Network Layer (Private/Public Addressing, NAT and DHCP)

Fundamentos de Redes

Mestrado Integrado em Engenharia de Computadores e Telemática

DETI-UA, 2019/2020

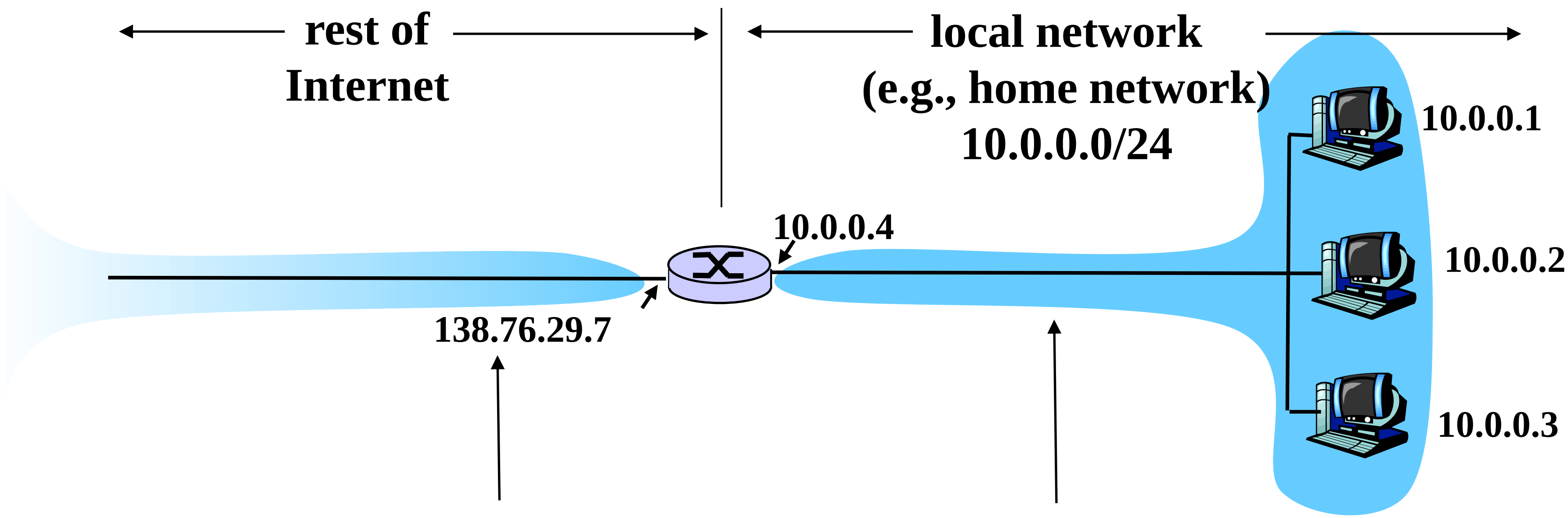
PRIVATE ADDRESSING

Blocks of private addresses

Prefix	Lowest address	Highest address
10/8	10.0.0.0	10.255.255.255
172.16/12	172.16.0.0	172.31.255.255
192.168/16	192.168.0.0	192.168.255.255
169.254/16	169.254.0.0	169.254.255.255

- These addresses can be used freely in private networks
- IP packets with destination addresses belonging to these blocks are not routed in the public network
- For communications with the Internet, the private addresses must be translated into public addresses

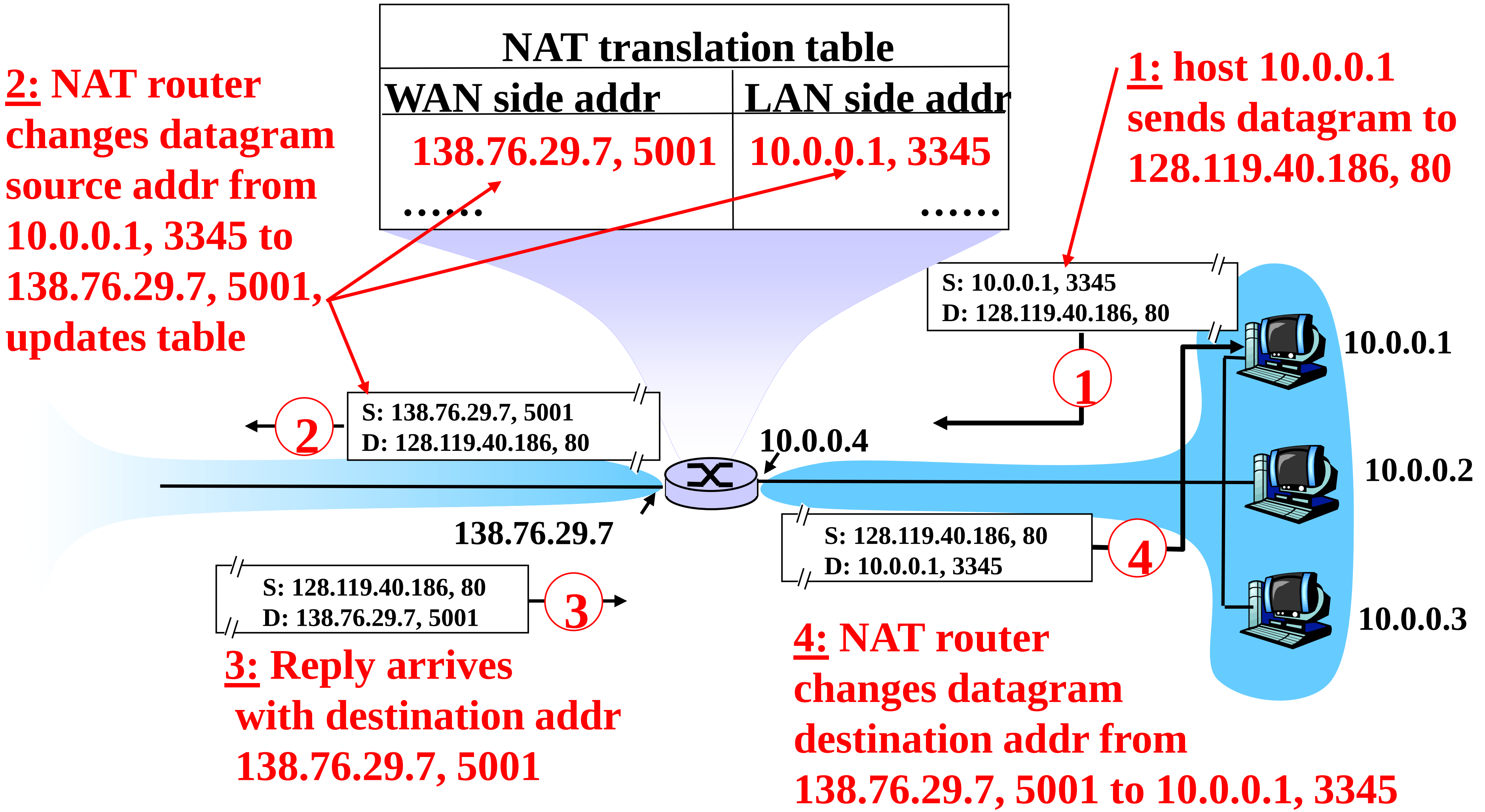
NAT: Network Address Translation



All **datagrams** ***leaving*** **local network have same single source NAT IP address: 138.76.29.7, different source port numbers**

Datagrams with source or destination in this network have 10.0.0.0/24 address for source, destination (as usual)

NAT: Network Address Translation



NAT: Network Address Translation

Implementation: NAT router must:

- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr.
- *remember (in NAT translation table)* every (source IP address, port #) to (NAT IP address, new port #) translation pair
- *incoming datagrams: replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

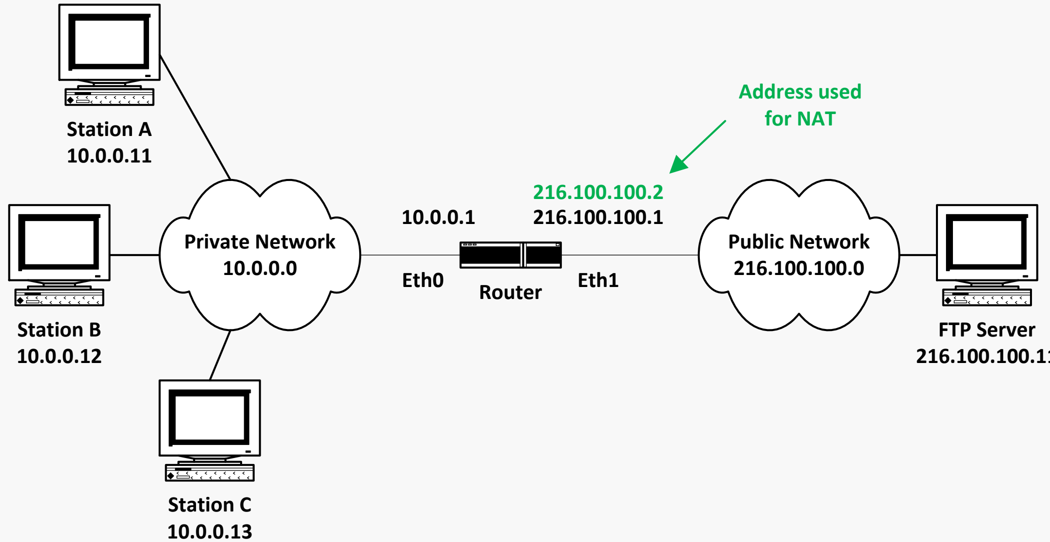
NAT: Network Address Translation

- Motivation: local network uses just one IP address as far as outside world is concerned:
  - range of addresses not needed from ISP: just one (or a few) IP address for all devices
  - can change addresses of devices in local network without notifying outside world
  - can change ISP without changing addresses of devices in local network
  - devices inside local net not explicitly addressable, visible by outside world (a security additional advantage).

