Chapter 10

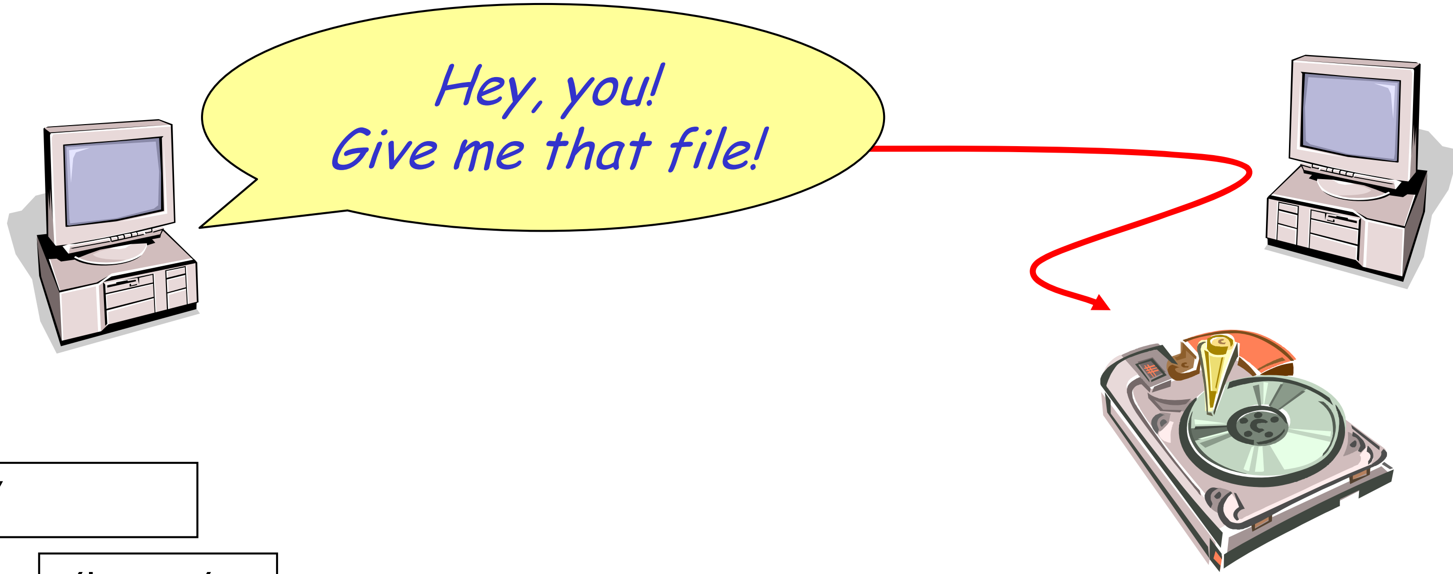
Advanced computer networking

Internet Protocols

Thomas Fuhrmann

Network Architectures
Computer Science Department
Technical University Munich

Addressing and Naming



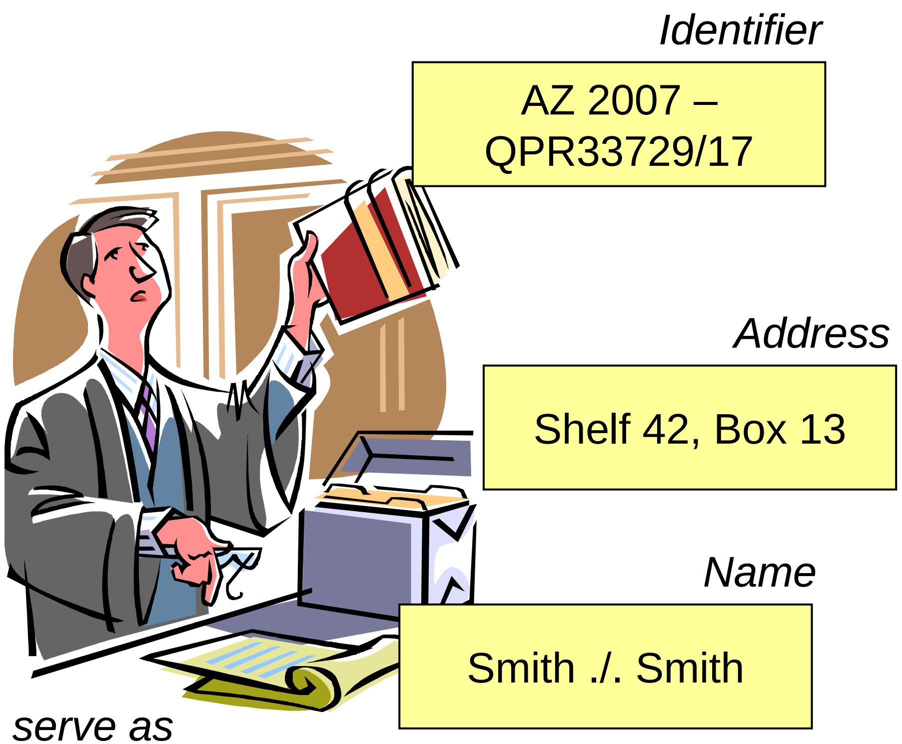

```
/
  /home/
     /home/fuhrmann
     /home/smith
  /usr/
  /var/
```

A file system maps human readable names to sectors on a hard disk.

