www.kom.tu-darmstadt.de
www.httc.de

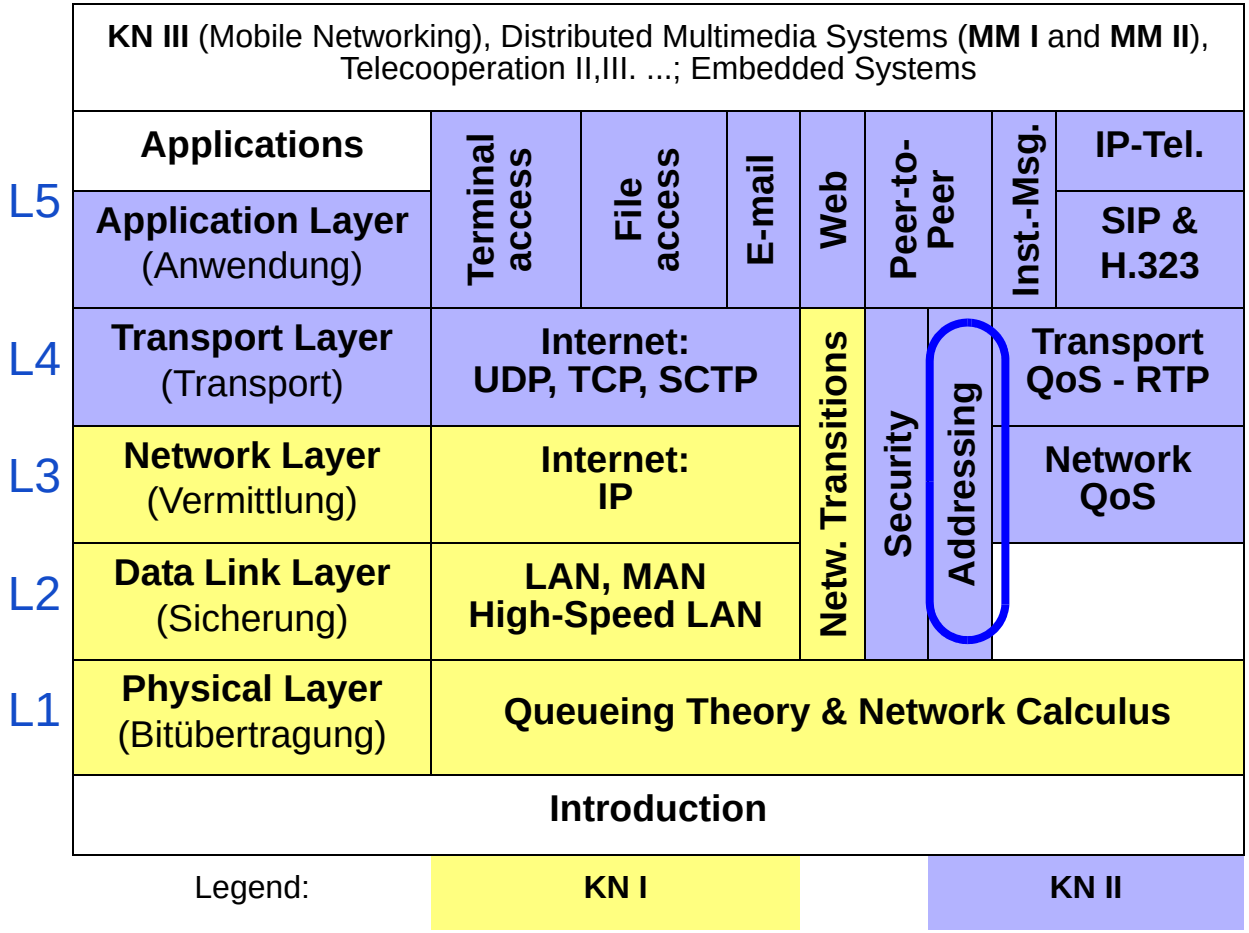
Communication Networks II
Addressing - Protocols

Prof. Dr.-Ing. **Ralf Steinmetz**

TU Darmstadt - Technische Universität Darmstadt,
Dept. of Electrical Engineering and Information Technology, Dept. of Computer Science
KOM - Multimedia Communications Lab
Merckstr. 25, D-64283 Darmstadt, Germany, Ralf.Steinmetz@KOM.tu-darmstadt.de
Tel.+49 6151 166151, Fax. +49 6151 166152

httc - Hessian Telemedia Technology Competence-Center e.V
Merckstr. 25, D-64283 Darmstadt, Ralf.Steinmetz@httc.de

Scope


KN III (Mobile Networking), Distributed Multimedia Systems (MM I and MM II), Telecooperation II,III. ...; Embedded Systems
L5
Applications
Application Layer (Anwendung)
Terminal access
File access
E-mail
Web
Peer-to-Peer
Inst.-Msg.
IP-Tel.
SIP & H.323
L4
Transport Layer (Transport)
Internet: UDP, TCP, SCTP
Netw. Transitions
Security
Addressing
Transport QoS - RTP
L3
Network Layer (Vermittlung)
Internet: IP
Network QoS
L2
Data Link Layer (Sicherung)
LAN, MAN High-Speed LAN
L1
Physical Layer (Bitübertragung)
Queueing Theory & Network Calculus
Introduction
Legend:
KN I
KN II

Overview

1. Addressing in General
2. Domain Name Service (DNS)
3. Ports - Addressing Concept
4. Dynamic Host Configuration Protocol (DHCP)
5. Address Resolution Protocol (ARP)

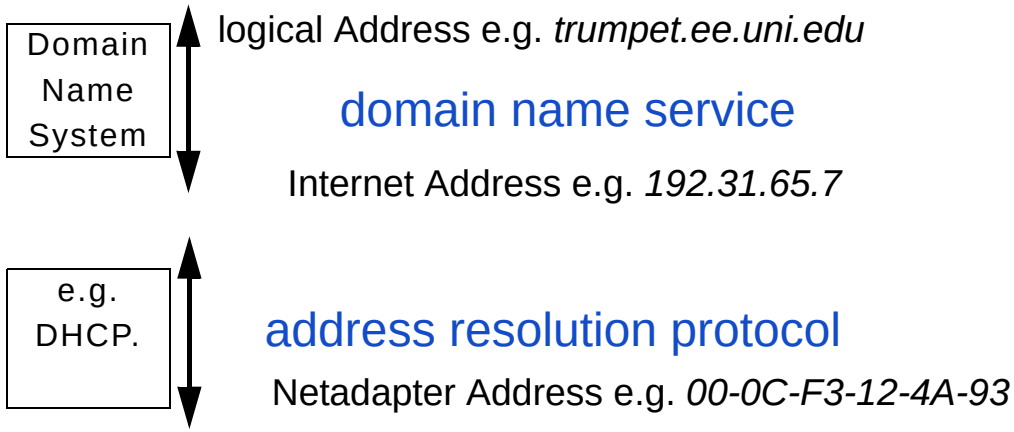
1. Addressing in General

Addressing means 3 types of identifiers: names, addresses and routes

"The **NAME** of a resource indicates WHAT we seek,
an **ADDRESS** indicates WHERE it is, and
a **ROUTE** tells HOW TO GET THERE [Shoch 78]

