strežnik DNS storitve najde druge strežnike DNS?
  - + poznati mora njihove IP naslove
  - + datoteka /etc/namedb/named.root

# VSE JE V ŠTEVILKAH

- ✗ 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?

The UDP Segment Format



```

#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
#
# The latest IANA port assignments can be gotten from
#
#   http://www.iana.org/assignments/port-numbers
#
# The Well Known Ports are those from 0 through 1023.
# The Registered Ports are those from 1024 through 49151
# The Dynamic and/or Private Ports are those from 49152 through 65535
#
# $FreeBSD: src/etc/services,v 1.89 2002/12/17 23:59:10 eric Exp $
#   From: @(#)services    5.8 (Berkeley) 5/9/91
#
# WELL KNOWN PORT NUMBERS
#
rtmp          1/ddp      #Routing Table Maintenance Protocol
tcpmux        1/udp      # TCP Port Service Multiplexer
tcpmux        1/tcp      # TCP Port Service Multiplexer
#
#                               Mark Lottor <MKL@nisc.sri.com>
nbp           2/ddp      #Name Binding Protocol
compressnet   2/udp      # Management Utility
compressnet   2/tcp      # Management Utility

...

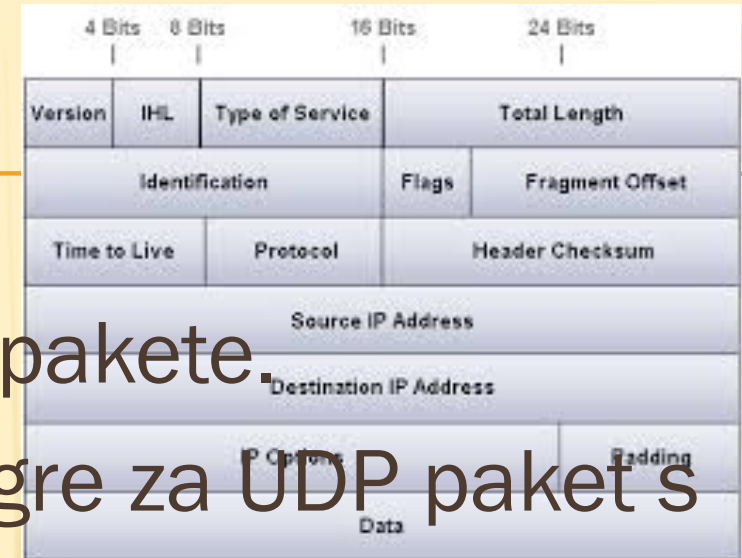
ftp-data      20/udp      # File Transfer [Default Data]
ftp-data      20/tcp      # File Transfer [Default Data]
ftp           21/udp      # File Transfer [Control]
ftp           21/tcp      # File Transfer [Control]
ssh           22/udp      # SSH Remote Login Protocol
ssh           22/tcp      # SSH Remote Login Protocol
telnet        23/udp      # Telnet
telnet        23/tcp      # Telnet
smtp          25/udp      # Simple Mail Transfer
smtp          25/tcp      # Simple Mail Transfer

...

```

# VSE JE V ŠTEVILKAH

- ✘ DNS protokol uporablja UDP pakete.
- ✘ V glavi paketa označimo, da gre za UDP paket s številko 17.
- ✘ Nimamo storitve, ki bi preslikovala med imenom UDP in 17
  - + 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?



# IN OD KJE PRIDEJO ŠTEVILKE

- ✘ svetovni dogovor o številkah
- ✘ številke hrani in oglašča IANA – *The Internet Assigned Numbers Authority*, [www.iana.org](http://www.iana.org)
  - + korenski DNS strežniki:  
[www.iana.org/domains/root/db/arpa.html](http://www.iana.org/domains/root/db/arpa.html)
  - + vrata: [www.iana.org/assignments/port-numbers](http://www.iana.org/assignments/port-numbers)
    - ✘ izziv: napišite program, ki tvori samodejno datoteko services iz podatkov na IANA strežniku
  - + protokoli: [www.iana.org/protocols/](http://www.iana.org/protocols/)
    - ✘ izziv: kakšni podatki so na [www.iana.org/domains/root/db/si.html](http://www.iana.org/domains/root/db/si.html)?