The domain name system (DNS) maps human readable host names to IP addresses.

Identifier, Address, Name

- Identifier
  - System wide unique
- Address
  - Physical identifier
- Name
  - Logical identifier
  - Human readable



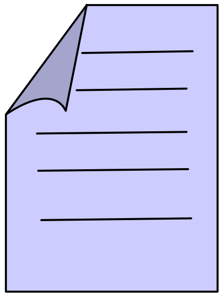
Note that typically both the address and the name can serve as identifier because they are unique!

Neither of these criteria is generally accepted.
There is no exact definition of the terms.

Network Configuration Files

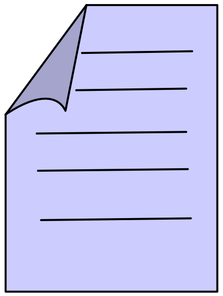
How does a host know the IP addresses of the (other) hosts in the network?

`/etc/hosts`



List of host names with according IP addresses.

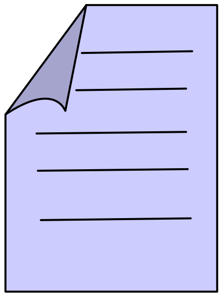
`/etc/sysconfig/…`



IP addresses of local interfaces, default router, …

⇔ ***Boot scripts***

`/etc/resolv.conf`

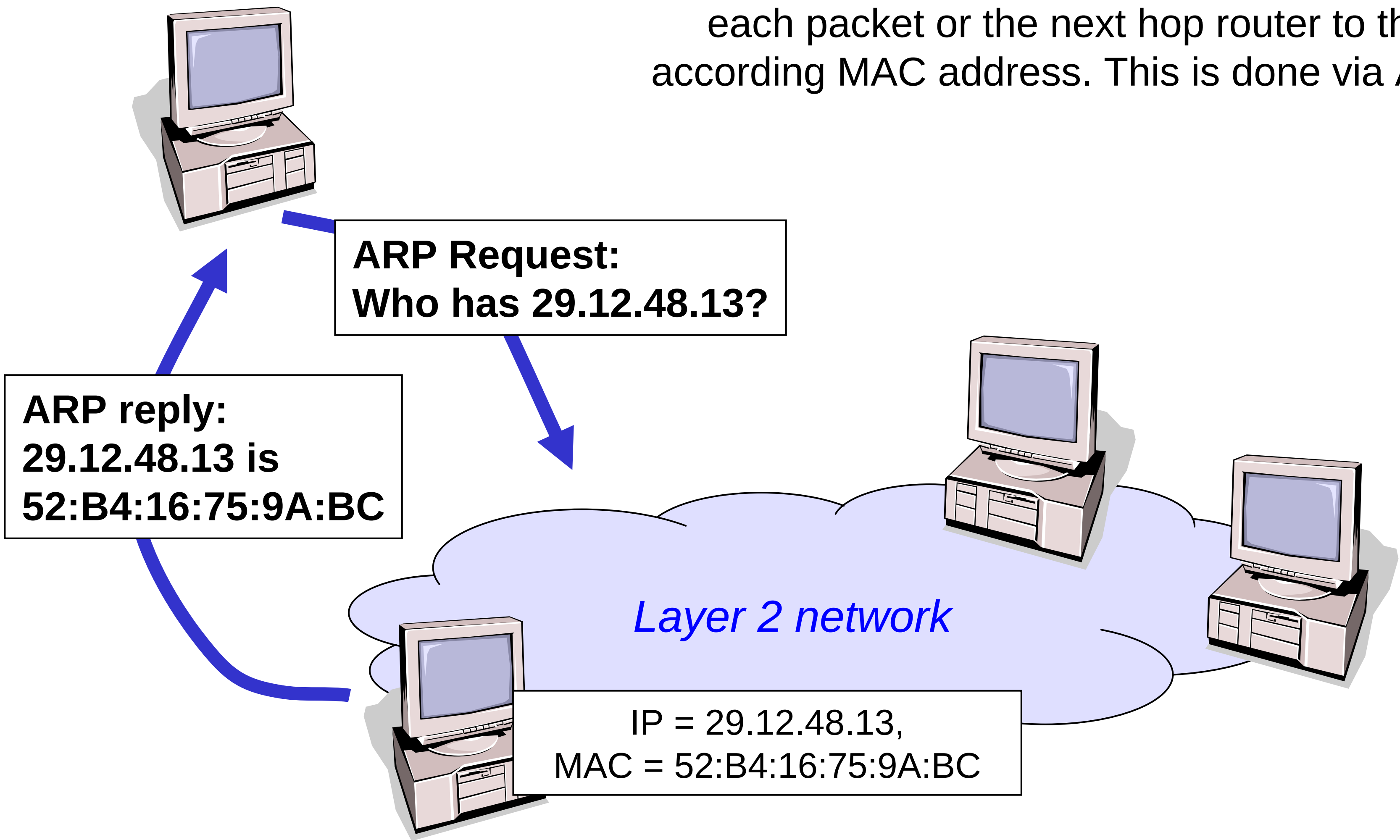


Domain name server

⇔ ***Boot scripts, DHCP***

Recap: Address Resolution

Internet hosts need to resolve the IP address of each packet or the next hop router to the according MAC address. This is done via ARP.