NAT: Network Address Translation

- More than one public IP address may be available (NAT versus PAT)
- 16-bit port-number field:
  - more than 60,000 simultaneous connections with a single public IP address!
- NAT is controversial:
  - routers should only process up to layer 3
  - violates end-to-end argument
    - NAT possibility must be taken into account by application designers, e.g, P2P applications
  - address shortage should instead be solved by IPv6

Example (I)



Access of A and B to the FTP server :

Exp6RI_c.cap : 1/22 Ethernet packets

No.	Sta.	Source Address	Dest Address	Layer	Summary
1	Ok	10.0.0.12	216.100.100.11	TCP	1032->File T
2	Ok	216.100.100.11	10.0.0.12	TCP	File Transfe
3	Ok	10.0.0.12	216.100.100.11	TCP	1032->File T
4	Ok	216.100.100.11	10.0.0.12	FTP	220 Serv-U F
5	Ok	10.0.0.12	216.100.100.11	TCP	1032->File T
6	Ok	10.0.0.12	216.100.100.11	FTP	USER anonymo
7	Ok	216.100.100.11	10.0.0.12	FTP	331 User nar

Exp6RI_c.cap : 12/22 Ethernet packets

No.	Sta.	Source Address	Dest Address	Layer	Summary
12	Ok	10.0.0.11	216.100.100.11	TCP	1033->File
13	Ok	216.100.100.11	10.0.0.11	TCP	File Transf
14	Ok	10.0.0.11	216.100.100.11	TCP	1033->File
15	Ok	216.100.100.11	10.0.0.11	FTP	220 Serv-U
16	Ok	10.0.0.11	216.100.100.11	TCP	1033->File
17	Ok	10.0.0.11	216.100.100.11	FTP	USER anonyr
18	Ok	216.100.100.11	10.0.0.11	FTP	331 User na

Captures in private network

Exp6RE_c.cap : 1/22 Ethernet packets

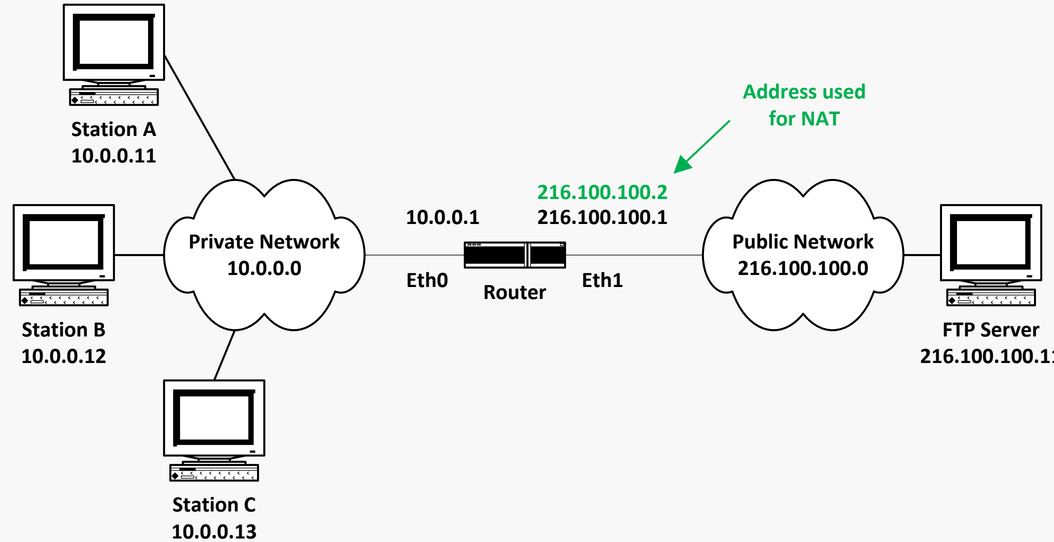
No.	Sta.	Source Address	Dest Address	Layer	Summary
1	Ok	216.100.100.2	216.100.100.11	TCP	1032->File
2	Ok	216.100.100.11	216.100.100.2	TCP	File Transf
3	Ok	216.100.100.2	216.100.100.11	TCP	1032->File
4	Ok	216.100.100.11	216.100.100.2	FTP	220 Serv-U
5	Ok	216.100.100.2	216.100.100.11	TCP	1032->File
6	Ok	216.100.100.2	216.100.100.11	FTP	USER anonyr
7	Ok	216.100.100.11	216.100.100.2	FTP	331 User na

Exp6RE_c.cap : 12/22 Ethernet packets

No.	Sta.	Source Address	Dest Address	Layer	Summary
1	Ok	216.100.100.2	216.100.100.11	TCP	1033->File
1	Ok	216.100.100.11	216.100.100.2	TCP	File Transf
1	Ok	216.100.100.2	216.100.100.11	TCP	1033->File
1	Ok	216.100.100.11	216.100.100.2	FTP	220 Serv-U
1	Ok	216.100.100.2	216.100.100.11	TCP	1033->File
1	Ok	216.100.100.2	216.100.100.11	FTP	USER anonyr
1	Ok	216.100.100.11	216.100.100.2	FTP	331 User na

Captures in public network

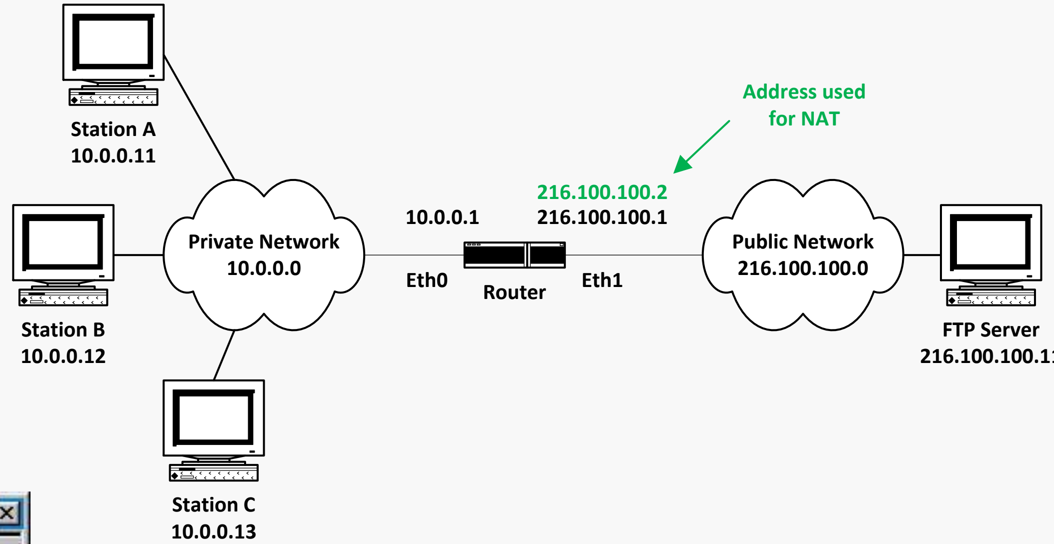
Example (II)



Access of A and B to the FTP server :

```
Router#show ip nat translation verbose
Pro Inside global       Inside local       Outside local       Outside global
tcp 216.100.100.2:1032 10.0.0.12:1032     216.100.100.11:21   216.100.100.11:21
    create 00:00:35, use 00:00:24, left 23:59:35,
    flags:
extended, use_count: 0
tcp 216.100.100.2:1033 10.0.0.11:1033     216.100.100.11:21   216.100.100.11:21
    create 00:00:12, use 00:00:06, left 23:59:53,
    flags:
extended, use_count: 0
```

Example (III)



Second access of B to the FTP server :

Exp6RI_c2.cap : 1/11 Ethernet packets

No.	Sta.	Source Address	Dest Address	Layer	Summary
1	Ok	10.0.0.12	216.100.100.11	TCP	1033->File
2	Ok	216.100.100.11	10.0.0.12	TCP	File Trans
3	Ok	10.0.0.12	216.100.100.11	TCP	1033->File
4	Ok	216.100.100.11	10.0.0.12	FTP	220 Serv-U
5	Ok	10.0.0.12	216.100.100.11	TCP	1033->File
6	Ok	10.0.0.12	216.100.100.11	FTP	USER anony
7	Ok	216.100.100.11	10.0.0.12	FTP	331 User n

private network

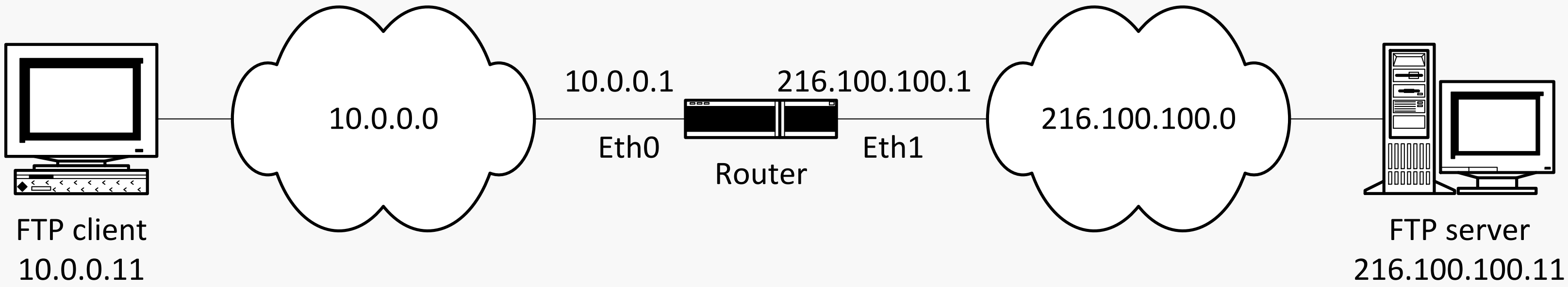
Exp6RE_c2.cap : 1/11 Ethernet packets

No.	Sta.	Source Address	Dest Address	Layer	Summary
1	Ok	216.100.100.2	216.100.100.11	TCP	1024->File
2	Ok	216.100.100.11	216.100.100.2	TCP	File Trans
3	Ok	216.100.100.2	216.100.100.11	TCP	1024->File
4	Ok	216.100.100.11	216.100.100.2	FTP	220 Serv-U
5	Ok	216.100.100.2	216.100.100.11	TCP	1024->File
6	Ok	216.100.100.2	216.100.100.11	FTP	USER anony
7	Ok	216.100.100.11	216.100.100.2	FTP	331 User n

public network

```
Router#show ip nat translation verbose
Pro Inside global       Inside local       Outside local       Outside global
tcp 216.100.100.2:1024 10.0.0.12:1033     216.100.100.11:21   216.100.100.11:21
    create 00:00:49, use 00:00:42, left 23:59:17,
    flags:
extended, use_count: 0
tcp 216.100.100.2:1032 10.0.0.12:1032     216.100.100.11:21   216.100.100.11:21
    create 00:02:42, use 00:02:31, left 23:57:28,
    flags:
extended, use_count: 0
tcp 216.100.100.2:1033 10.0.0.11:1033     216.100.100.11:21   216.100.100.11:21
    create 00:02:18, use 00:02:13, left 23:57:46,
    flags:
extended, use_count: 0
```