Addressing must occur at many levels of abstraction,

⇒ **e.g.**



Addressing in General (2)

address identifies type of or specific

- **application (e.g. ssh client)**
- **user (e.g. in instant messageing system, e.g. in IP-telephony skype)**
- **service (e.g. outlook directory)**
- **network (e.g. subnet)**
- **machine (e.g. IP address, peer in P2P overlay network)**
- **interface (e.g. network address), ...**

involves also (in general)

- **overlay networks**
  - in Peer-to-Peer use of distributed hash tables DHT
- **directory services**
- **OSI, X.25 addr.**
- **IP addresses, incl. IP v.6**
- **network addr.**
- **Mobile IP addr.**

2. Domain Name Service (DNS)

Purpose:

- **Internet Protocol address is a 32-bit integer**
- **People prefer to assign machines pronounceable names (host names)**

  - with “tv” domain of Tuvalu (Islands in South Pacific)
- **hard-coded IP addresses within applications may become outdated**
  - e.g., when moving mailserver / web server to other server with different address

⇒ **mapping from name to IP address needed**

Approaches:

- **use file with mapping on every host ("`hosts`" file), updated regularly**
  - doesn’t scale nowadays (file too large, too many file update operations)
- **use of decentralized hierarchical scheme**

⇒ **DNS**

Domain Name Service (DNS) - Basics

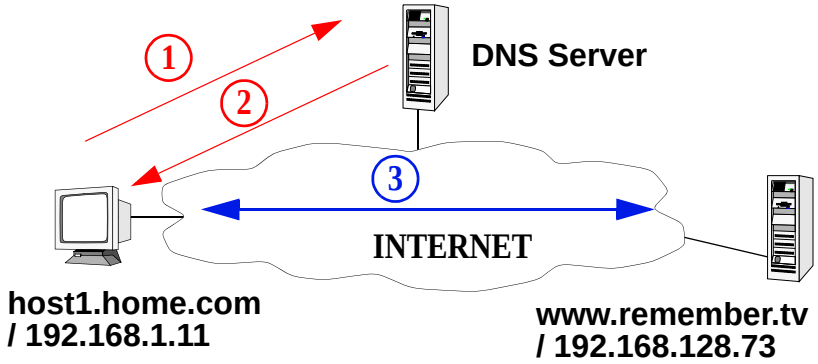
Standards:

- **Basics: RFC 1034, RFC 1035**
  - and lot of documents describing additional features

DNS characteristics:

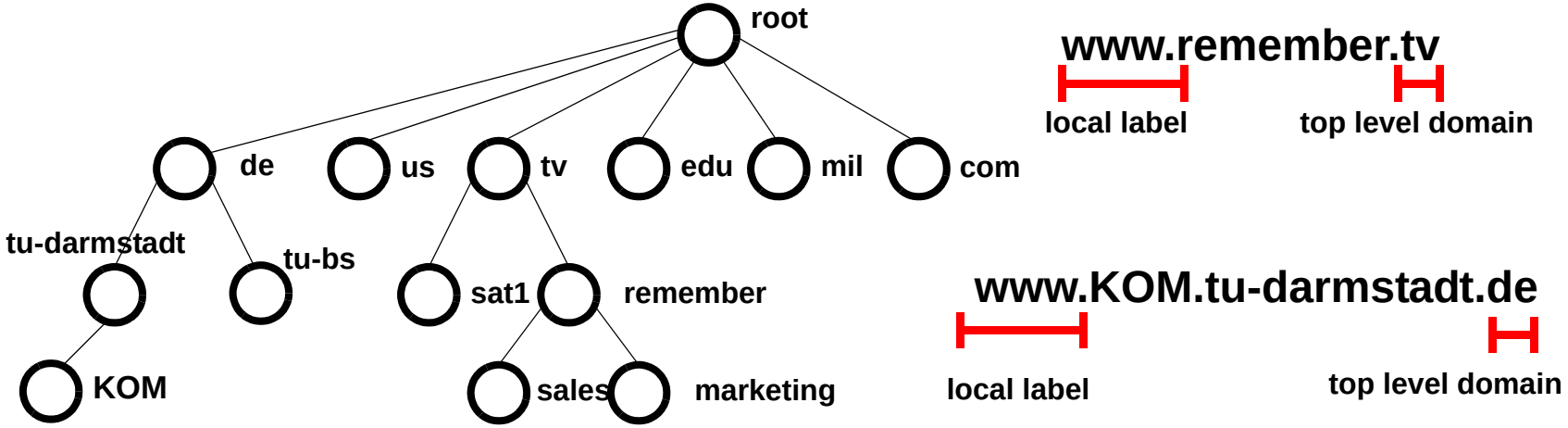
- **distributed (responsibility and physical) database**
- **provides mapping between host names and IP addresses**
- **additional services: e.g. mail routing information**

Operation - basic description (requesting a www site):



1. **Host1 sends a DNS request to its DNS Server and asks for the IP address of www.remember.tv**
2. **The DNS server sends the IP address (192.168.128.73)**
3. **Host1 is now able to communicate with www.remember.tv**
   tv = top-level domain of island Tuvalu

2.1 DNS: Name Space



Top-level domains

- unnamed root
- 1 arpa domain (arpa)
- generic domains: 7*3-char. domains (com, edu, gov, int, mil, net, org)
- country domains: based on (2-char.) country codes (ISO 3166: tv, de, ...)

Registration

- geographical (e.g. remember.tv)
- organizational (e.g. remember.com)

Domains, subdomains, ...

- by local authorities (e.g. admin of remember)
- e.g. sales, marketing, ...