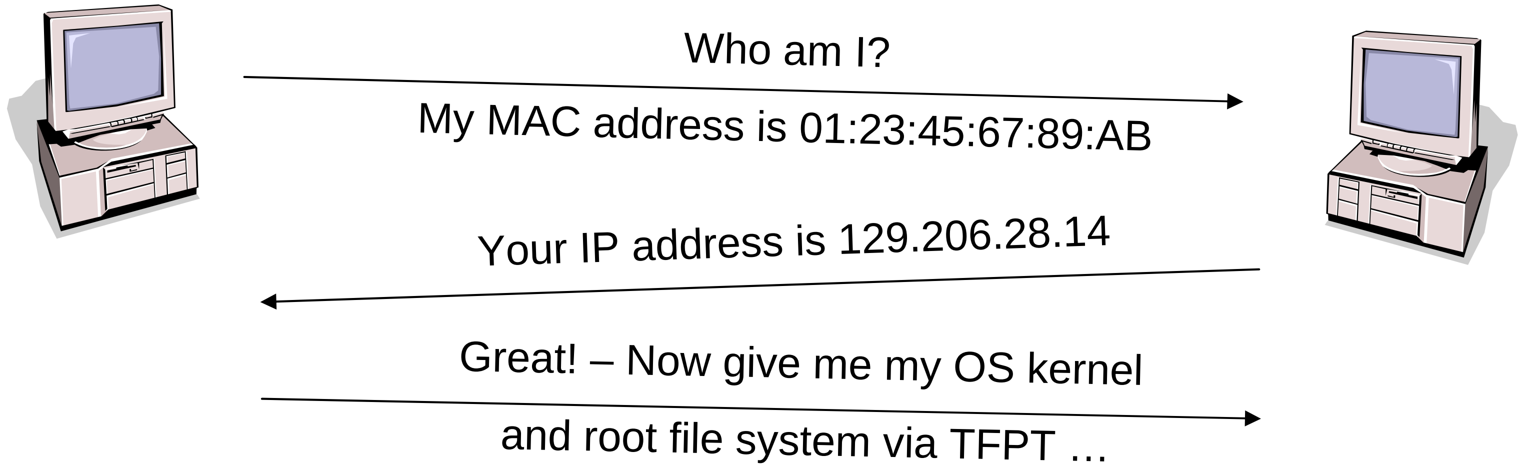


ARP Packet Format

Hardware type e.g. Ethernet = 1		Protocol type, e.g. IPv4 = 2048
Hardware length e.g. Ethernet = 6	Protocol length e.g. IPv4 = 4	Operation 1 = request, 2 = reply
Sender hardware address		
Sender protocol address		
Target hardware address (set to zero in request)		
Target protocol address		

Note: Of course, ARP is carried in a data link layer datagram such as an Ethernet frame!

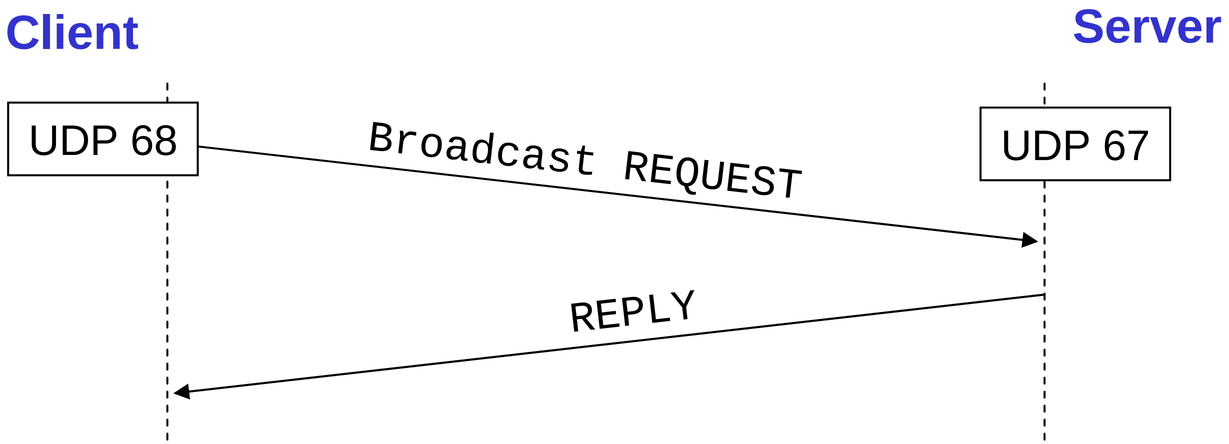
Reverse ARP



- Reverse ARP (=RARP) uses the ARP packet format to do an inverse mapping of hardware address to network layer address.
- RARP was designed to boot diskless computers from the network.
- RARP clients assume that the host of the RARP server that responded to their request also runs a TFTP server.
- After having received an image of the OS kernel and root file system, everything can proceed "as usual".
- Note that still every host needs to be configured, namely by providing a per host file system image on the TFTP server. (The images are named by the hex string of the IP address.)