Example (IV)



Translation of IP address in FTP message (Layer 5):

```
ETHER-II: 00-60-97-9B-9E-07 ==> 00-D0-58-A9-3E-37
Internet Protocol
  Version(MSB 4 bits): 4
  Header length(LSB 4 bits): 5 (32-bit word)
  Service type: Precd=Routine,Delay=Normal,Thrput=Normal,Reli=Normal
  Total length: 61 (Octets)
  Fragment ID: 14595
  Flags: Do not fragment,Last fragment,Offset=0 (0x00)
  Time to live: 128 seconds/hops
  IP protocol type: TCP (0x06)
  Checksum: 0x7B3D
  IP address 10.0.0.11 ->216.100.100.11
  No option
TCP: 1036->File Transfer (Control),S=82367,A=171360,W=86
File Transfer Protocol
  PORT 10,0,0,11,4,13
Sliced Packet( Data Length = 75)
```

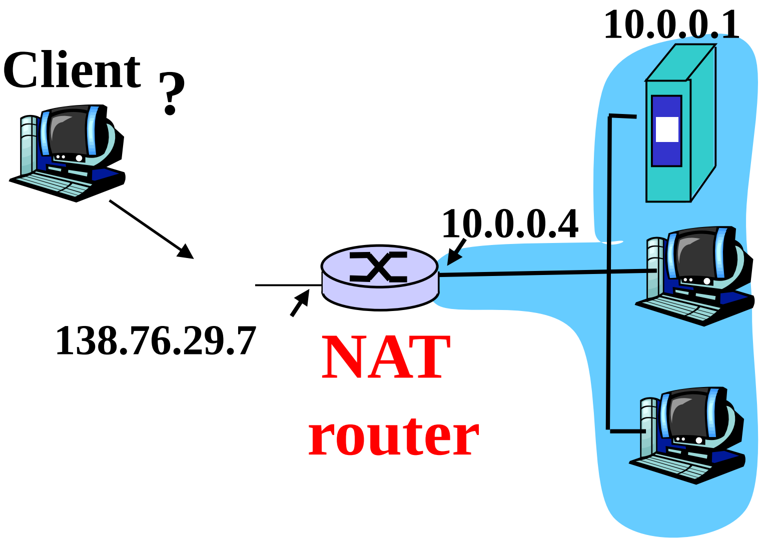
private network

```
ETHER-II: 00-D0-58-A9-3E-38 ==> 00-60-97-D4-9F-9A
Internet Protocol
  Version(MSB 4 bits): 4
  Header length(LSB 4 bits): 5 (32-bit word)
  Service type: Precd=Routine,Delay=Normal,Thrput=Normal,Reli=Normal
  Total length: 65 (Octets)
  Fragment ID: 14595
  Flags: Do not fragment,Last fragment,Offset=0 (0x00)
  Time to live: 127 seconds/hops
  IP protocol type: TCP (0x06)
  Checksum: 0x49DD
  IP address 216.100.100.2 ->216.100.100.11
  No option
TCP: 1036->File Transfer (Control),S=82367,A=171360,W=8610
File Transfer Protocol
  PORT 216,100,100,2,4,13
Sliced Packet( Data Length = 79)
```

public network

NAT traversal problem

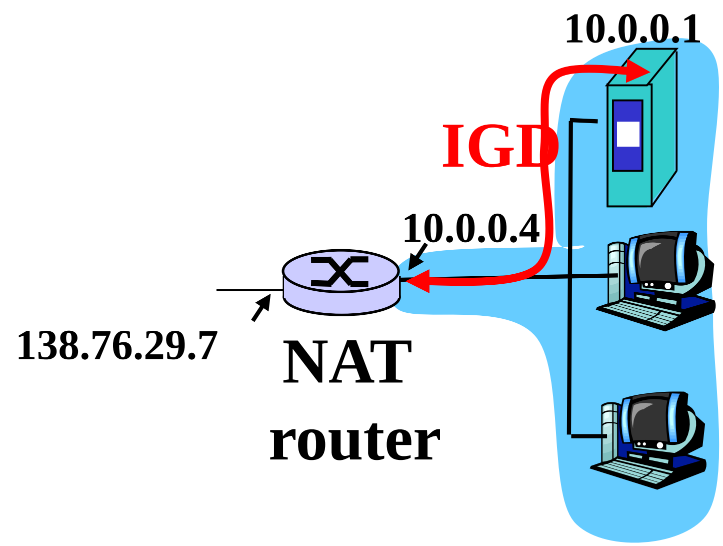
- client wants to connect to server with address 10.0.0.1
  - server address 10.0.0.1 local to LAN (client cannot use it as destination address)
  - only one externally visible NATted address: 138.76.29.7
- solution 1: statically configure NAT to forward incoming connection requests at given port to server
  - e.g., (138.76.29.7, port 2500) always forwarded to 10.0.0.1 port 25000



NAT traversal problem

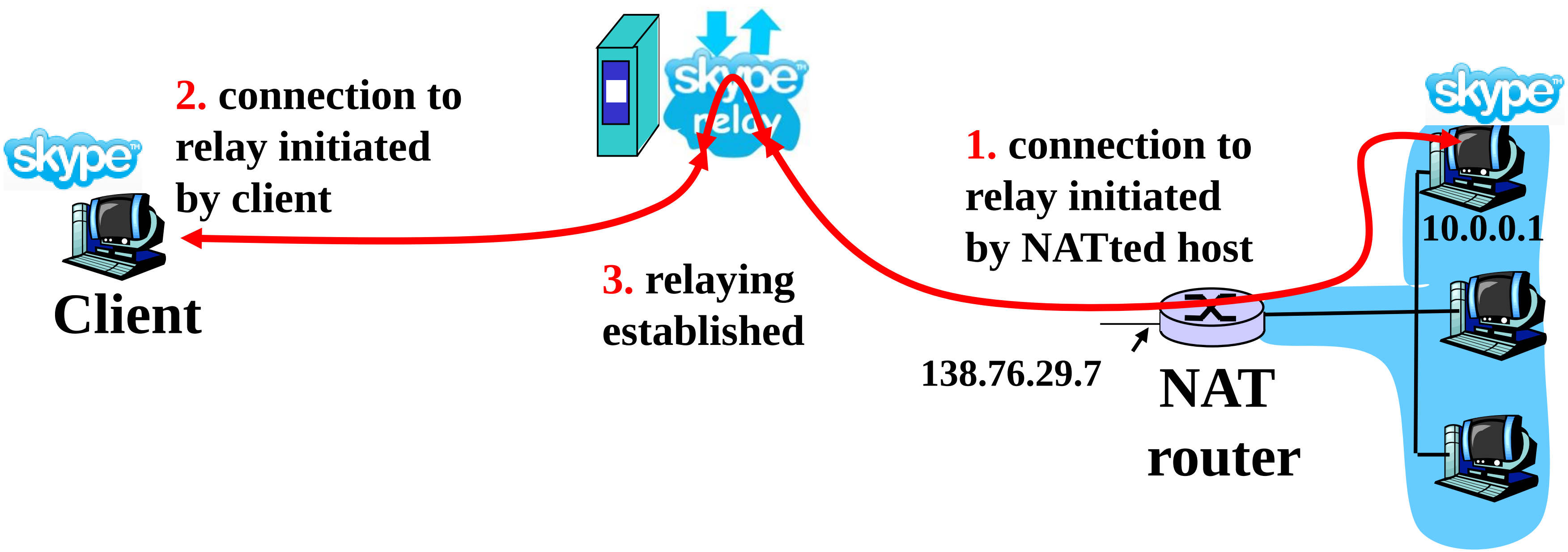
- solution 2: Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Allows NATted host to:
  - Application running on NATted host requests mapping between (private IP address, private port number) and (public IP address, public port number)
  - Application can advertise (public IP address, public port number)

i.e., automate static NAT port map configuration



NAT traversal problem

- solution 3: relaying (used in Skype)
  - NATed server establishes connection to Relay
  - External client connects to Relay
  - Relay bridges packets between two connections



DHCP

DHCP: Dynamic Host Configuration Protocol

Goal: allow host to *dynamically* obtain its IP address from network server when it joins network

- Can renew its lease on address in use
- Allows reuse of addresses (only hold address while connected and "on")
- Support for mobile users who want to join network
- Also allows client to learn subnet mask, default gateway, local DNS server

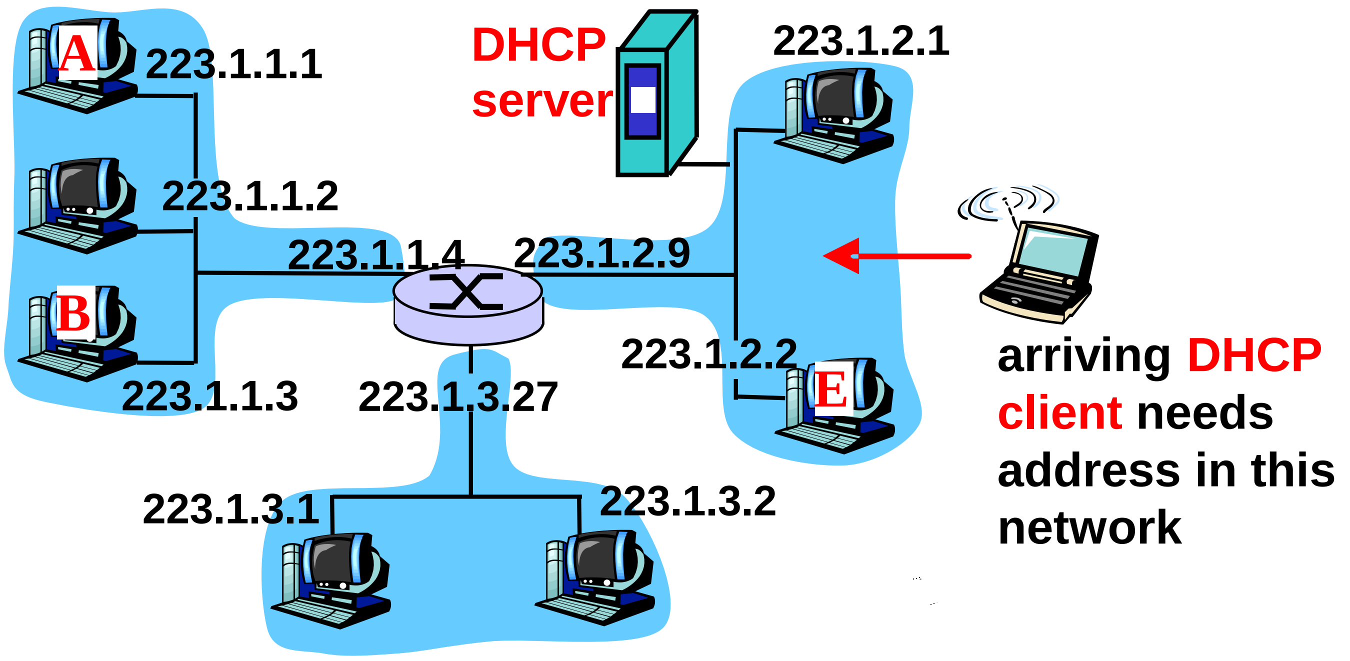
DHCP overview:

- host broadcasts "DHCP discover" message
- DHCP server responds with "DHCP offer" message
- host requests IP address: "DHCP request" message
- DHCP server sends address: "DHCP ack" message

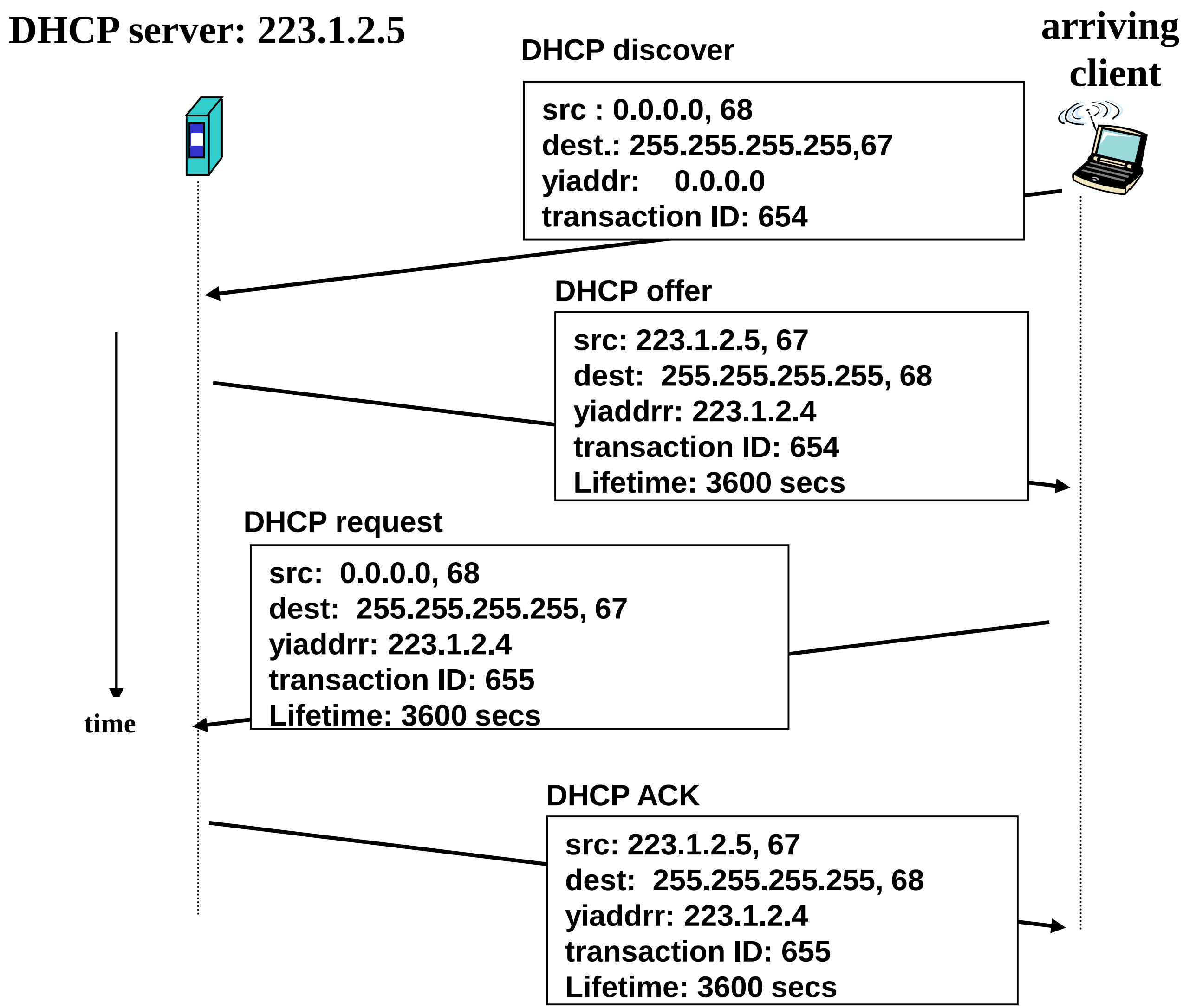
Configuration of a DHCP server

- Lease address range
  - Pool of IP addresses to be leased defined by the first and the last address
- Excluding address range
  - Pool of IP addresses inside the lease address range that are not to be leased
- Reserved addresses
  - IP addresses to be assigned statically for specific MACs
- Lease time
  - Time duration of the lease of an address

DHCP client-server scenario



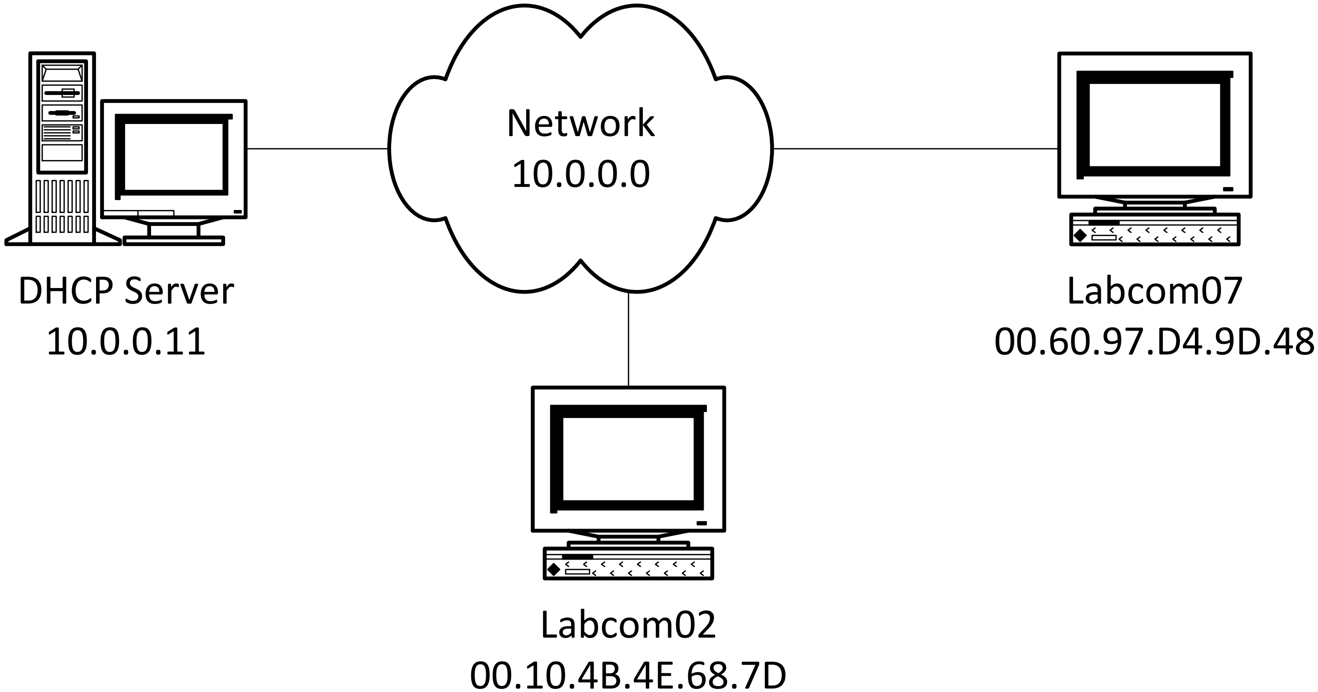
DHCP client-server scenario



DHCP versus BOOTP

- Extension of *Bootstrap Protocol*, BOOTP, (RFC 1542)
  - Run over UDP
  - Server port number: 67
  - Client port number: 68
  - Originally, BOOTP enables a diskless terminal to find its IP address, a server address and a configuration filename to be requested to the server and locally executed.