# NALAGANJE OS Z OMREŽJA

---

- ✘ ob zagonu računalnik lahko ali pa ne pozna nekaterih svojih podatkov:
  - + ime
  - + IP naslov
  - + ...
- ✘ vsekakor mora znati govoriti protokol, ki bo omogočal nalaganje OS
  - + podobno, kot mora poznati način branja podatkov z diska – gonilnik
  - + rokovalnik protokola, ki mora biti jedrnat

# NALAGANJE OS Z OMREŽJA – KORAKI

- ✘ Za uspešno nalaganje mora računalnik:
  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
- ✘ Zadnji korak je enak kot pri nalaganju z diska
- ✘ Načrtovalska odločitev: koraka 1. in 2. v enem protokolu (bootp) in korak 3. v drugem protokolu (npr. tftp)

# PROTOKOL BOOTP

---

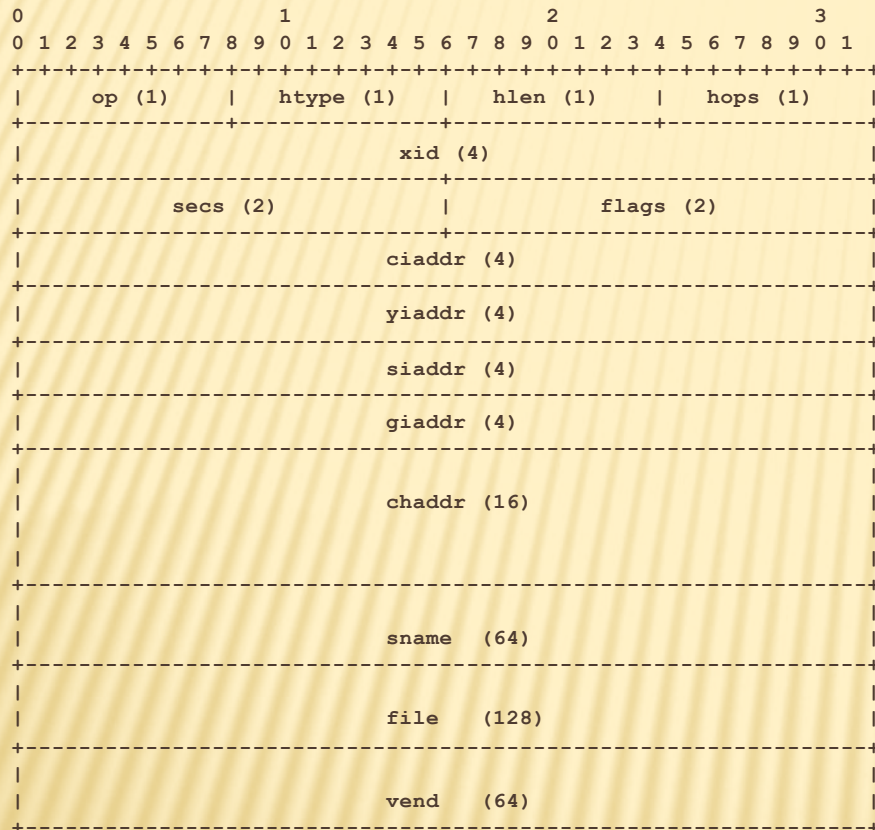
- ✘ definiran v RFC 951, **BOOTSTRAP PROTOCOL (BOOTP)**
  - ★ *obvezno: poiščite ga na spletu ter ga preberite – literatura!*
  - ★ *izziv: poiščite še ostale RFC dokumente, ki se ukvarjajo z bootp ter preverite, kaj piše v njih.*
- ✘ koračni pogovor med odjemalcem in strežnikom: odjemalec vpraša in strežnik odgovori
- ✘ lahko je hkrati prisotnih več strežnikov in lahko hkrati več odjemalcev želi naložiti OS