BOOTP

- BOOTP (=bootstrap protocol, RFC 951) replaces RARP and provides extensive functionality to boot diskless computers.
- It uses UDP/IP packets to carry messages, but hosts are still identified by their MAC address.
- Request uses special case IP address 255.255.255.255 (=limited broadcast).
- Reply normally uses client's IP address, but may be broadcast. Hence the well-known client port.
- Messages are not forwarded by routers, but optional proxies may forward BOOTP messages.
- Note: Request transmission uses random timeout to avoid synchronization.



BOOTP Message Format (1)

Opcode 1 = Request, 2 = Reply	Hardware type 1 = Ethernet	Hardware address length	Hop Count
Transaction ID (→ set by the client and repeated by the server to match replies to requests)			
Number of seconds (… the client has been trying to bootstrap)		unused	
Client IP address (→ set by the client if it knows its IP address, otherwise zero)			
Your IP address (→ set by the server to the client's actual IP address)			
Server IP address (→ set by the server to the server's actual IP address)			
Gateway IP address (→ set by the proxy server, if any)			
Client hardware address (→ set by the client to simplify processing at the server, 16 bytes)			

● ● ● ● ● ● ● ● ●

BOOTP Message Format (2)

• • • • • • • • •

Server host name (→ optional null terminate string; 64 bytes)
Boot file name (→ optional null terminate string; 128 bytes)
Options (64 bytes)

BOOTP – Chicken / Egg Issues

How can the server send an IP datagram to the client, if the client doesn't know its own IP address (yet)? Whenever a bootreply is being sent, the transmitting machine performs the following operations:

1. If the client knows its own IP address ('client IP addr' field is nonzero), then the IP can be sent 'as normal', since the client will respond to ARPs.
2. If the client does not yet know its IP address ('client IP addr' zero), then the client cannot respond to ARPs sent by the transmitter of the bootreply.

There are two options for the solution:

a. If the transmitter has the necessary kernel or driver hooks to 'manually' construct an ARP address cache entry, then it can fill in an entry using the 'client hardware addr' and 'your IP addr' fields.

b. If the transmitter lacks these kernel hooks, it can simply send the bootreply to the IP broadcast address on the appropriate interface. This is only one additional broadcast over the previous case.

Cited from RFC 951

IP Broadcast

- Limited Broadcast
  - IP packets with destination address 255.255.255.255 are delivered to all hosts in the local subnet.
  - Typically, the link layer provides a broadcast mechanism so that IP broadcasts can be handled efficiently.
  - Note: If a host has multiple interfaces, the broadcast is often only sent via the first interface.
- Network broadcast
  - IP packets where the host part of the address is all one bits are sent to the respective network and broadcast there.
  - Example: Packets with destination 141.3.255.255 are routed to the network 141.3.0.0 and broadcast there.
  - Note that this mechanism is obsolete due to security reasons ($\rightarrow$multicast).

Further Notes on Addresses

- The special address 0.0.0.0 denotes the „unknown address“. It is used as source address when a host has not yet obtained a valid address.
- A host part with all zeros denotes the respective network. Example: 141.3.0.0 is a network, 141.3.0.1 is a machine in that network.
- The network 127.0.0.0 is the “loop back” network. All traffic destined to that network is delivered at the local machine. Typically, 127.0.0.1 is used to denote the “local host”.
- The networks 10.0.0.0 and 192.168.x.0 are private networks, i.e. public Internet routers should drop all packets with these addresses.

BOOTP Shortcomings

- BOOTP was designed for relatively static environments where each host has a permanent network connection
  - The site's network administrators create a BOOTP configuration file with the parameters for each host.
  - This is only worth while when these files are typically stable for long periods.
- Dial-up hosts and wireless hosts are much more dynamic.
  - Network administrators want to reserve a pool of IP addresses for these hosts.
  - Upon joining the network, hosts are assigned one of these addresses.
  - After a node leaving the network, its address may be re-used.
- Dynamic address assignment is typical today, but it breaks the original Internet spirit where IP addresses identify hosts.

Dynamic Host Configuration Protocol

DHCP extends BOOTP for more dynamic host configuration:

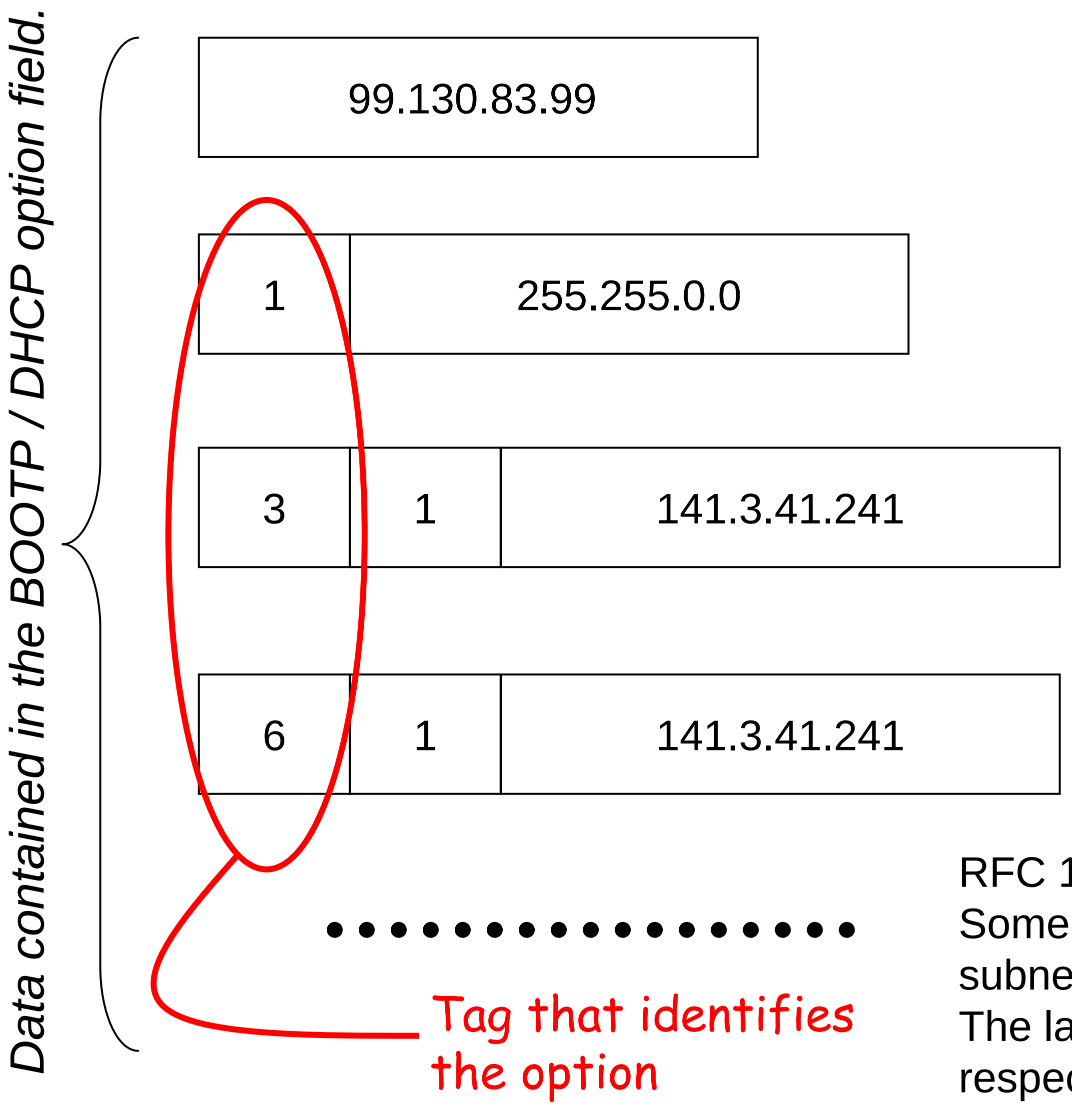
- Manual allocation – The DHCP server hands out IP addresses based on a table of MAC addresses. This table needs to be set up by the network administrator.
- Automatic allocation – The DHCP server automatically chooses a free IP address from the range that it was given by the network administrator. This address then permanently belongs to the respective client.
- Dynamic allocation – Like automatic allocation, but the address is associated with a finite lease time after which the server may assign the address to a different client.

With dynamic allocation addresses expire so that DHCP can deal with ungracefully leaving clients.

- While staying in the network clients need to regularly refresh their lease.

Like with BOOTP, options convey the addresses of DNS servers, etc.

BOOTP / DHCP Options (RFC 1497)



Magic number, used to identify a RFC conformant option list.