DNS: Name Space (2)

Tree leaves represent domains without further subdomains

- **but with IP equipment (computers, printers, ...)**

Distribution with regard to organizational issues

- **but without regard to physical connections**
- **hierarchy can be distributed at the underlaying network**

Allows multiple naming hierarchies to be embedded

- **specified by object types: e.g. MX: Mail Exchanger, NS: Name Server**

Several domains can be hosted by one server

- **e.g. domains sales.remember.tv, marketing.remember.tv hosted by one server**

'Popular' domains have been used up

- **especially in .com**

New top-level domains have been approved by ICANN recently

- **The Internet Corporation for Assigned Names and Numbers**
- **www.icann.org**

2.2 DNS: Name Server Types

No server has all name-to-IP address mappings

Local name servers:

- **each ISP, company has LOCAL (DEFAULT) NAME SERVER**
- **host DNS query first goes to local name server**

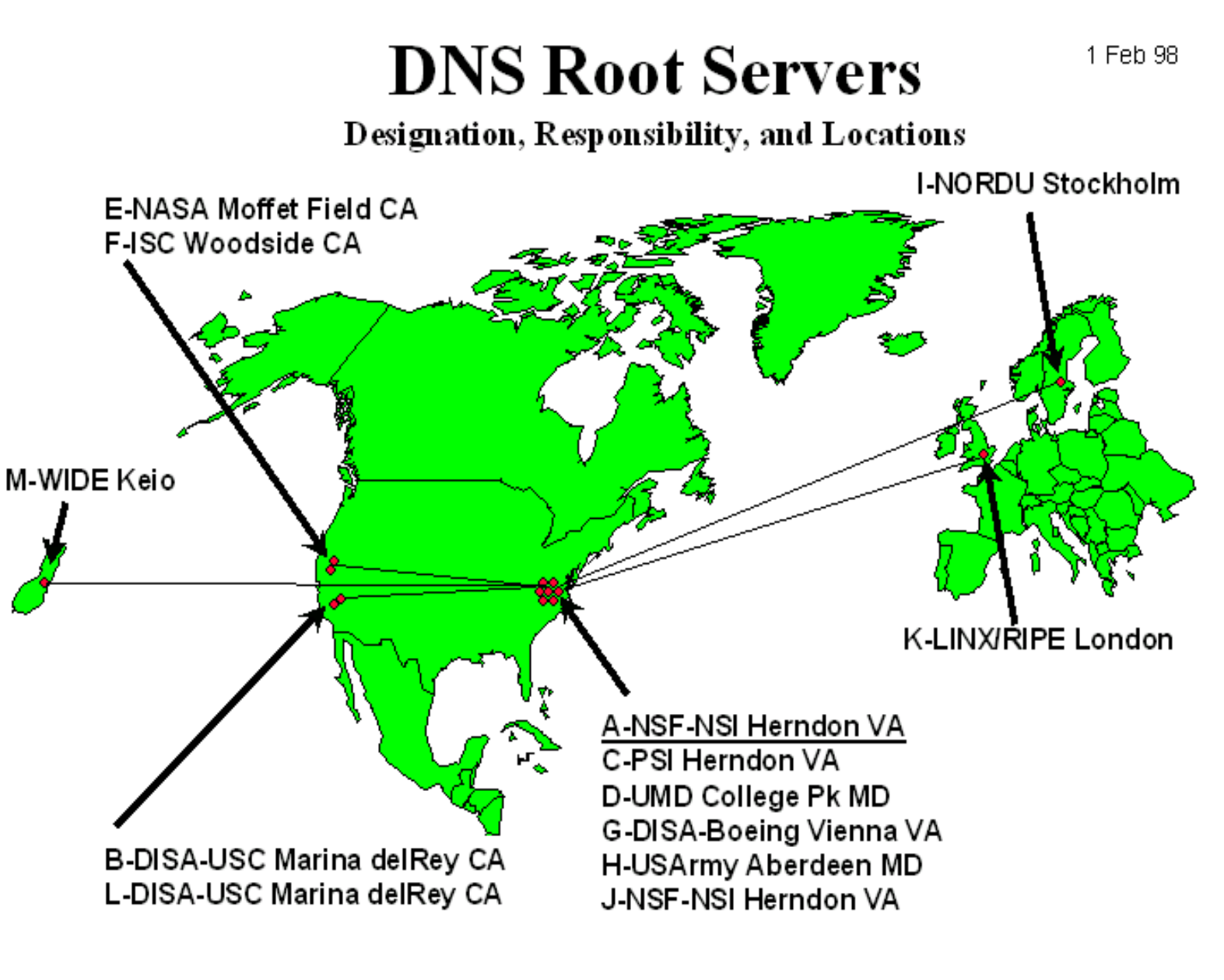
Authoritative name server:

- **for a host: stores that host's IP address, name**
- **can perform name/address translation for that host's name**

Root name server:

- **contacted by local name server that can not resolve name**
- **root name server:**
  - contacts authoritative name server if name mapping not known
  - gets mapping
- **returns mapping to local name server**

DNS: Root Name Servers



DNS: Resource Records

Each domain can have set of RESOURCE RECORDS (RR) associated with it

- **Different types: most common are IP address**

DNS maps domain names onto resource records

Resource record format are five tuples:

Domain_name	Time_to_live	Class	Type	Value
domain to which this record applies	'stability' of the record	IN for Internet information (others possible)	record type (see below)	number, domain, ASCII string depends on type

DNS: Resource Records (2)

Some record types:

Type	Meaning	Value
A	IP address of named host	32 bit integer giving IP address
MX	Mail exchange associated with name	Priority, domain willing to accept email
NS	Name server	Name of server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
...		