# BOOTP – NEKAJ PODROBNOSTI

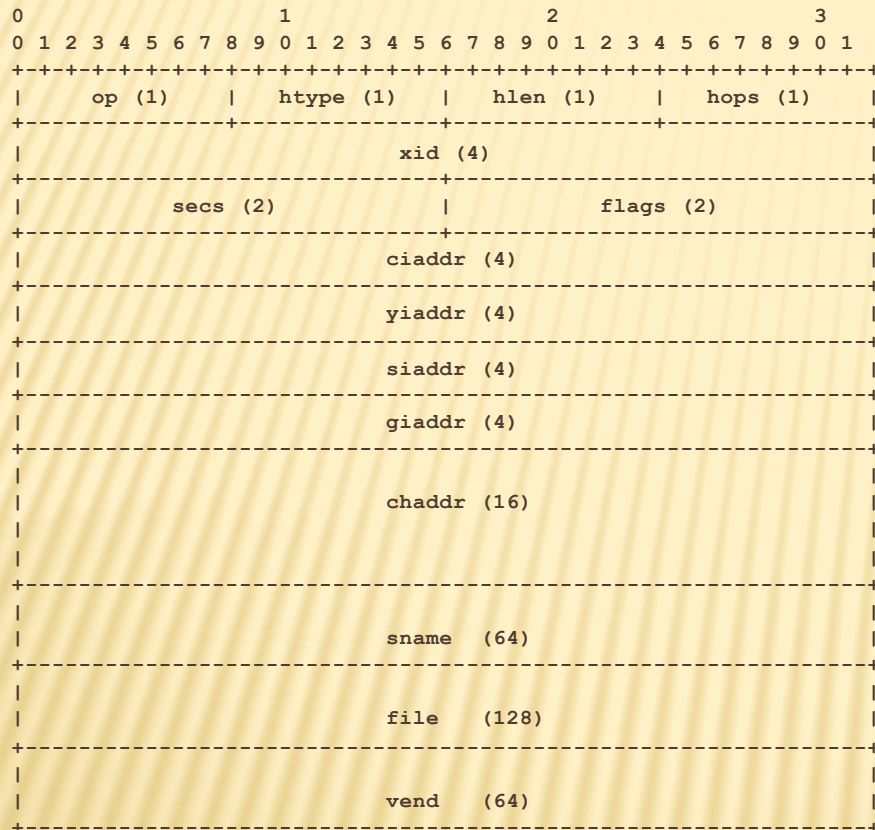
- ✘ Odjemalec na začetku ne pozna IP naslova strežnika, zato razpošlje (*broadcast*) na 2. plasti na lokalni mreži željo po nalaganju OS
- ✘ Strežnik dodeli odjemalcu IP naslov (ali pa ne) ter mu sporoči, kje se nahaja odjemalčev OS
  - + ni nujno, da na lokalni mreži
- ✘ bootp je aplikacija, ki na prenosni plasti uporablja nepovezavni način – UDP protokol
- ✘ Tukaj se pogovor zaključi
  - ✘ Izziv: kako je z varnostjo in trojanskimi konji? Preverite RCPje.

# BOOTP – OBLIKA PAKETA



- op: zahteva ali odgovor
- htype: vrsta medija
- hlen: dolžina naslova
- chaddr: odjemalčev naslov plasti 2
- hops: število skokov
- xid: id zahteve
- secs: koliko časa je minilo od prvega pošiljanja
- flags: zastavice – samo razpošiljanje ali ne

# BOOTP – OBLIKA PAKETA



- ciaddr: odjemalčev naslov
- yiaddr: nastavljen naslov
- siaddr: strežnikov naslov
- giaddr: naslov prehoda
- sname: ime strežnika z OS
- file: datoteka z OS
- vend: možne razširitve
  - izziv: zajemite oba paketa na mreži ter ju komentirajte

# PROGRAMSKA OPREMA

- ✘ na FreeBSD: bootpd in bootpgw
- ✘ konfiguracija v /etc/bootptab
  - izziv: poiščite priročnik ter samo nastavite datoteko ter poženite strežnik in prehodni strežnik.

```
client.test.net:\
:ht=ether:\
:ha=CCCCCCCCCCCC:\
:sm=255.255.255.0:\
:lg=192.168.1.5:\
:ip=192.168.1.10:\
:hn:\
:bf=[/tftpboot/]OS:\
:bs=auto:\
:rp=/export/client/root/:\
:vm=auto:
:vm=rfc1048:
```

- izziv: zgornji zapis uporablja posebno obliko zapisa podatkov – format. Ali se še kje uporablja? Kako je točno definirana? Kako izgleda vmesnik v C-ju za branje? Je zgornji zapis brez napak?