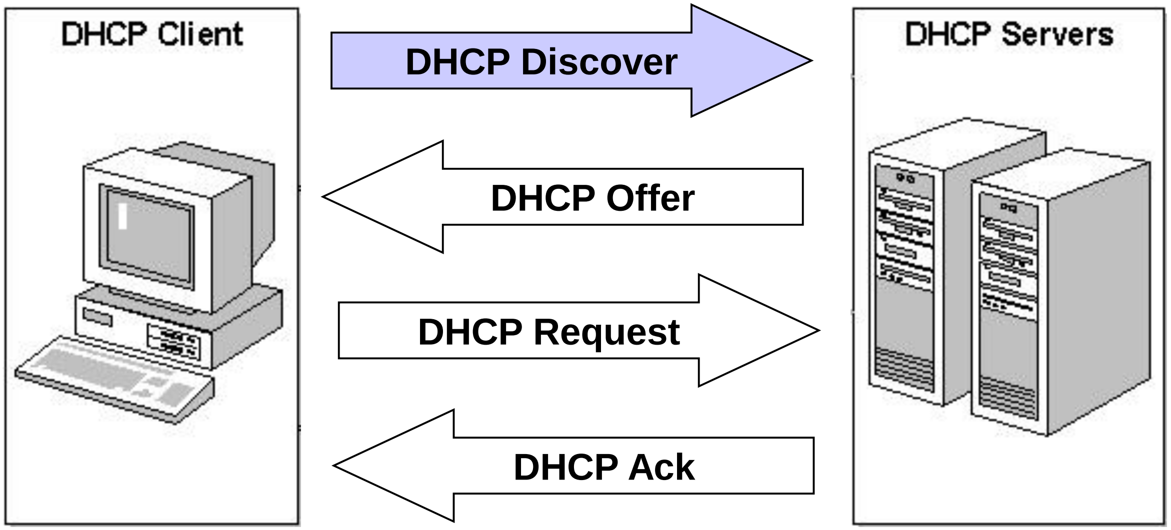
Example



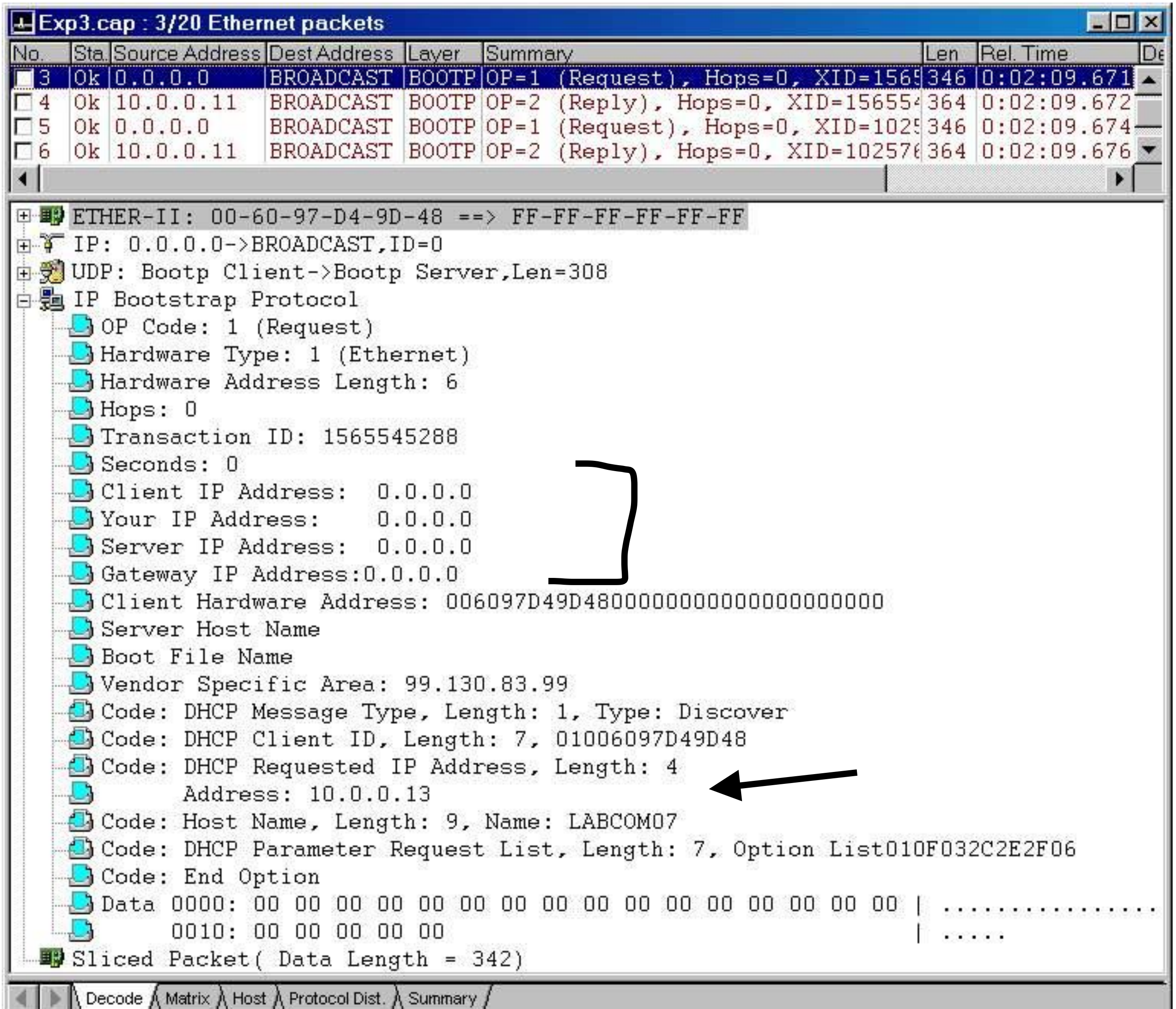
Network
10.0.0.0
DHCP Server
10.0.0.11
Labcom07
00.60.97.D4.9D.48
Labcom02
00.10.4B.4E.68.7D

DHCP Discover

***DHCP Discover* message is encapsulated on a *BootP Request* message. It is sent to discover available DHCP servers. The client can also indicate an IP address.**

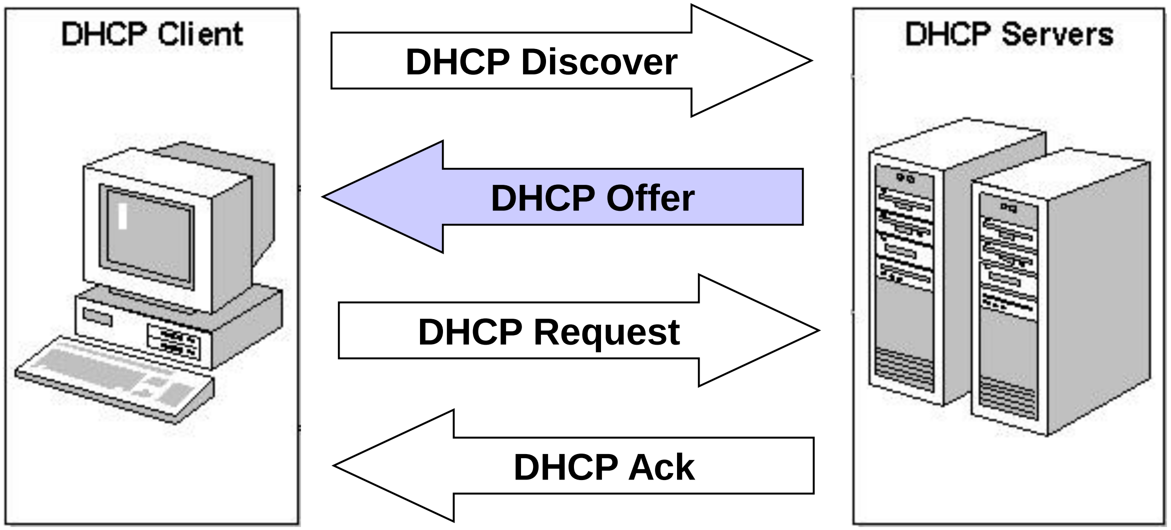


DHCP Discover



DHCP Offer

DHCP Offer **message is encapsulated on a** ***BootP Reply*** **message. Each server offers one IP address to lease <u>(if possible, servers offer the IP address indicated by the client).</u>**

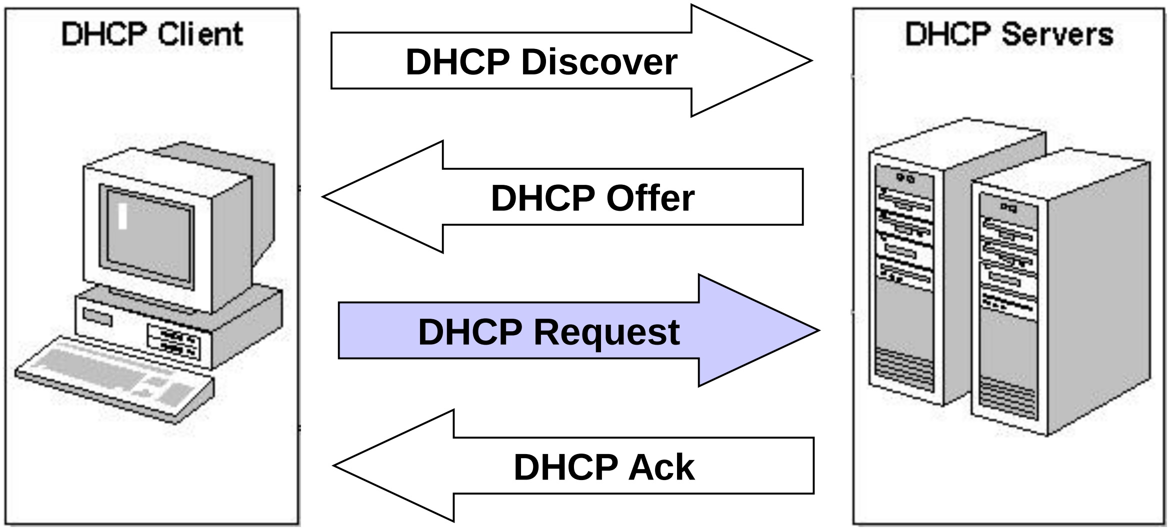


DHCP Offer

```
ETHER-II: 00-60-97-9B-9B-3F ==> FF-FF-FF-FF-FF-FF
IP: 10.0.0.11->BROADCAST,ID=4250
UDP: Bootp Server->Bootp Client,Len=326
IP Bootstrap Protocol
    OP Code: 2 (Reply)
    Hardware Type: 1 (Ethernet)
    Hardware Address Length: 6
    Hops: 0
    Transaction ID: 1565545288
    Seconds: 0
    Client IP Address:  0.0.0.0
    Your IP Address:    10.0.0.13
    Server IP Address:  10.0.0.11
    Gateway IP Address:0.0.0.0
    Client Hardware Address: 006097D49D480000000000000000000
    Server Host Name
    Boot File Name
    Vendor Specific Area: 99.130.83.99
    Code: DHCP Message Type, Length: 1, Type: Offer
    Code: Subnet Mask, Length: 4      Address255.0.0.0
    Code: DHCP Renewal (T1) Time, Length: 4, Value:150
    Code: DHCP Rebinding  (T2) Time, Length: 4, Value:262
    Code: DHCP IP Address Lease Time, Length: 4, Value:300
    Code: DHCP Server ID, Length: 4
          Address: 10.0.0.11
    Code: Domain Name, Length: 17, Name: labcom.det.ua.pt
    Code: Router, Length: 4
          Address: 10.0.0.1     router
    Code: NetBIOS Name Server, Length: 8
          Address: 193.136.173.202
          Address: 193.136.173.203
    Code: NetBIOS over TCP/IP, Length: 1, Node Type:0x8 H-node
    Code: Domain Name Server, Length: 4
          Address: 10.0.0.11      domain name server
    Code: End Option
Sliced Packet( Data Length = 360)
```