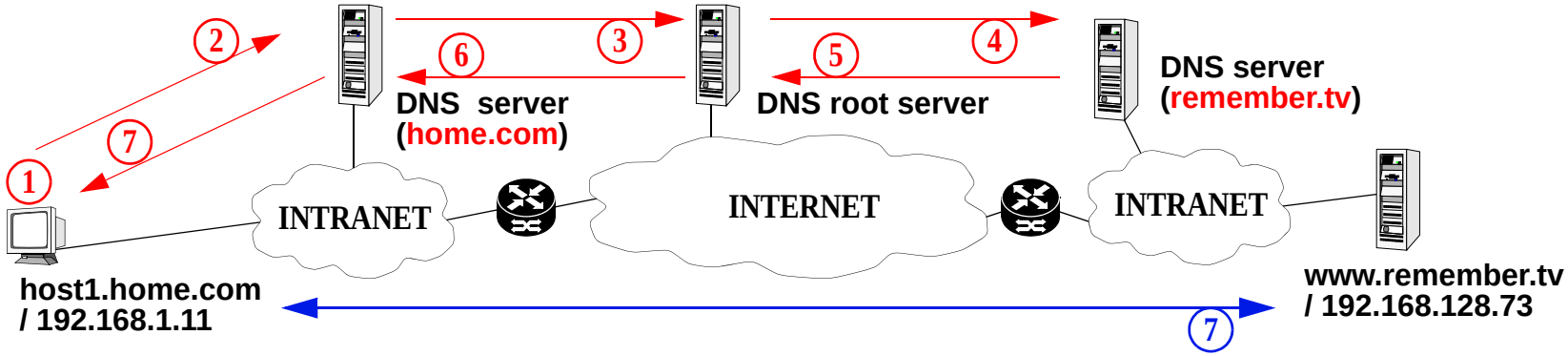
DNS Database

Example:

```
$TTL86400
;
; Start of Authority:
; Nameserver:
@INNS agitator.ibr.cs.tu-bs.de.
@INNS infbssys.ips.cs.tu-bs.de.
@INNS oker.escape.de.
;
; Mail Exchanger fuer ibr.cs.tu-bs.de:
;
@INMX 10 agitator.ibr.cs.tu-bs.de.
cipINMX10 pott.cip.ibr.cs.tu-bs.de.
mail.cipINCNAMEpott.cip.ibr.cs.tu-bs.de.
;
; IPv6:
;
ipv6INNSagitator.ibr.cs.tu-bs.de.
IN      NS      oker.escape.de.
IN      NS      ns.ipv6.tm.uka.de.
asaft INA134.169.34.100
INMX  10 agitator.ibr.cs.tu-bs.de.
osaft INA134.169.34.101
INMX  10 agitator.ibr.cs.tu-bs.de.
salvatorINA134.169.34.17
INMX  10 agitator.ibr.cs.tu-bs.de.
nis INCNAMEsalvator
loghost INCNAMEsalvator
```

2.3 DNS: Protocol

typical operation - extended description



DNS recursive resolution:

- in many steps
  1. local appplication wants to resolve address
  2. Host1 sends a DNS request to its local DNS server and Host1 asks for the IP address of www.remember.tv...
  3. ...
  4. ...
  5. ...
  6. ...
  7. Host1 is now able to communicate with www.remember.tv

DNS: Protocol (2)

1. **Application on Host1**
   - calls local “resolver”, asks for IP addr. of www.remember.tv (name as parameter)
2. **Host1**
   - sends a DNS request (using UDP) to its local DNS server and asks for IP addr.
3. **DNS server can not resolve the request**
   - forwards the request to one of the toplevel root server
   - request marked as “*recursive resolution*”
4. **toplevel DNS server**
   - knows the location of the DNS server(s) responsible for remember.tv
   - request is (also recursive) forwarded to this DNS server
5. **DNS server**
   - is capable to resolve the request
   - sends the IP address (192.168.128.73) back to the root server
6. **root server**
   - sends the answer to the home.com DNS server

(continued on the next page)

DNS: Protocol (3)

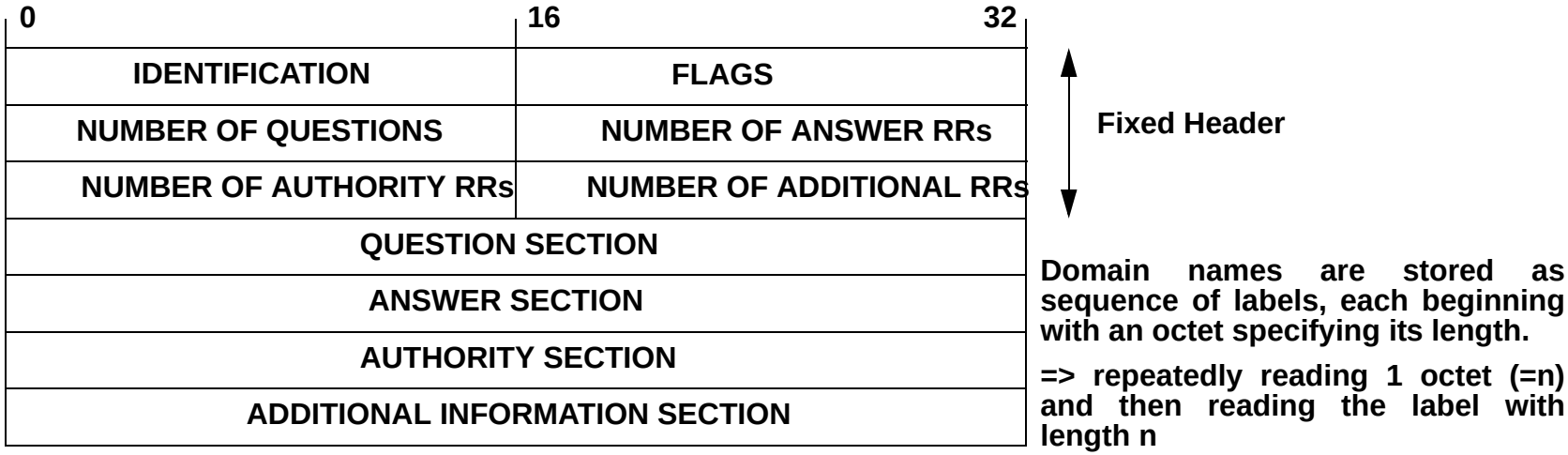
7. **home.com DNS server**
   - sends the answer to host1
8. **Host1 is now able to communicate with www.remember.tv**

⇒ **Obviously optimizations are necessary**

⇒ **Efficient Translation, Caching name server**

DNS: Protocol (4)

Domain Server Message Format:



Domain names are stored as sequence of labels, each beginning with an octet specifying its length.

=> repeatedly reading 1 octet (=n) and then reading the label with length n

Client fills in only the QUESTION SECTION, server returns QUESTIONS and ANSWERS in response

The QUESTION SECTION contains the queries consisting of:

- *QUERY DOMAIN NAME*: the name (stored as labels)
- *TYPE*: e.g. *MX* Mail Exchanger, *NS* Name Server
- *QUERY CLASS*: different classes, e.g. official Internet names

The ANSWER SECTION contains the answers consisting of:

- *TIME TO LIVE*: specifies how long the information can be cached
- *RESOURCE DATA*: each record describes one domain name and mapping
- Other fields: *TYPE, CLASS, RESOURCE DATA LENGTH, RESOURCE*

2.4 DNS - Summary

Transport protocol:

- **normally UDP (PORT = 53) due to efficiency**
- **TCP also possible (due to security reasons)**

Host names <-> IP addresses

- **1 host : 1 IP addr**
- **n hosts : 1 IP addr**
  - many host names may correspond to a single IP address
  - allows a single machine to serve many web sites.
- **1 host : n IP addr**
  - a single host name may correspond to many IP addresses
  - facilitates fault tolerance and load distribution
  - allows a site to move physical location without being noticed at DNS level