# PROTOKOL TFTP

---

- ✘ definiran v RFC 1350, The TFTP Protocol (*Trivial File Transfer Protocol*)
  - \* **obvezno: poiščite ga na spletu ter ga preberite – literatura!**
  - \* **izziv: poiščite še ostale RFC dokumente, ki se ukvarjajo s tftp ter preverite, kaj piše v njih.**
- ✘ 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
  - \* **izziv: kaj je to sindrom čarovnikovega pomočnika (SAS)? Kje in kako to zadeva tftp?**

# TFTP – NEKAJ PODROBNOSTI

---

- ✘ 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?

# TFTP – 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?)
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

# TFTP – OBLIKA PAKETA

## RRQ, WRQ:

2 bytes	string	1 byte	string	1 byte
-----				
Opcode	Filename	0	Mode	0
-----				

## DATA:

2 bytes	2 bytes	n bytes
-----		
Opcode	Block #	Data
-----		

## ACK:

2 bytes	2 bytes
-----	
Opcode	Block #
-----	

- Opcode: zahteva
- Filename 0: ime datoteke
- Mode 0: oblika zapisa podatkov
- Block #: številčenje poslanih paketov
  
- izziv: zajemite pakete na mreži ter jih komentirajte



# PROGRAMSKA OPREMA

---

- ✘ na FreeBSD: tftpd
- ✘ ni konfiguracijske datoteke
- ✘ datoteke, ki jih streže so v imeniku /tftpboot
- ✘ primer celovite komunikacije nalaganja OS na [www.eventhelix.com/RealtimeMantra/Networking/Bootp.pdf](http://www.eventhelix.com/RealtimeMantra/Networking/Bootp.pdf)
  - izziv: poiščite priročnik ter namestite ftpt strežnik s poljubnimi datotekami. tftp ne dovoli v imenu datoteke nizov oblike „.../” ali „/../” – čemu?

# 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
- ✘ Protokol bootp v prvem koraku odjemalcu pošlje tudi podatke za nastavitev IP naslova in nastavitev IP naslova prehoda
  - + ideja!! – uporabimo bootp protokol

# BOOTP PROTOKOL ZA PRIKLOP NA OMREŽJE

- ✘ Ideja ni slaba, le težave:
  - + poleg IP naslova, potrebujemo še naslov prehoda, naslov DNS strežnika, naslov vmesnega (*proxy*) strežnika, ...
- ✘ Uporabimo / spremenimo namen polja *vend* v bootp protokolu

# RAZŠIRITVE VEND

- ✘ definirane v RFC 1497, BOOTP Vendor Information Extensions
  - ✘ **obvezno: poiščite ga na spletu ter ga preberite – literatura!**
  - ✘ **izziv: poiščite še ostale RFC dokumente, ki se ukvarjajo s to vsebino ter preverite, kaj piše v njih.**
- ✘ prva vrednost je „čarobni piškot“ (*magic cookie*) z vrednostjo 99.130.83.99
- ✘ dve vrsti polj (po dolžini):
  - + stalna: zlog 1: značka [podatki]
    - ✘ Subnet Mask Field (značka: 1, podatki: 4 zlogi): 1.255.255.255.0
  - + spremenljiva: zlog 1: značka, zlog 2: dolžina podatkov, ostali zlogi: podatki
    - ✘ Gateway Field (značka: 3, podatki: N/4 naslovov): 3.4.1.2.3.4
- ✘ značke 128-254: lokalne razširitve
  - ✘ **izziv: uporabite bootp in dodajte svojo lastno razširitev.**

# PROTOKOL DHCP

---

- ✘ obstajata različici za IPv4 in IPv6, najprej IPv4
- ✘ definiran v RFC 2131, **Dynamic Host Configuration Protocol**
  - ★ *obvezno: poiščite ga na spletu ter ga preberite – literatura!*
  - ★ *izziv: poiščite še ostale RFC dokumente, ki se ukvarjajo z DHCP ter preverite, kaj piše v njih.*
- ✘ dejansko razširitev bootp protokola
  - + preimenovanje *vend* polja v *options* in njegovo podaljšanje – RFC 2132, *DHCP Options and BOOTP Vendor Extension*