DHCP Request

DHCP Request **message is encapsulated on a** ***BootP Request*** **message. After selecting one of the offers, the client indicates the selected IP address.**

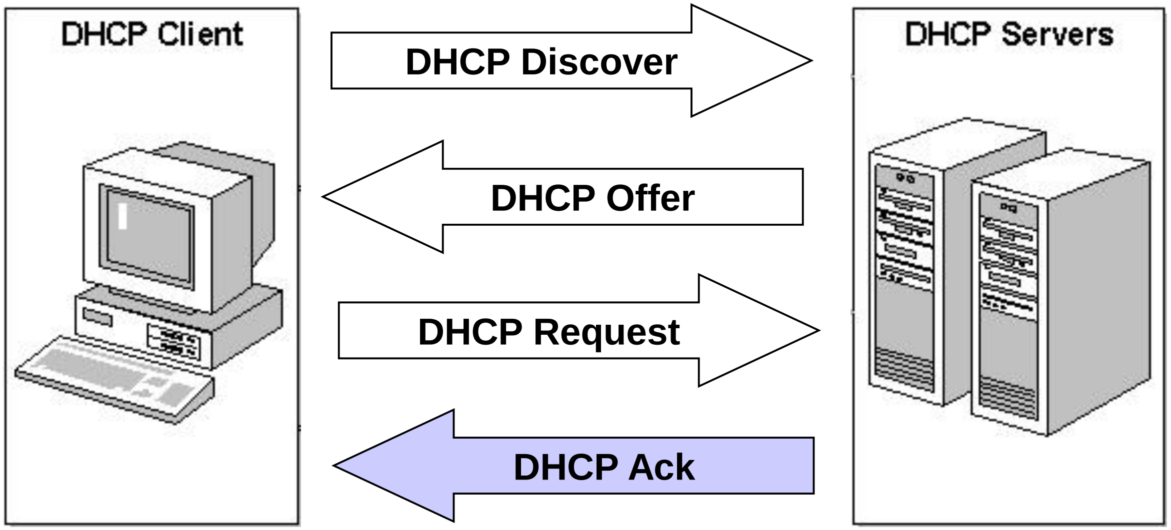


DHCP Request

```
⊞ ETHER-II: 00-60-97-D4-9D-48 ==> FF-FF-FF-FF-FF-FF
⊞ IP: 0.0.0.0->BROADCAST,ID=256
⊞ UDP: Bootp Client->Bootp Server,Len=308
⊟ IP Bootstrap Protocol
    OP Code: 1 (Request)
    Hardware Type: 1 (Ethernet)
    Hardware Address Length: 6
    Hops: 0
    Transaction ID: 1025766432
    Seconds: 0
    Client IP Address:  0.0.0.0
    Your IP Address:    0.0.0.0
    Server IP Address:  0.0.0.0
    Gateway IP Address:0.0.0.0
    Client Hardware Address: 006097D49D48000000000000000000000
    Server Host Name
    Boot File Name
    Vendor Specific Area: 99.130.83.99
    Code: DHCP Message Type, Length: 1, Type: Request
    Code: DHCP Client ID, Length: 7, 01006097D49D48
    Code: DHCP Requested IP Address, Length: 4
          Address: 10.0.0.13
    Code: DHCP Server ID, Length: 4
          Address: 10.0.0.11
    Code: Host Name, Length: 9, Name: LABCOM07
    Code: DHCP Parameter Request List, Length: 7, Option List010F032C2E2F06
    Code: End Option
    Data 0000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    | ...............
  Sliced Packet( Data Length = 342)
```

DHCP Ack

***DHCP Ack* message is encapsulated in a *BootP Reply* message. The server acknowledges positively the lease of the IP address indicating also other information of interest.**



DHCP Ack

```
ETHER-II: 00-60-97-9B-9B-3F ==> FF-FF-FF-FF-FF-FF
IP: 10.0.0.11->BROADCAST,ID=4251
UDP: Bootp Server->Bootp Client,Len=326
IP Bootstrap Protocol
    OP Code: 2 (Reply)
    Hardware Type: 1 (Ethernet)
    Hardware Address Length: 6
    Hops: 0
    Transaction ID: 1025766432
    Seconds: 0
    Client IP Address:  0.0.0.0
    Your IP Address:    10.0.0.13
    Server IP Address:  0.0.0.0
    Gateway IP Address:0.0.0.0
    Client Hardware Address: 006097D49D480000000000000000000
    Server Host Name
    Boot File Name
    Vendor Specific Area: 99.130.83.99
    Code: DHCP Message Type, Length: 1, Type: Ack
    Code: DHCP Renewal (T1) Time, Length: 4, Value:150
    Code: DHCP Rebinding  (T2) Time, Length: 4, Value:262
    Code: DHCP IP Address Lease Time, Length: 4, Value:300
    Code: DHCP Server ID, Length: 4
          Address: 10.0.0.11
    Code: Subnet Mask, Length: 4      Address255.0.0.0
    Code: Domain Name, Length: 17, Name: labcom.det.ua.pt
    Code: Router, Length: 4
          Address: 10.0.0.1
    Code: NetBIOS Name Server, Length: 8
          Address: 193.136.173.202
          Address: 193.136.173.203
    Code: NetBIOS over TCP/IP, Length: 1, Node Type:0x8 H-node
    Code: Domain Name Server, Length: 4
          Address: 10.0.0.11
    Code: End Option
Sliced Packet( Data Length = 360)
```

repara, que o server, desaoparece

Address leasing

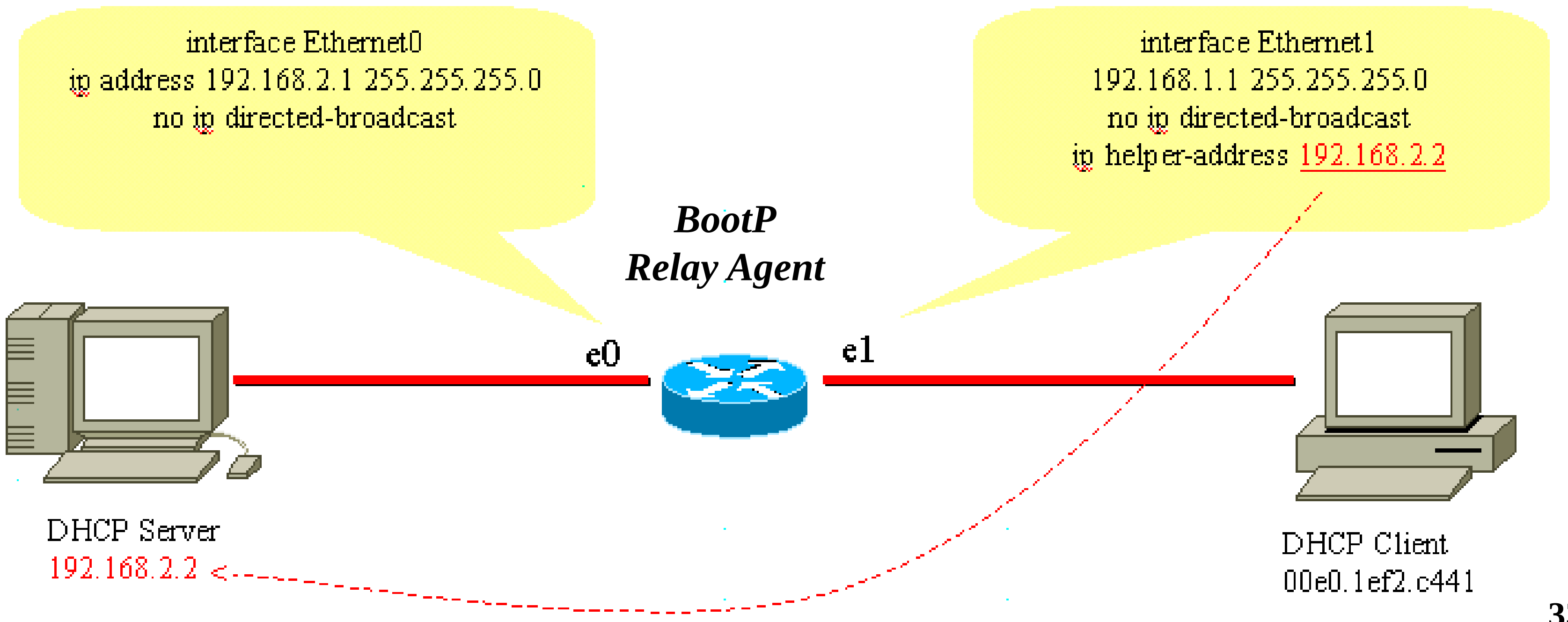
- *Renewal Time (50% of Lease Time)*
  - at this time, the client should try to renew the lease in the server that has assigned its IP address
- *Rebinding Time (85% of Lease Time)*
  - at this time, the client should try to renew the lease of its IP address (if it didn't succeed before) in any available server
- *Lease Time*
  - if the lease could not be renewed, at this time, the client stop using the IP address