Web site //remote.12dt.com/rns/ allows for reverse look-up

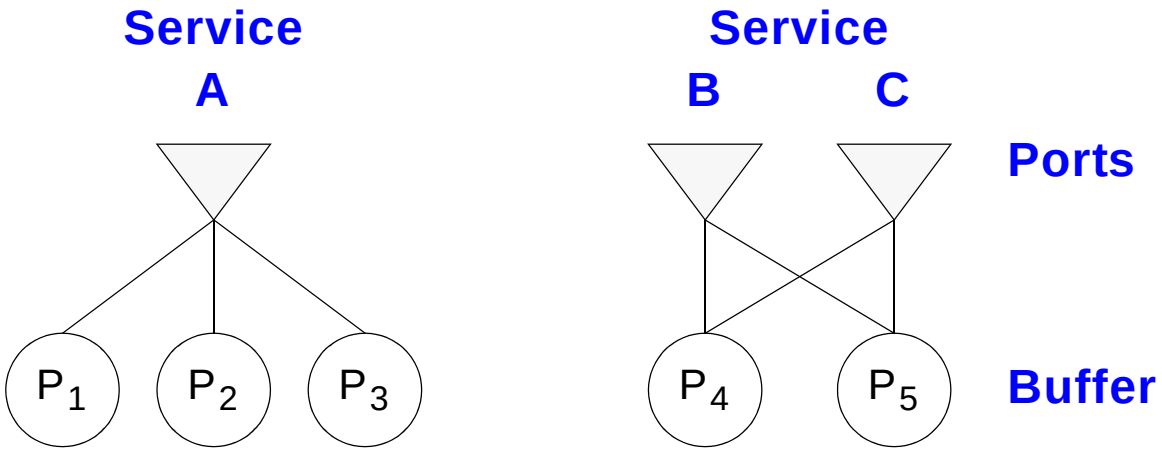
- **created by Frank Riherd**
- **return the name of the resolved IP address**

DNS

Problems:

- **e.g. coexistence with DHCP**
  - dynamic assignment of IP addresses
  - it is possible to integrate DHCP and DNS mechanisms
  - e.g.
    - Microsoft DNS Server, bind?
- **e.g. security**
  - DNS system is often used for attacks
  - e.g.
    - an attacker modifies the mapping by spoofing (vortäuschen) packets
    - Countermeasures:
      TCP transport protocol
      DNS authentication mechanisms

3. Ports - Addressing Concept



Addressing of an application of transport service

⇒ **Concept of an abstract communication finishing point: Port**

Service is

- **allocated to exactly one port**

Port access

- **both asynchronous and synchronous access possible**

Port is

- **associated with buffer**

Reserved Port Numbers

Decimal	Keyword	UNIX Keyword	Description
0			Reserved
1	TCPMUX		TCP Multiplex
5	RJE		Remote Job Entry
7	ECHO	echo	Echo
9	DISCARD	discard	Discard
11	USERS	systat	Active Users
13	DAYTIME	daytime	Daytime
17	QUOTE	qotd	Quote of the Day
19	CHARGEN	chragen	Character Generator
20	**FTP-DATA**	**FTP-DATA**	**File Transfer Protocol (Data)**
21	**FTP**	**FTP**	**File Transfer Protocol**
23	**TELNET**	**TELNET**	**Terminal Conncetions**
25	**SMTP**	**SMTP**	**Simple Mail Transfer Protocol**
37	TIME	time	Time
42	NAMESERVER	name	Host Name Server

- **TCP and UDP have their own assignments; this table shows some examples for TCP**

Reserved Port Numbers (2)

Decimal	Keyword	UNIX Keyword	Description
43	NICNAME	whois	Who is
53	**DOMAIN**	**NAMESERVER**	**DOMAIN NAME SERVER**
79	FINGER	finger	Finger
80	**HTTP**	**HTTP**	**WORLD WIDE WEB**
101	HOSTANME	hostname	NIC Host Name Server
102	ISO-TSAP	iso-tsap	ISO TSAP
103	X400	x400	X.400 Mail Service
104	X400-SND	x400-snd	X.400 Mail Sending
110	**POP3**	**POP3**	**REMOTE EMAIL ACCESS**
111	SUN RPC	sunrpc	SUN Remote Procedure Call
113	AUTH	auth	Authentication Service
117	UUCP-PATH	uucp-path	UUCP Path Services
119	NNTP	nntp	USENET News Transfer Protocol
129	PWDGEN		Password Generator Protocol
139	NETBIOS-SSN		NETBIOS Session Protocol
160-1023	Reserved		

3.1 UDP and TCP Port - Link to Application

Application itself

- **example**
  - decompression of video data
  - read process from database or filesystem
- **implementing the application**
  - process, thread
- **interface to communication systems**
  - buffers with predefined access routines

Sender and receiver create

- **socket or streams**
- **several connections inform about a socket (e.g. TCP socket identification) by means of a unique number containing:**
  - IP address of the endsystem
  - 16-bit port number
    - 0..1024: predefined ports, “well known”
    - additional ones are managed dynamically

Example:

```
192.169.100.17:80 socket with
IP address 192.169.100.17 and port no 80
```