# DHCP – NEKAJ PODROBNOSTI

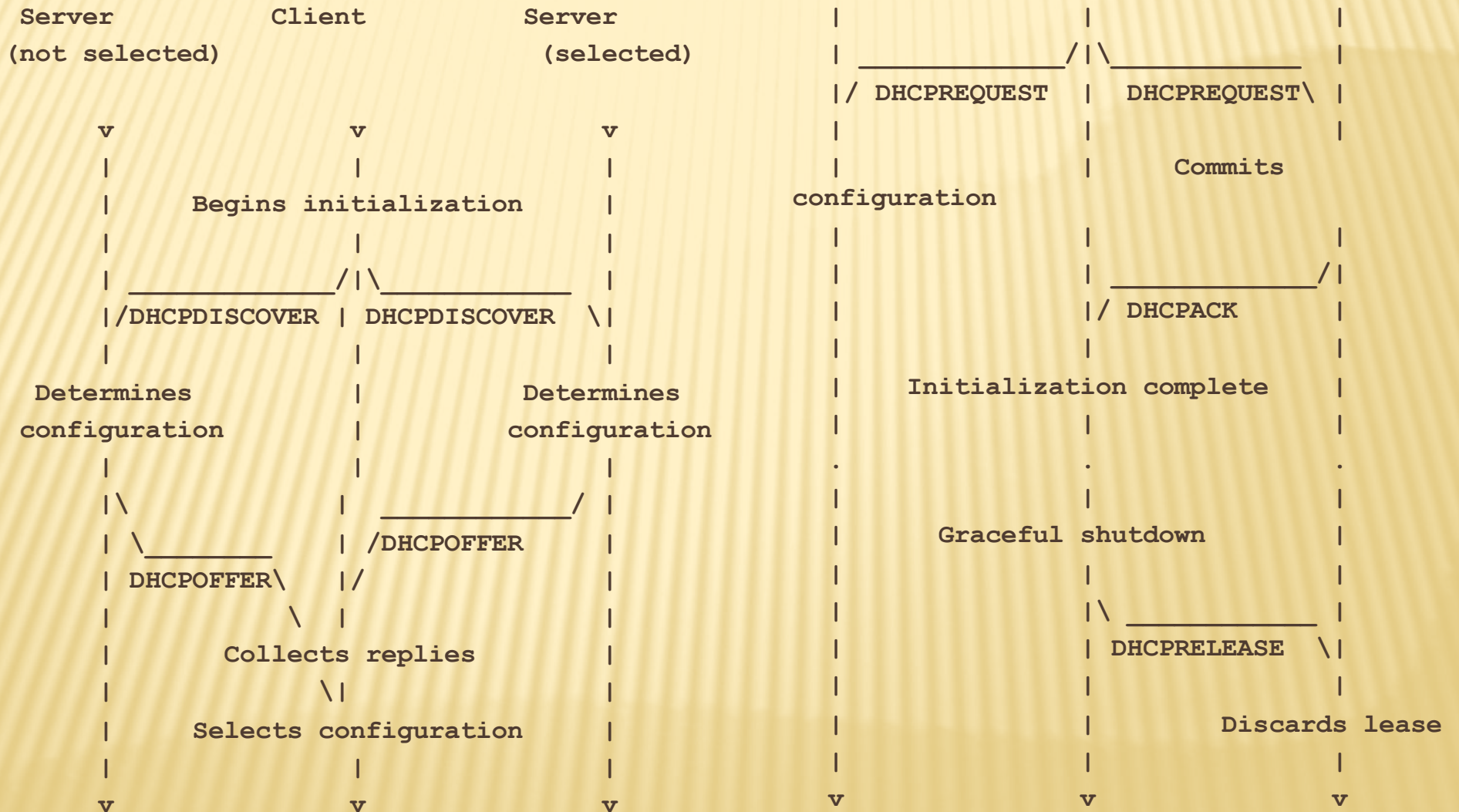
---

- ✘ Odjemalec na začetku ne pozna IP naslova strežnika
- ✘ DHCP je aplikacija, ki na prenosni plasti uporablja nepovezavni način – UDP protokol
  - ✘ Izziv: kako je z varnostjo pri DHCP protokolu? Če se da, naredite napad na odjemalca

# DHCP – JEDRO PROTOKOLA

- ✘ 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 sosoje 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
- ✘ posebna značka v *options*: *DHCP message type*
  - ✘ izziv: kakšno vrednost ima ta značka?