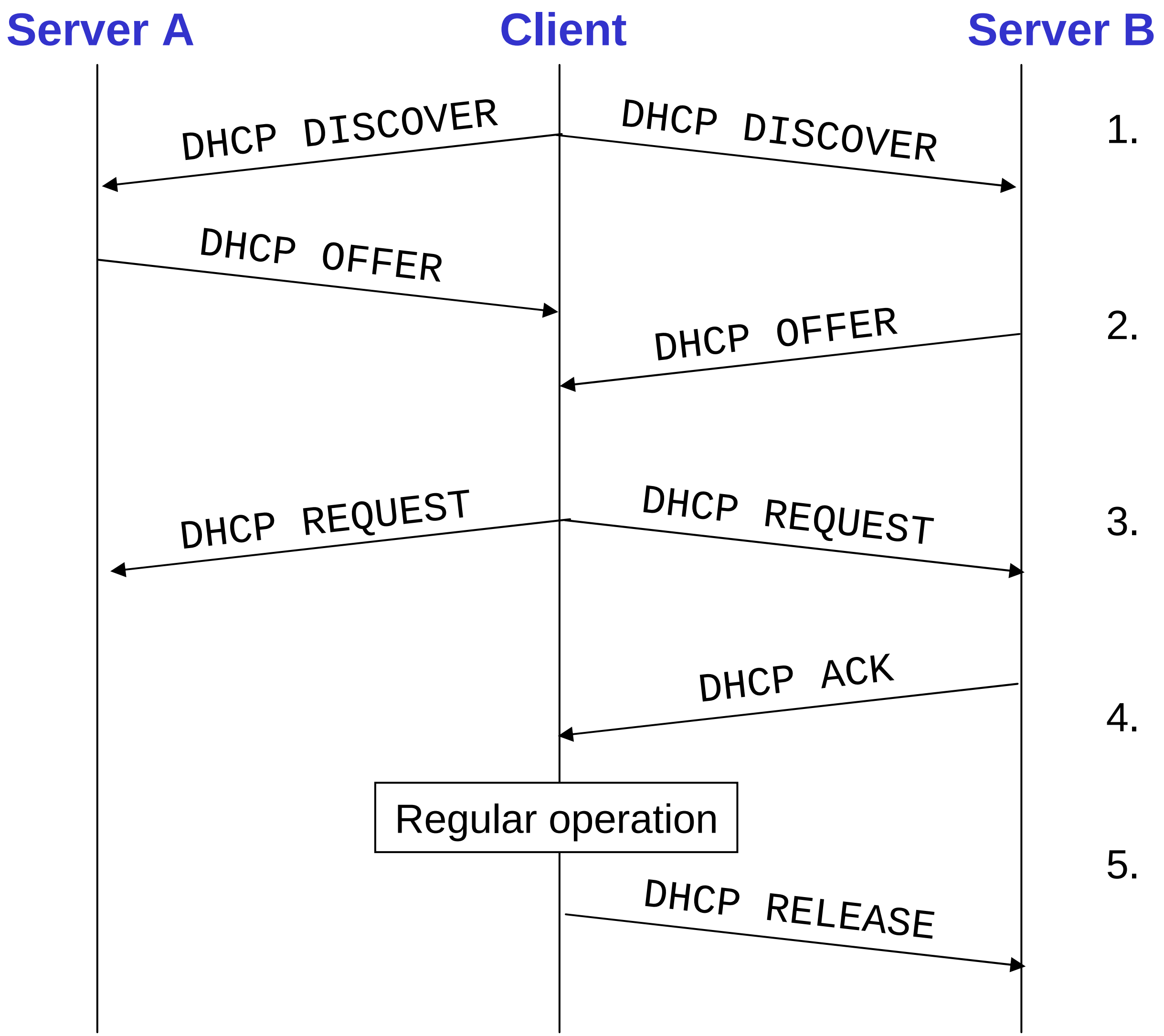
Subnet mask option

List of routers option

List of DNS servers option

RFC 1497 and others list many options. Some have fixed length such as the subnet mask, some have variable length. The latter indicate the length of the respective list immediately after the tag.

DHCP Operation (RFC 1531)