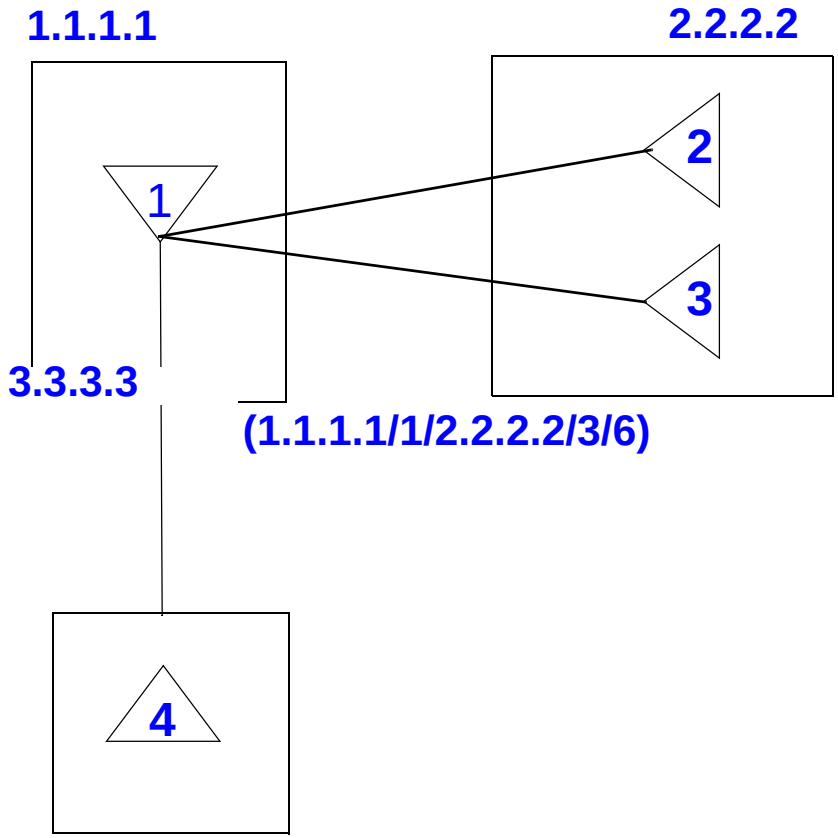
3.2 TCP Connection - Addressing

TCP connection is clearly defined by a quintuple consisting of

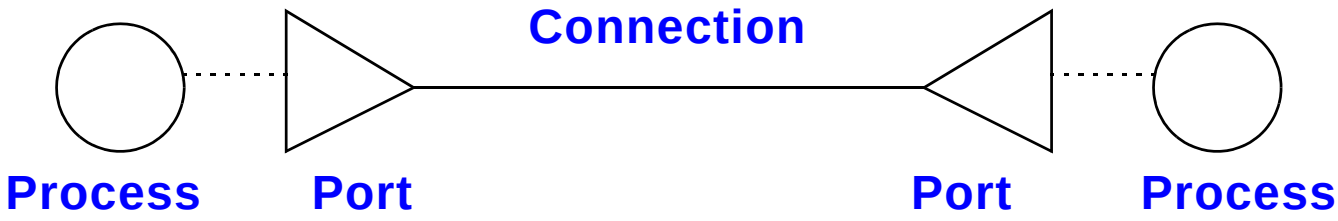
- **IP addresses of**
  - sender and
  - receiver
- **port address of**
  - sender and
  - receiver
- **TCP protocol identifier**

⇒ **Applications can use the same local ports for several connections**

IP addr.sender/port sender/ ..
(1.1.1.1/1/2.2.2.2/2/6)
../IP addr.rec/port rec/TCPid



Addressing



passive open:

- **process indicates that it would accept connect request**

Active open:

- **process requests a connection**

Addressing:

- **port number + protocol identification**
  - clearly identifies entity in the ES
- **IP address**
  - clearly identifies ES

4. Dynamic Host Configuration Protocol (DHCP)

Standards

- **basics: RFC 2131, RFC 1542**
- **BOOTP: RFC 951**

Goals

- **to enable hosts to obtain IP configuration from a server**
- **to reduce administrative work in IP network**
- **to optimize usage of total amount of IP addresses in network**

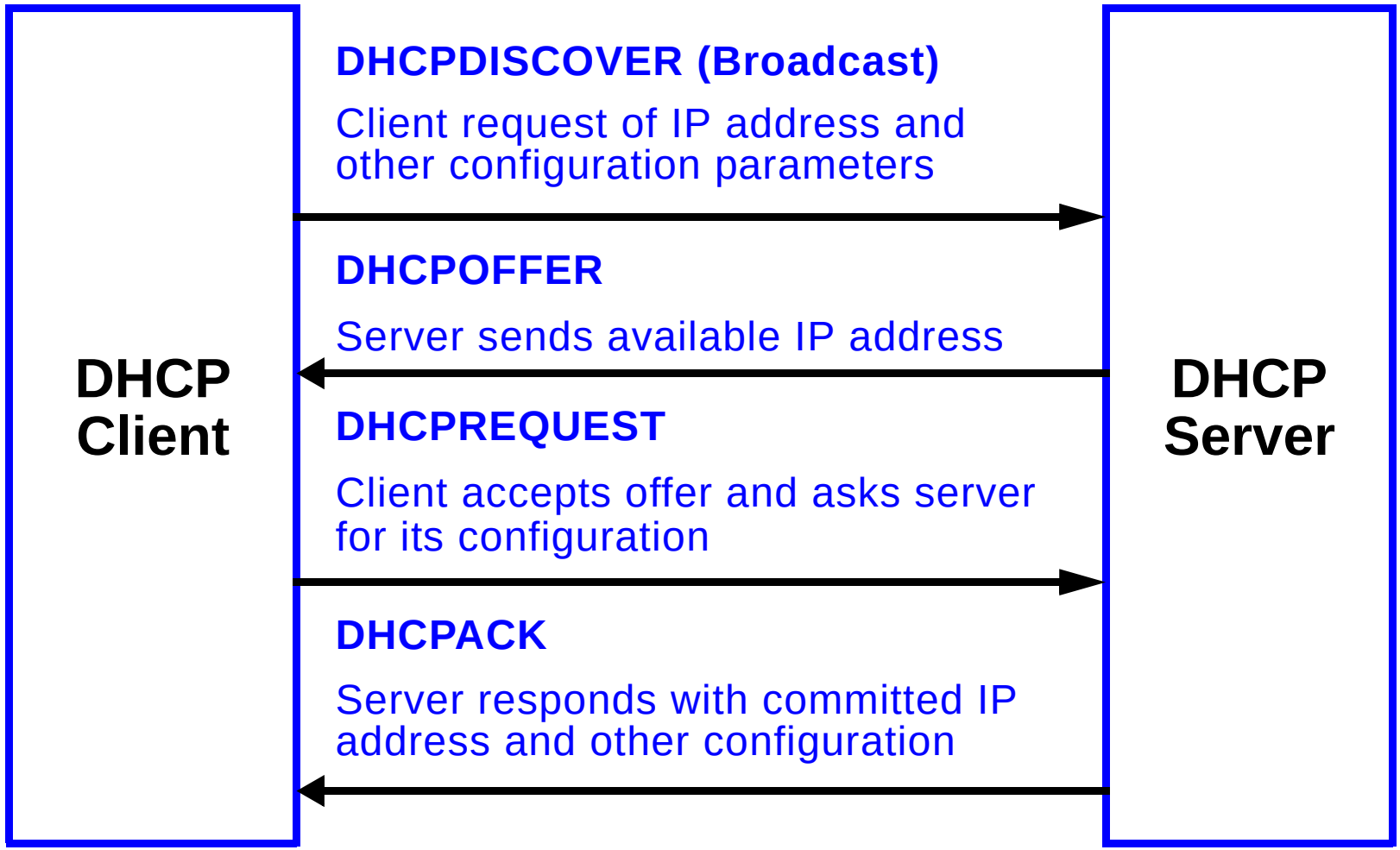
DHCP characteristics

- **extension of BOOTP mechanism**
- **uses UDP as transport protocol**
  - Client: Server: Port 67;
  - Server: Client: Port 68

Mechanisms to assign IP address

- **automatic allocation: permanent IP address is assigned**
- **manual allocation: address assigned by admin, conveyed by DHCP**
- **dynamic allocation; IP address is assigned for a limited period (lease)**

4.1 DHCP Session



4.2 DHCP Configuration

Configuration file:

- **Global parameters**
  - define global parameters like lease time and domain name
  - Parameters are also valid for all subsections
- **e.g.**

```
max-lease-time 3600;
default-lease-time 3600;

option domain-name "acme.com";
option domain-name-servers  a.b.c.d, a.b.c.e;
option netbios-name-servers a.b.c.f, a.b.c.g;
option netbios-node-type 8;
```

DHCP Configuration (2)

Subnets

- if necessary subnets can be specified
- DHCP server can administer large networks
- non-contiguous ranges can be specified
- e.g.