Other DHCP messages

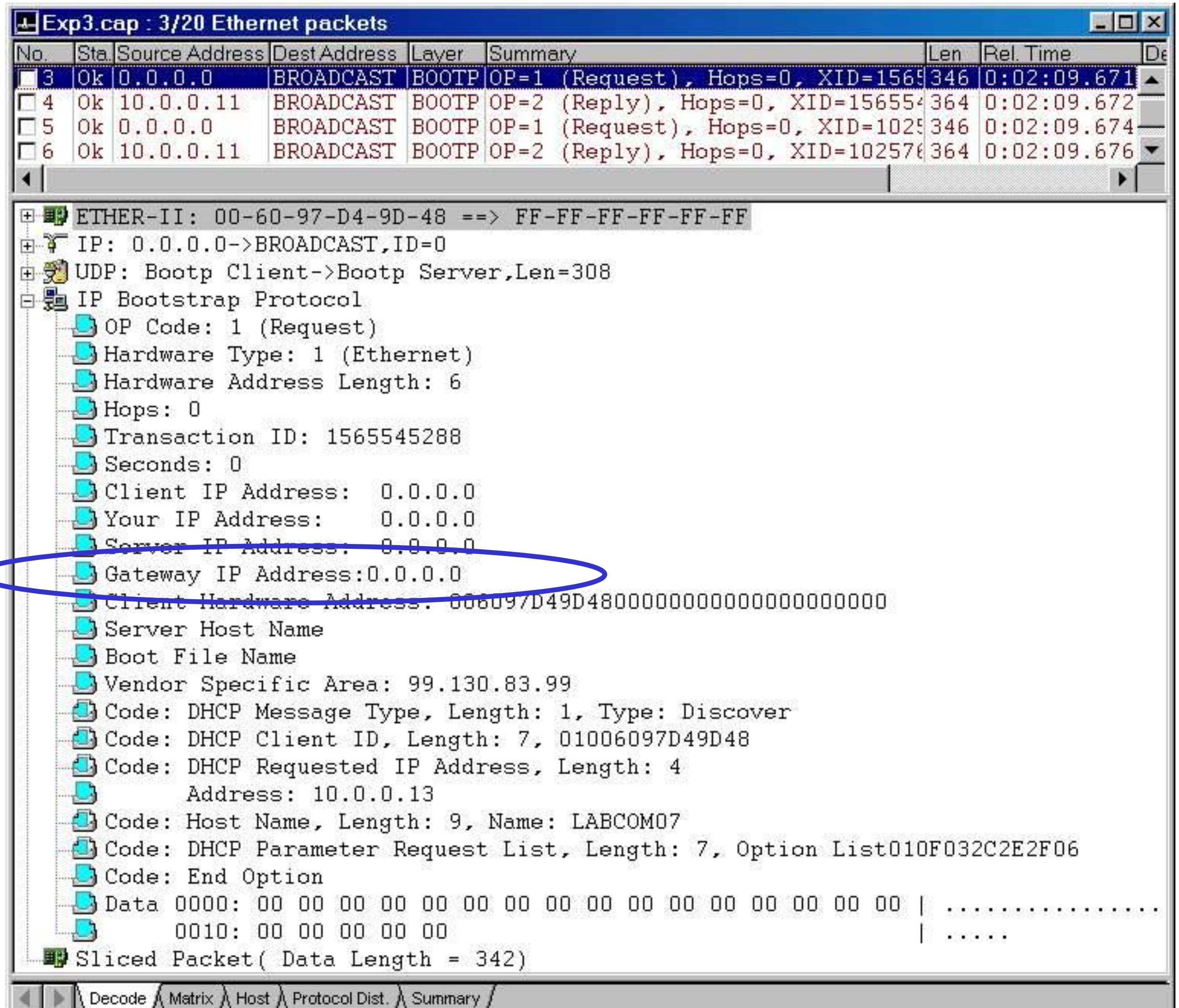
- *DHCP Decline*:
  - The client rejects the offer of a server and restarts the acquisition of an IP address
- *DHCP Nack*:
  - The server informs that it cannot renew the lease of an IP address
- *DHCP Release*:
  - The client informs the server that it is no longer interested on an IP address
- *DHCP Inform*:
  - The client requests additional information (in this case, the client has an IP address but requires, for example, the DNS server address)

Client and server in different subnets

- It is necessary to make routers behave as *BootP Relay Agents*
- *Routers* reroute *BootP Request* messages to the IP address of the DHCP server inserting the IP address of the receiver interface in the *Gateway IP address* field of the DHCP messages
- DHCP server sends *BootP Reply* messages to the *Gateway IP address*



Gateway IP address field



Bibliography to study

- J. Kurose, K. Ross, "Computer Networking: A Top-Down Approach", Addison-Wesley, 4th Edition
  - Section 4.4.2 "IPv4 Addressing"

IPv6

IPv6 Features

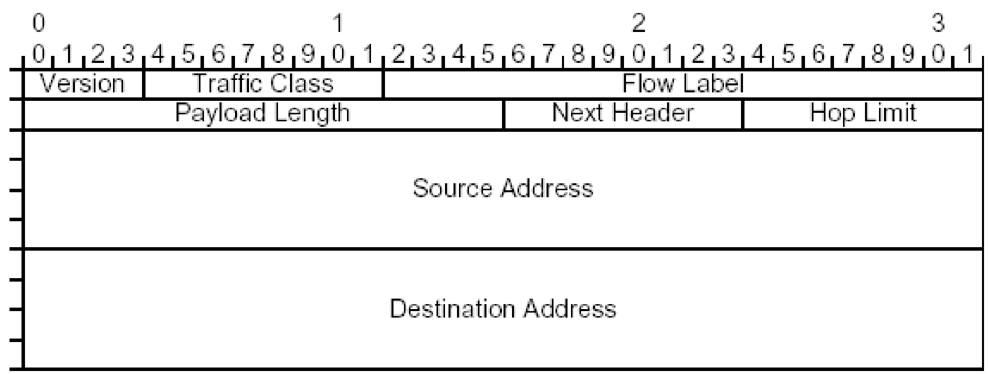
- Larger address space enabling:
  - Global reachability, flexibility, aggregation, multihoming, autoconfiguration, “plug and play” and renumbering
- Simpler header enabling:
- Routing efficiency, performance and forwarding rate scalability
- Improved option support

IPv6 Addressing

- IPv4: 4bytes/32 bits
  - ~ 4,294,967,296 possible addresses
- IPv6: 16bytes/128 bits
  - 340,282,366,920,938,463,463,374,607,431,768,211,456 possible addresses
- Representation
  - 16-bit hexadecimal numbers
  - Hex numbers are not case sensitive
  - Numbers are separated by (:)
  - Abbreviations are possible
    - Leading zeros in contiguous block could be represented by (::)
    - Example:
    - 2001:0db8:0000:130F:0000:0000:087C:140B = 2001:0db8:0:130F::87C:140B
    - Double colon only appears once in the address
  - Address's prefix is represented as: prefix/mask_number_of_bits

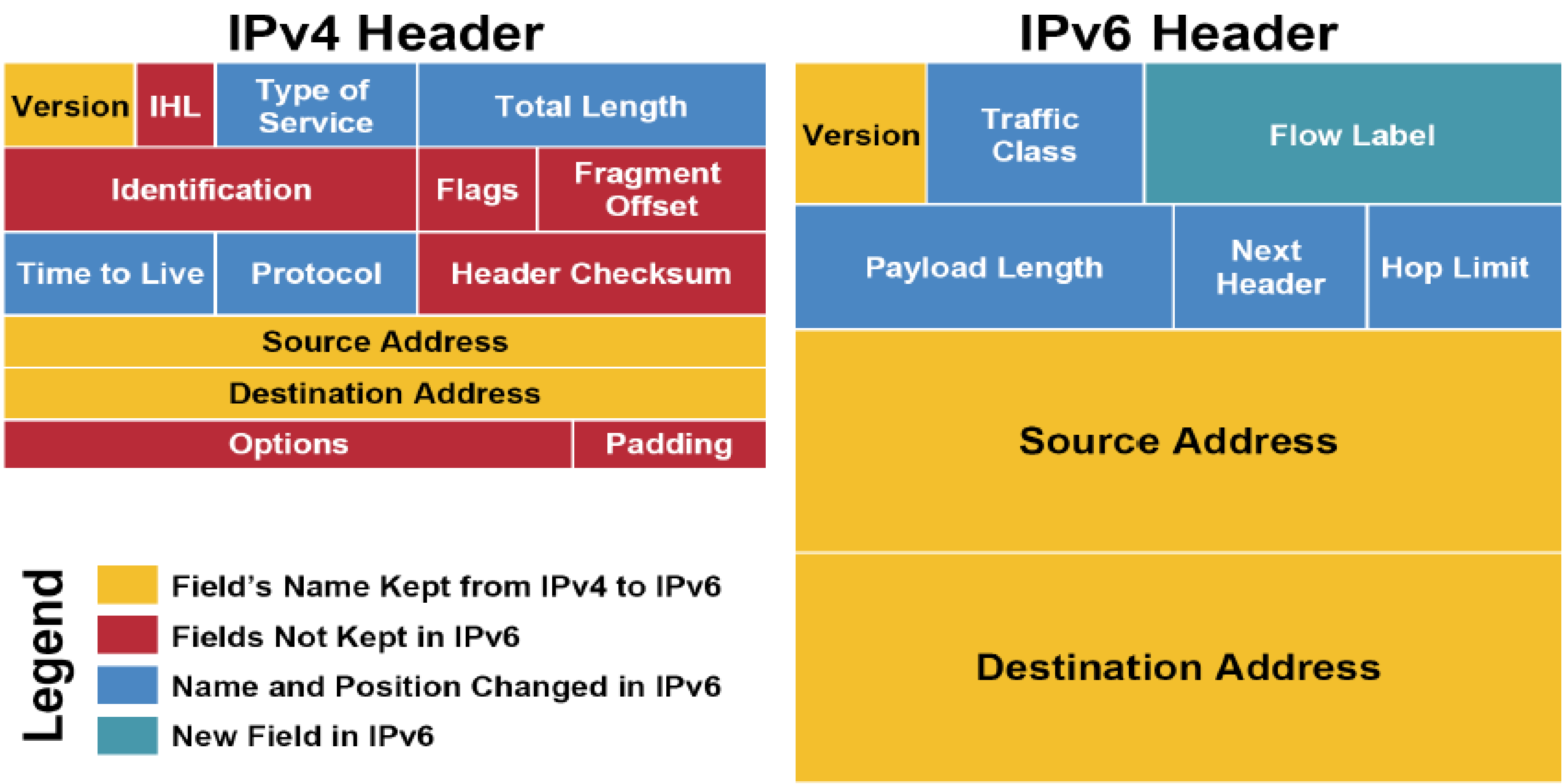
Header

- Fixed-length
  40 byte header
- Checksum removed
- No options in base header



- New flow label field (use is currently not defined, actually several uses exist for each purpose)
- Faster packet processing (with hardware support)
- Options in flexible and extensible extension headers (can be transparent for transit nodes)

IPv6 Header compared to IPv4 Header



Extension Headers

- Hop-by-hop (various, e.g. discard packet w/o ICMP)
- Routing (address list → source routing)
- Fragment (fragmentation only done by source)
- Destination options (various , e.g. discard packet w/o ICMP)
- Authentication (integrity and data origin auth.)
- Encapsulating security payload (confidentiality)