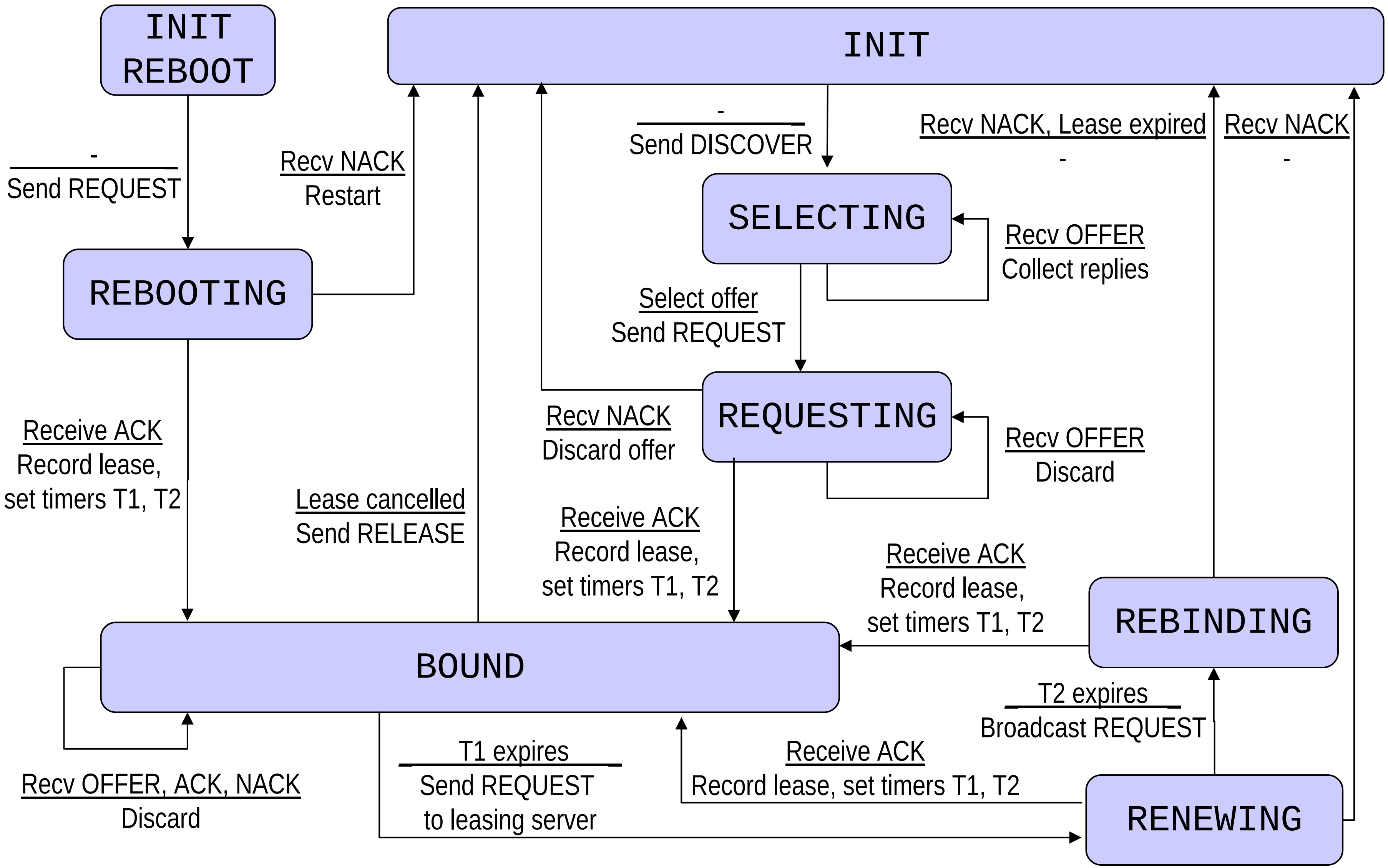


1. Client broadcasts message to discover servers.
2. Servers determine configuration and offer available addresses.
3. Client collects replies and requests selected configuration.
4. Server acknowledges request.
5. Client releases address. Server may keep record of binding.

DHCP Message Overview

- DHCPDISCOVER - Client broadcast to locate available servers.
- DHCPOFFER - Server to client in response to DHCPDISCOVER with offer of configuration parameters.
- DHCPREQUEST - Client message to servers either (a) requesting offered parameters from one server and implicitly declining offers from all others, (b) confirming correctness of previously allocated address after, e.g., system reboot, or (c) extending the lease on a particular network address.
- DHCPACK - Server to client with configuration parameters, including committed network address.
- DHCPNAK - Server to client indicating client's notion of network address is incorrect (e.g., client has moved to new subnet) or client's lease as expired
- DHCPDECLINE - Client to server indicating network address is already in use.
- DHCPRELEASE - Client to server relinquishing network address and cancelling remaining lease.
- DHCPINFORM - Client to server, asking only for local configuration parameters; client already has externally configured network address.

DHCP State Diagramm



Layering Issues

- ARP is a network layer protocol
  - It is performed by the OS network subsystem.
  - It uses its own protocol type next to IP.
- BOOTP & DHCP are a network layer protocols …
  - Even though they are performed by user land processes.
  - Even though they use IP/UDP packets.
- Routing protocols are network layer protocols …
  - Even though they are performed by user land processes.
  - Even though they use IP/UDP packets.
- The same argument applies for DNS, the domain name system …

The classification of DHCP and DNS into the network layer is ***not*** commonly accepted. Some other place these protocols in the application layer.

Zero Configuration Networking (RFC 3927)

Connect two IP hosts with a network cable

With DHCP a single cable is not enough. The hosts must either be configured manually, or the network needs to contain a DHCP server.

Solution: Automatic Private IP Addressing (APIPA)

1. Automatic Allocation of IP address without DHCP server .
2. Name resolution without DNS server.
3. Automatic finding of services in network without directory server.
4. Existing, already configured infrastructure should not be affected.

ZeroConf Operation

Host chooses randomly* an address in 169.254.0.0/16, the "link local" block.

Host sends ARP requests for this address:

3 times …

Sender Address: 0.0.0.0
Receiver Address: 169.254.xxx.xxx

Host announces its address via ARP:

2 times …

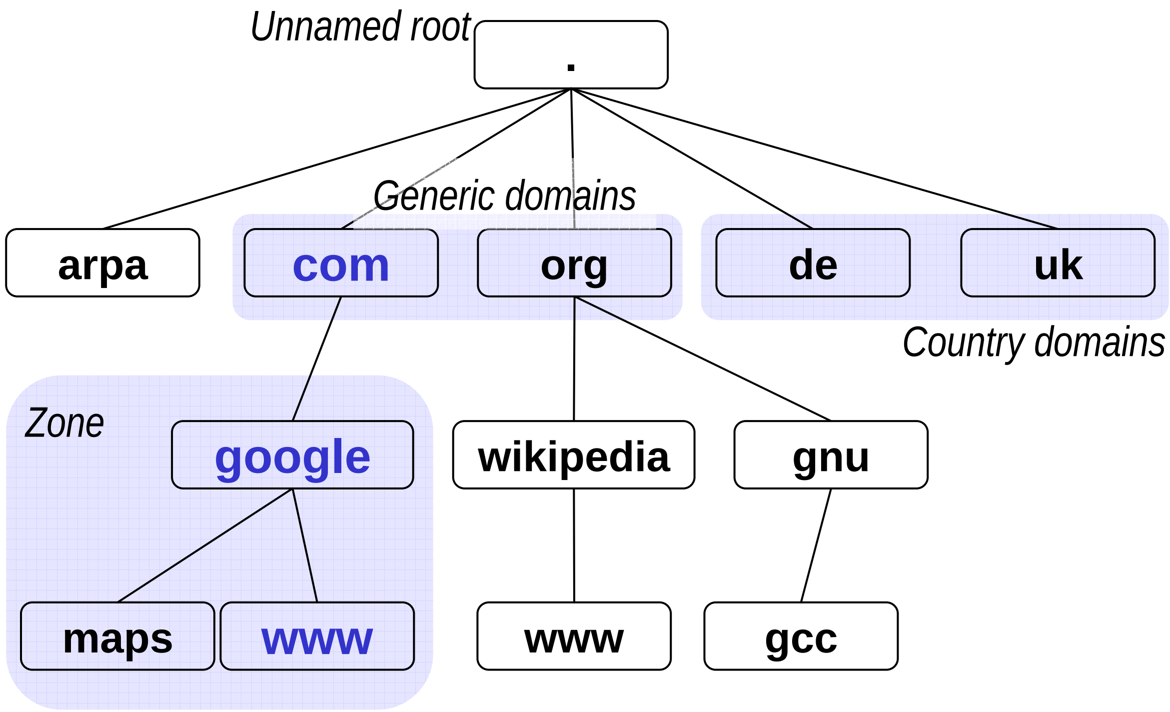
Sender Address: 169.254.xxx.xxx
Receiver Address: 169.254.xxx.xxx

* Ideally, a host would only choose an address once and reuse after all subsequent boots.