# DHCP – ŽIVLJENJSKI CIKEL





# DHCP NEVARNOSTI

---

- × DHCP ne predvideva avtentikacije
- × možni napadi:
  - + neavtorizirani strežniki posredujejo napačno informacijo
  - + neavtorizirani odjemalci pridobijo dostop do virov, do katerih bi ne smeli
  - + izpraznenje virov s strani neavtoriziranih odjemalcev
    - × izziv: izvedite vsaj enega od zgornjih napadov. O čem govori RFC 3118 in kako deluje?

# PROGRAMSKA OPREMA

- ✘ na FreeBSD odjemalec *dhclient* s konfiguracijsko datoteko `/etc/dhclient.conf`
- ✘ glej:  
[www.freebsd.org/doc/handbook/network-dhcp.html](http://www.freebsd.org/doc/handbook/network-dhcp.html)
  - izziv: skonfigurirajte odjemalca in ga poženite. Kaj pravzaprav pomeni desna konfiguracija?

```
send host-name "andare.fugue.com";
send dhcp-client-identifier 1:0:a0:24:ab:fb:9c;
send dhcp-lease-time 3600;
supersede domain-name "fugue.com home.vix.com";
prepend domain-name-servers 127.0.0.1;
request subnet-mask, broadcast-address, time-offset, routers,
        domain-name, domain-name-servers, host-name;
require subnet-mask, domain-name-servers;
timeout 60;
retry 60;
reboot 10;
select-timeout 5;
initial-interval 2;
script "/etc/dhclient-script";
media "-link0 -link1 -link2", "link0 link1";
reject 192.33.137.209;

alias {
    interface "ep0";
    fixed-address 192.5.5.213;
    option subnet-mask 255.255.255.255;
}

lease {
    interface "ep0";
    fixed-address 192.33.137.200;
    medium "link0 link1";
    option host-name "andare.swiftmedia.com";
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.33.137.255;
    option routers 192.33.137.250;
    option domain-name-servers 127.0.0.1;
    renew 2 2000/1/12 00:00:01;
    rebind 2 2000/1/12 00:00:01;
    expire 2 2000/1/12 00:00:01;
}
```

# PROGRAMSKA OPREMA

- ✘ na FreeBSD strežnik *net/isc-dhcp31-server* s konfiguracijsko datoteko `/usr/local/etc/dhcpd.conf`

- izziv: skonfigurirajte strežnik in ga poženite. Kaj počne program `dhcp_probe` – namestite ga in ga poženite.

```
option domain-name "example.com";
option domain-name-servers 192.168.4.100;
option subnet-mask 255.255.255.0;

default-lease-time 3600;
max-lease-time 86400;
ddns-update-style none;

subnet 192.168.4.0 netmask 255.255.255.0 {
    range 192.168.4.129 192.168.4.254;
    option routers 192.168.4.1;
}

host mailhost {
    hardware ethernet 02:03:04:05:06:07;
    fixed-address mailhost.example.com;
}
```

# PROTOKOL DHCPV6

---

- ✘ definiran v RFC 3315, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
  - \* **obvezno: poiščite ga na spletu ter ga preberite – literatura!**
  - \* **izziv: poiščite še ostale RFC dokumente, ki se ukvarjajo z DHCP ter preverite, kaj piše v njih.**
- ✘ povsem drugačen protokol za IPv6
- ✘ dva načina konfiguracije računalnika:
  - + brezstanjsko (*stateless*), kjer se računalnik lahko sam nastavi; in
  - + stanjsko (*statefull*), kjer računalnik nastavi s pomočju drugih enot