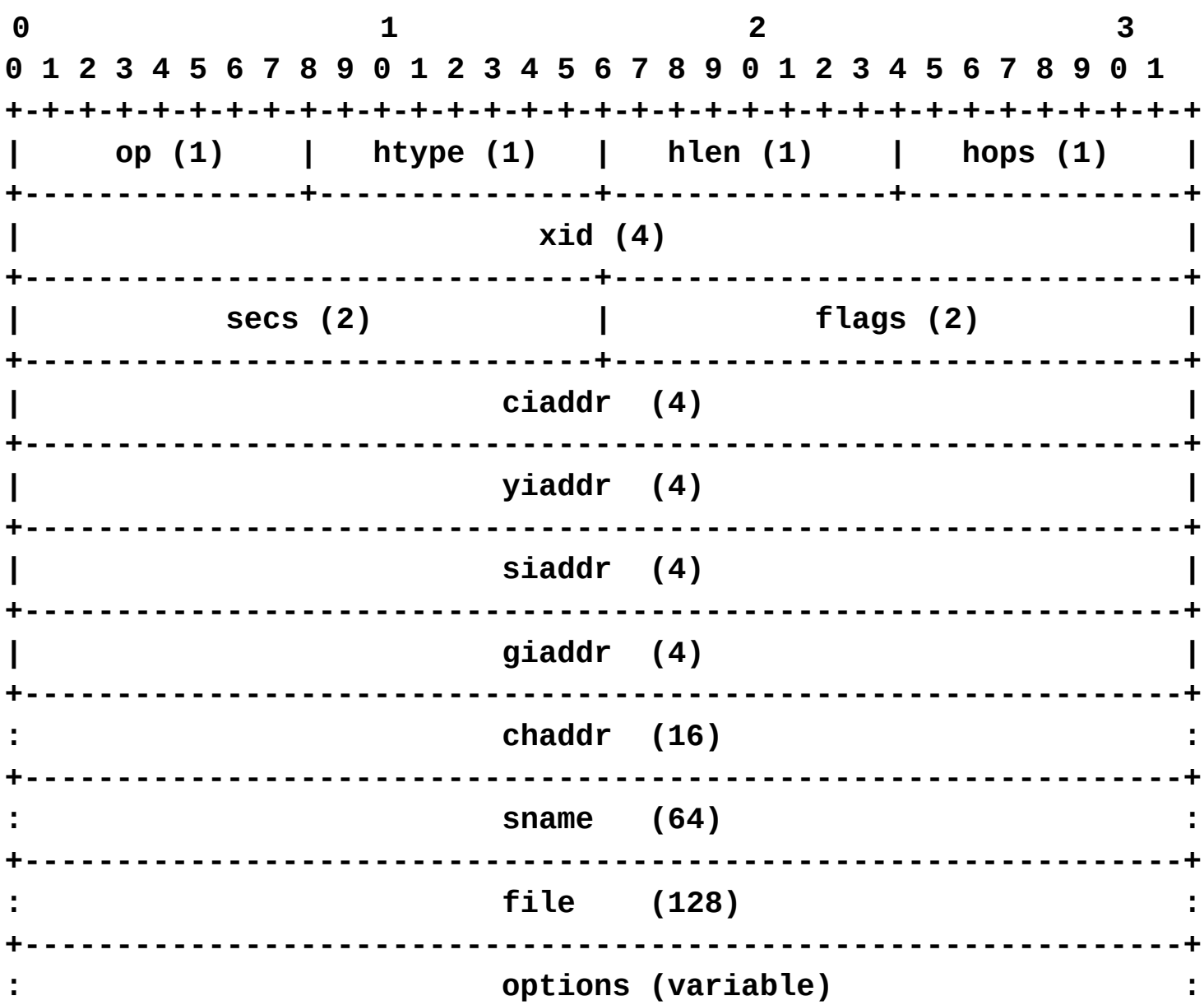
```
subnet a.b.c.192 netmask 255.255.255.224 {
        option routers a.b.c.222;
        range a.b.c.195 a.b.c.199;
        range a.b.c.200 a.b.c.221;
}
```

Fixed addresses for hosts

- used to assign specific address to host (e.g. server)
- e.g.

```
host fileserver {
            hardware ethernet 00:20:48:0c:48:d4;
            fixed-address a.b.c.200;
        }
```

4.3 DHCP - Parameters of the Protocol

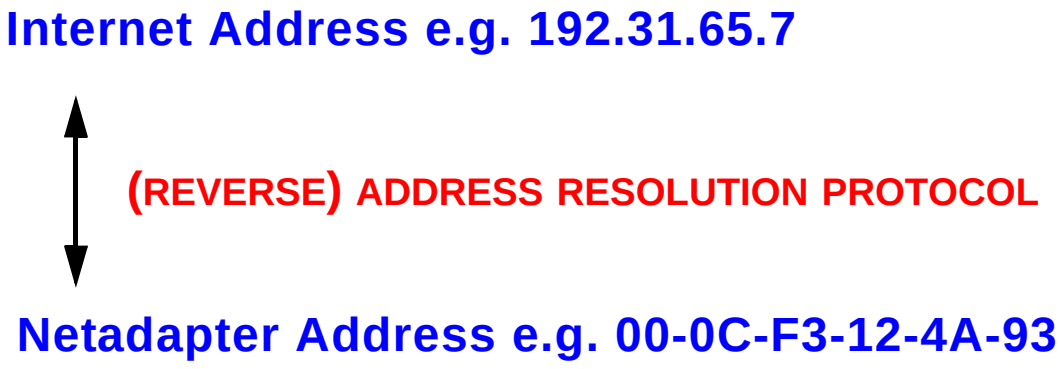


DHCP - Parameters of the Protocol (2)