IPv6 address format

2001:0DA8:E800:0000:0260:3EFF:FE47:0001

- 8 groups of 4 hexadecimal digits
  - Each group represents 16 bits
  - Separator is ":"
  - Case-independent
- Addresses in IPv6 very complex
  - Auto configured
  - Local addresses

IPv6 Addressing

Type	Binary	Hexadecimal
Global Unicast Address	0010	2
Link-Local Unicast Address	1111 1110 10	FE80::/10
Unique-Local Unicast Address	1111 1100 1111 1101	FC00::/8 FD00::/8
Multicast Address	1111 1111	FF00::/16

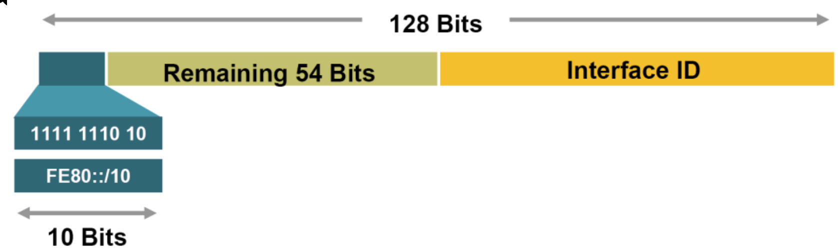
Examples

- 2200:A:A::1/64
- 2001:db8:a0b:12f0::1/64
- 2731:54:65fe:2::a7/64
- fe80::19b3:fddb:75f9:740/64

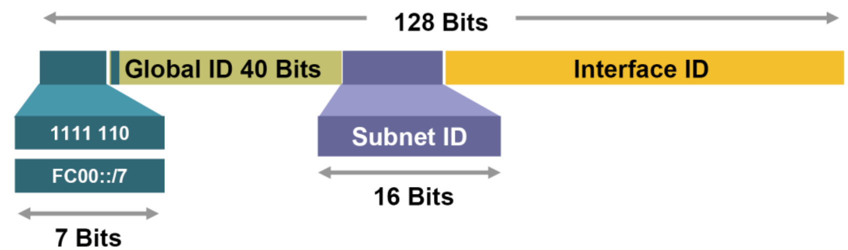
- My PC
  - IPv6 address: 2001:0:9d38:90d7:30a6:16a4:3f57:e09e
  - Link local IPv6 address:fe80::30a6:16a4:3f57:e09e

IPv6 Adressing Scheme

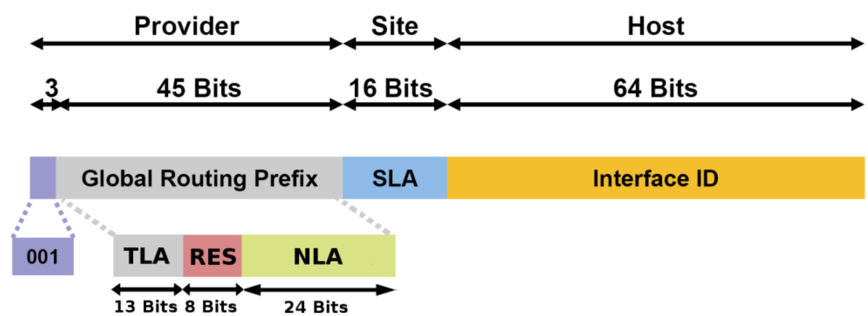
- Interface have multiple addresses
- Addresses have scope:
  - Link Local
    - Valid within the same LAN or link



  - Unique Local
    - Valid within the same private domain
    - Can not be used in Internet



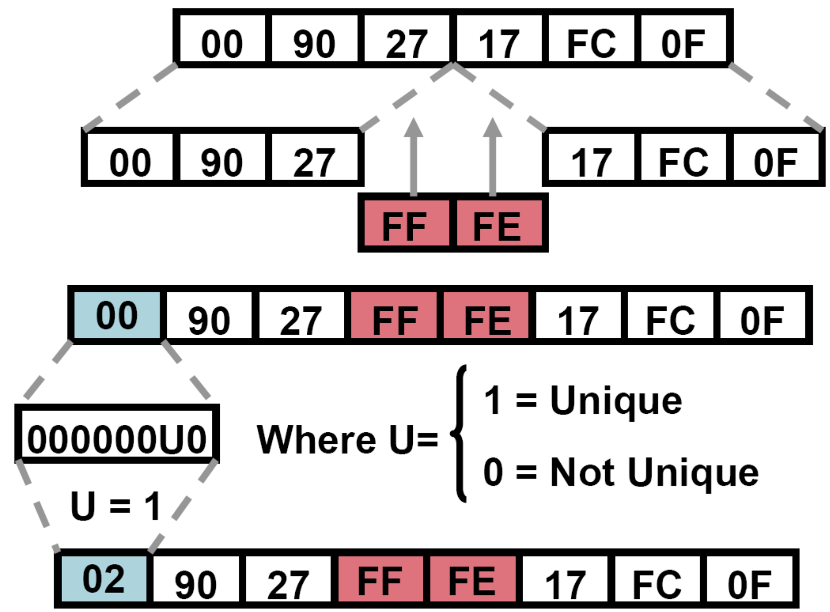
  - Global



- Addresses have lifetime
  - Valid and preferred lifetime

IPv6 Interface Identifier

- Lowest-Order 64-Bit field of any address:
  - Auto-configured from a 64-bit EUI-64, or expanded from a 48-bit MAC address (e.g. Ethernet address)
  - Auto-generated pseudo-random number
  - Assigned via DHCP
  - Manually configured



Auto-configuration

- Important concept in the IPv6 birth
  - Terminal needs to automatically obtain its configuration information;
  - Network configuration elements can be changed and automatically propagated to all terminals.
- Auto-configuration methods
  - Stateless: configuration determined by the network;
  - Stateful: configuration determined by the network management.
- Process:
  - Link-local is created;
  - Verify the uniqueness of the link-local address (DAD-duplicate address detection)
  - Selection of the configuration method;
  - Determine the information to be auto-configured (addresses, gateways, …)

Stateless auto-configuration

- Terminal generates its own address. It combines
  - Local information (e.g. MAC address);
  - Information is advertised by routers (prefix defines the local sub-network).
- Advantages:
  - No manual configuration of the terminals;
  - Minimal configuration in the routers;
  - No additional servers.
- If there are no routers, terminal creates its link-local address
  - This address is sufficient to allow a communication in the same local network segment.

Stateless auto-configuration

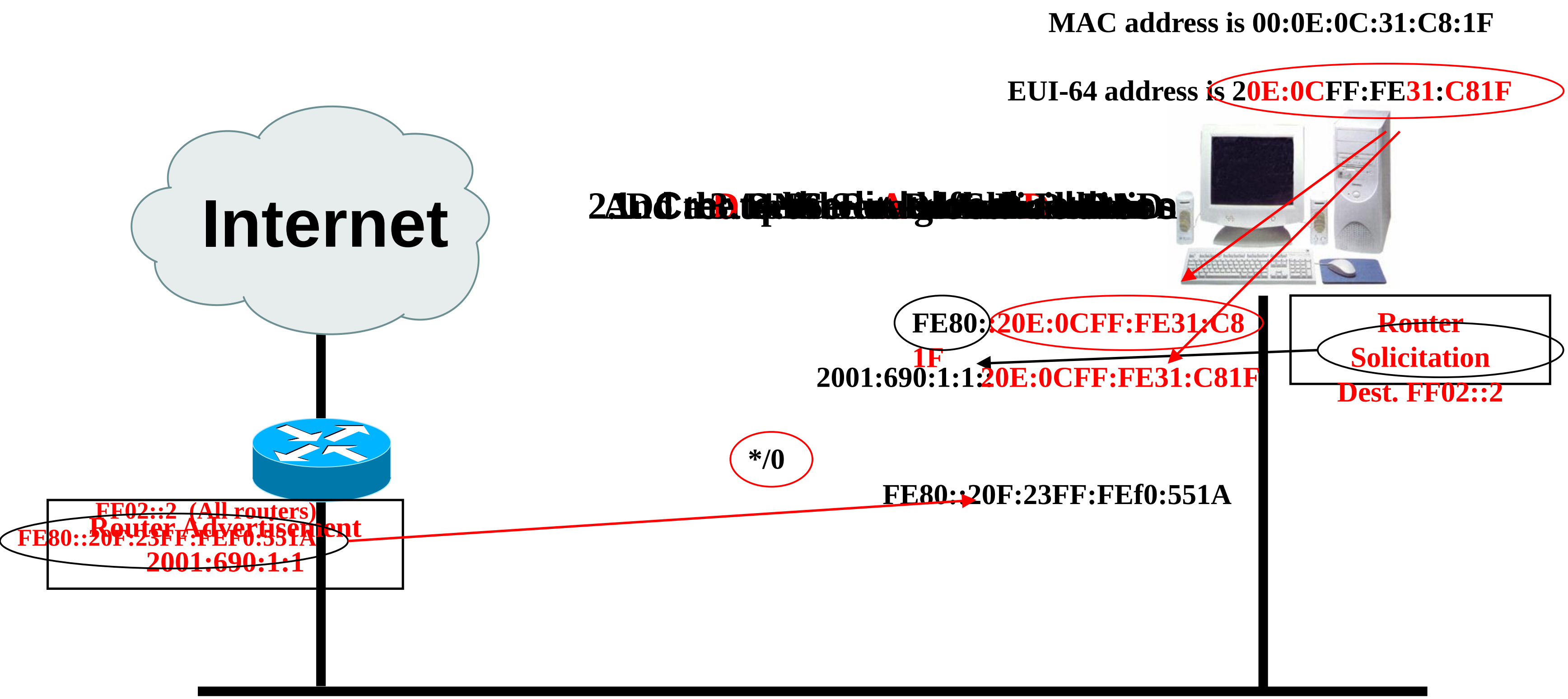
Assign address to Interface.

Node joins the All Routers Multicast group. FF02::1

Sends out a router Solicitation message to That group. ICMP type 133

Router responds with a Router advertisement. ICMP type 134

Stateless auto-configuration



Statefull auto-configuration

- Configuration based on a client-server model
  - Use the Dynamic Host Control Protocol (DHCPv6).
- DHCPv6 allows:
  - Allocation of IPv6 addresses to the terminals;
  - Delivery of specific configuration information of each terminal.
- Guarantees larger configuration control (no DADs required);
- Supports IPv6 concepts of automatic configuration
  - E.g. automatic change of addresses.