# DHCPV6 – NEKAJ PODROBNOSTI

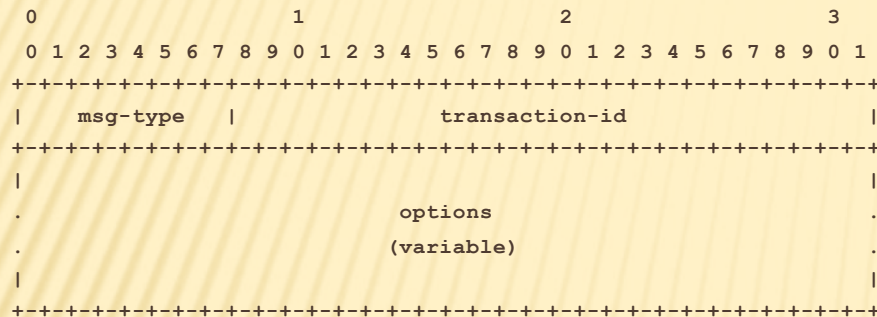
---

- ✘ Odjemalec na začetku ne pozna IP naslova strežnika
- ✘ DHCP je aplikacija, ki na prenosni plasti uporablja nepovezavni način – UDP protokol

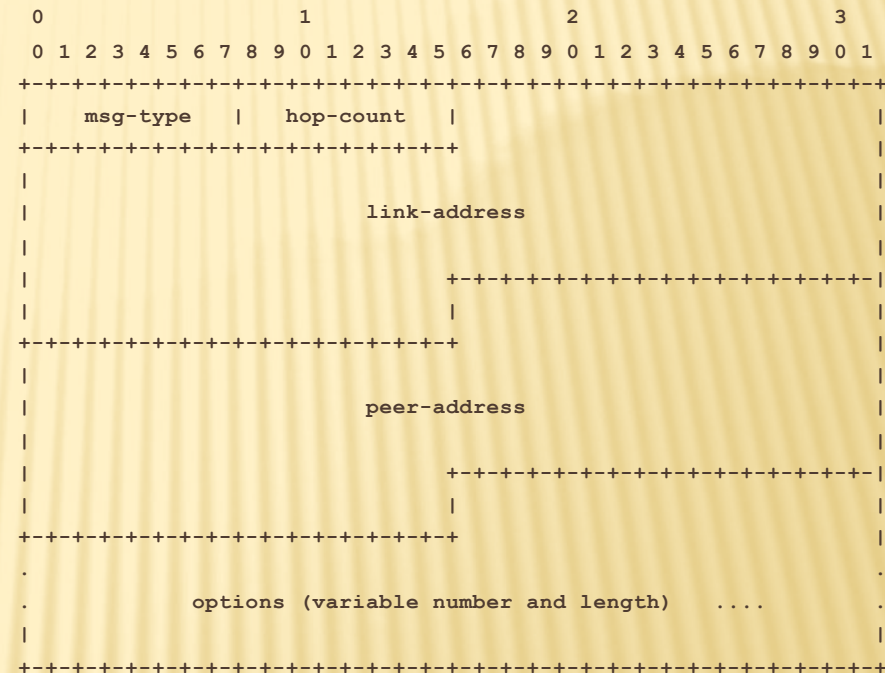
# DHCPV6 – JEDRO PROTOKOLA

- ✘ možne zahteve (*msg-type*):
  - + SOLICIT: prošnja za nastavitve
  - + 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
    - ✘ **izziv: kako deluje prepošiljanje zahtev?**

# DHCPV6 – OBLIKA SPOROČIL



običajno sporočilo



preposlano sporočilo

- ✘ izziv: kakšne možnosti (*options*) obstajajo? Kam so šla polja iz IPv4? Kaj je to DUID?

# PROGRAMSKA OPREMA

- ✘ na FreeBSD odjemalec, strežnik in prepošiljatelj *dhcp6* s konfiguracijsko datoteko  
`/usr/local/etc/dhcp6{c,s}.conf`

- izziv: skonfigurirajte odjemalca in ga poženite. Kaj pravzaprav pomeni desna konfiguracija?

```
option domain-name-servers 2001:db8::35;  
interface fxp0 { address-pool pool1 3600; };  
pool pool1 { range 2001:db8:1:2::1000 to 2001:db8:1:2::2000 ; };
```

nastavitvena datoteka strežnika



# ZAKLJUČEK

---

- ✘ ogledali smo si, kako se lahko računalnik obuje z mreže in
- ✘ kako se lahko priklopi na omrežje
- ✘ Naslednjič: upravljanje z omrežji