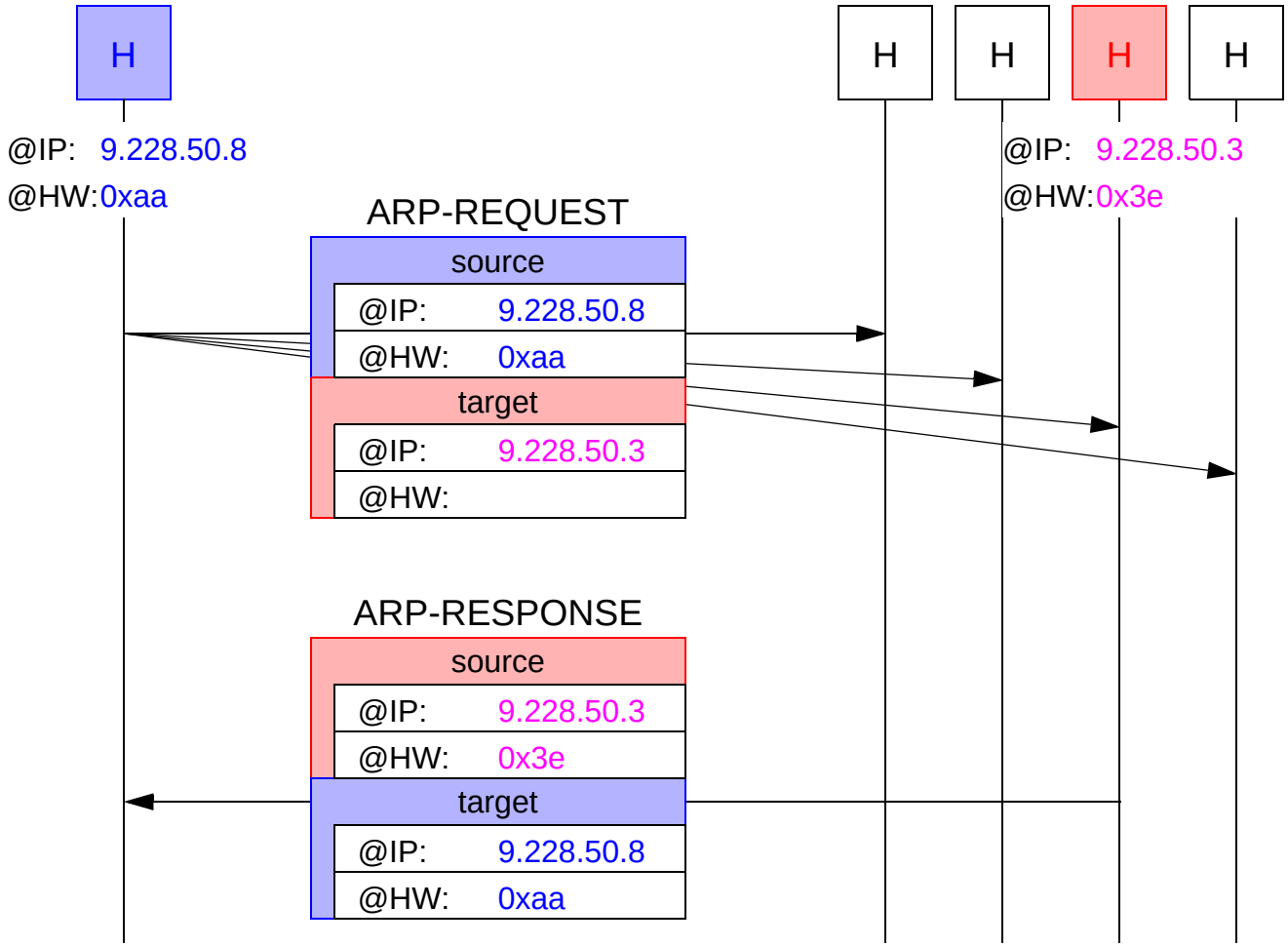
Field	Octets	Description
op	1	Message op code / message type
htype	1	Hardware address type
hlen	1	Hardware address length
hops	1	Client sets to zero, used by relay agents
xid	4	Transaction ID
secs	2	seconds elapsed since client began address acquisition
flags	2	Flags
ciaddr	4	Client IP address
yiaddr	4	'your' (client) IP address
siaddr	4	IP address of next server to use in bootstrap
giaddr	4	Relay agent IP address
chaddr	16	Client hardware address
sname	64	Optional server host name
file	128	Boot file name
option	var	Optional parameters field

5. Address Resolution Protocol (ARP)

Scope



Address Resolution Protocol (ARP) (2)



Address Resolution Protocol (ARP) (3)

Session:

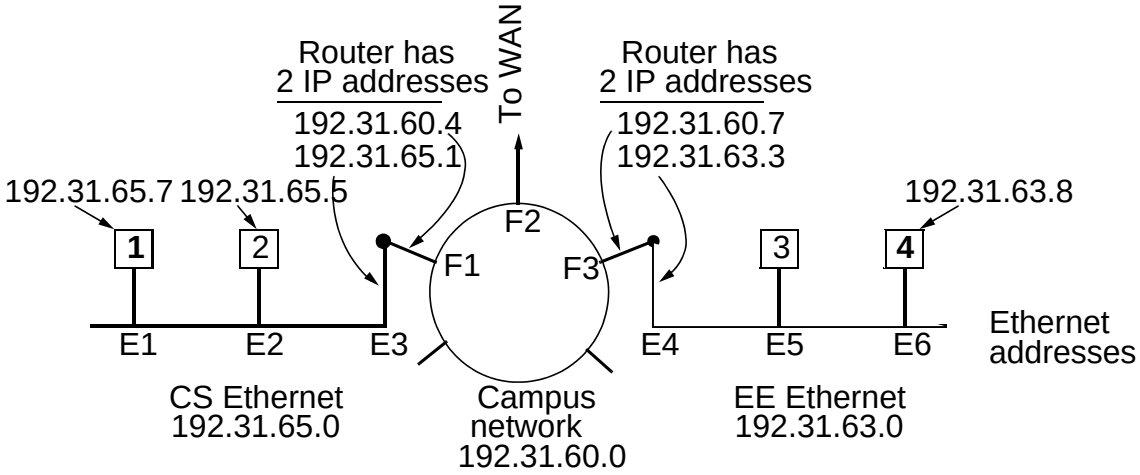
1. **broadcaast of ARP request-datagram**
   - with sender's
     - physical (HW) address
     - IP address
   - with receiver's
     - IP address
2. **response as ARP response-datagram**
   - with receiver's
     - physical address
3. **keep tuple IP-address, physical address in the cache**

Optimization

- **receiver of ARP request has (I,P) tuple in its Cache**
- **at boot time own table is distributed**
  - but, may be too old

Address Resolution Protocol (ARP) (4)

Endsystem can not directly be reached via broadcast



e.g. endsystem E1 to endsystem E 4

- **ARP would not get any answer**
  - Ethernet Broadcast is not transmitted through router

Solution 1: Proxy ARP

- **Local router**
  - knows al remote networks, i.e. respective routers
  - answers local ARP
- **Local E1 sends data for E4 always to local router**
  - local router sends data to remote router (by interpretation IP-address of the packet IP

Solution 2: remote network address well known

- **Local E1 sends data to respective remote router**
- **Local router forwards packets**