Domain Name System (RFC 882 / 883)

- The domain name system (DNS) is a distributed, hierarchical data base.
- DNS stores the association of names to IP addresses.
- DNS assumes that for each domain (=zone) a ‘primary’ server is manually administrated. DNS then distributes this data to other DNS servers and hosts where it is cached.
- Today, DNS is not only used to resolve names to host addresses, but also to discover services (see later).
- Moreover, DNS has become an efficient approach to split the logical resource locator, such as a server name, from the resource's actual physical location.
- This is used to support mobile hosts as well as to balance the load among potentially globally distributed servers.

DNS – Overview



13 DNS root servers answer requests for the top level domains.

Top Level Domains (=TLD)

Domains for organizations and people.

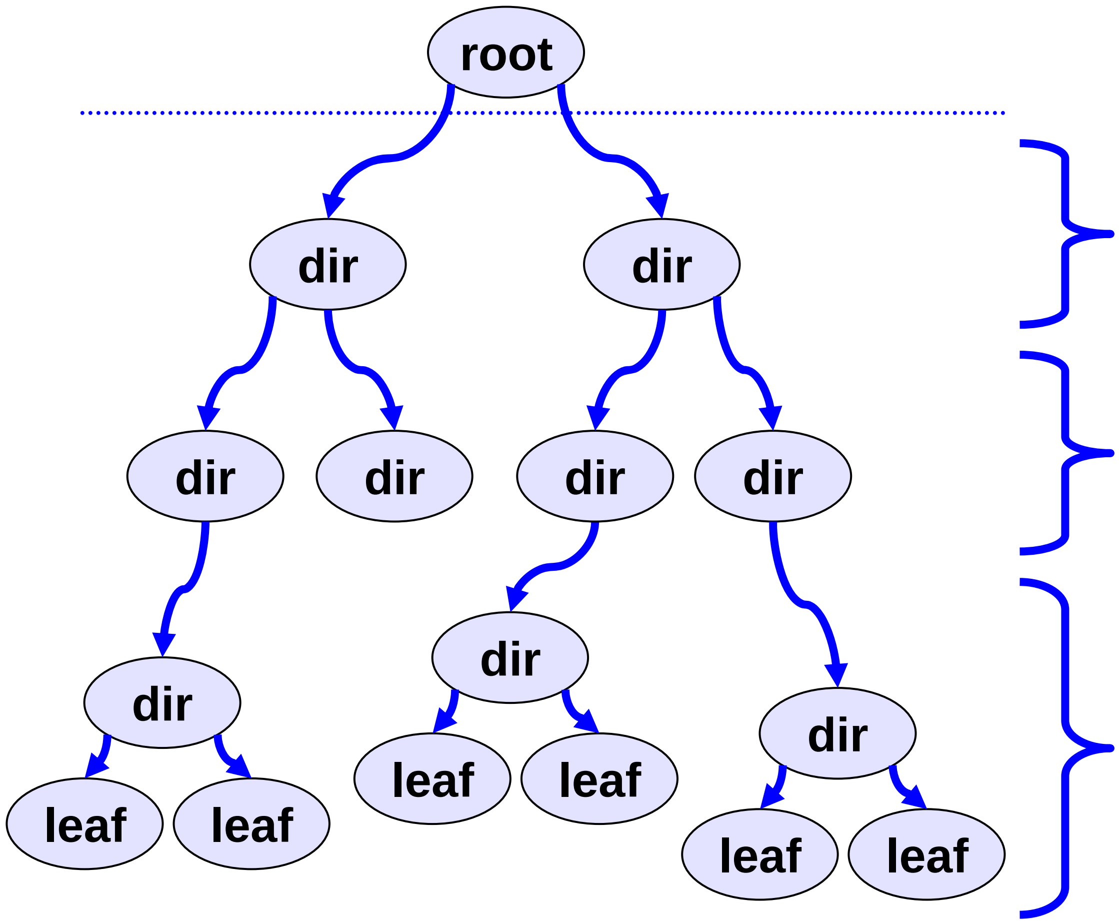
Subdomains and/or host names, often for services.

www.google.com.

= fully qualified domain name

Domain names are dot-separated lists of labels with of up to 63 alpha-numerical (incl. the minus sign) characters each. RFC 3490 extends domain names to unicode. A fully qualified domain name may consist of up to 255 characters.

Namensauflösung (1)



Closure Mechanism,
keine Änderungen!

Global Layer,
sehr seltene Änderungen

Administrational Layer,
seltene Änderungen,
daher lange Cache-Lebensdauer möglich.

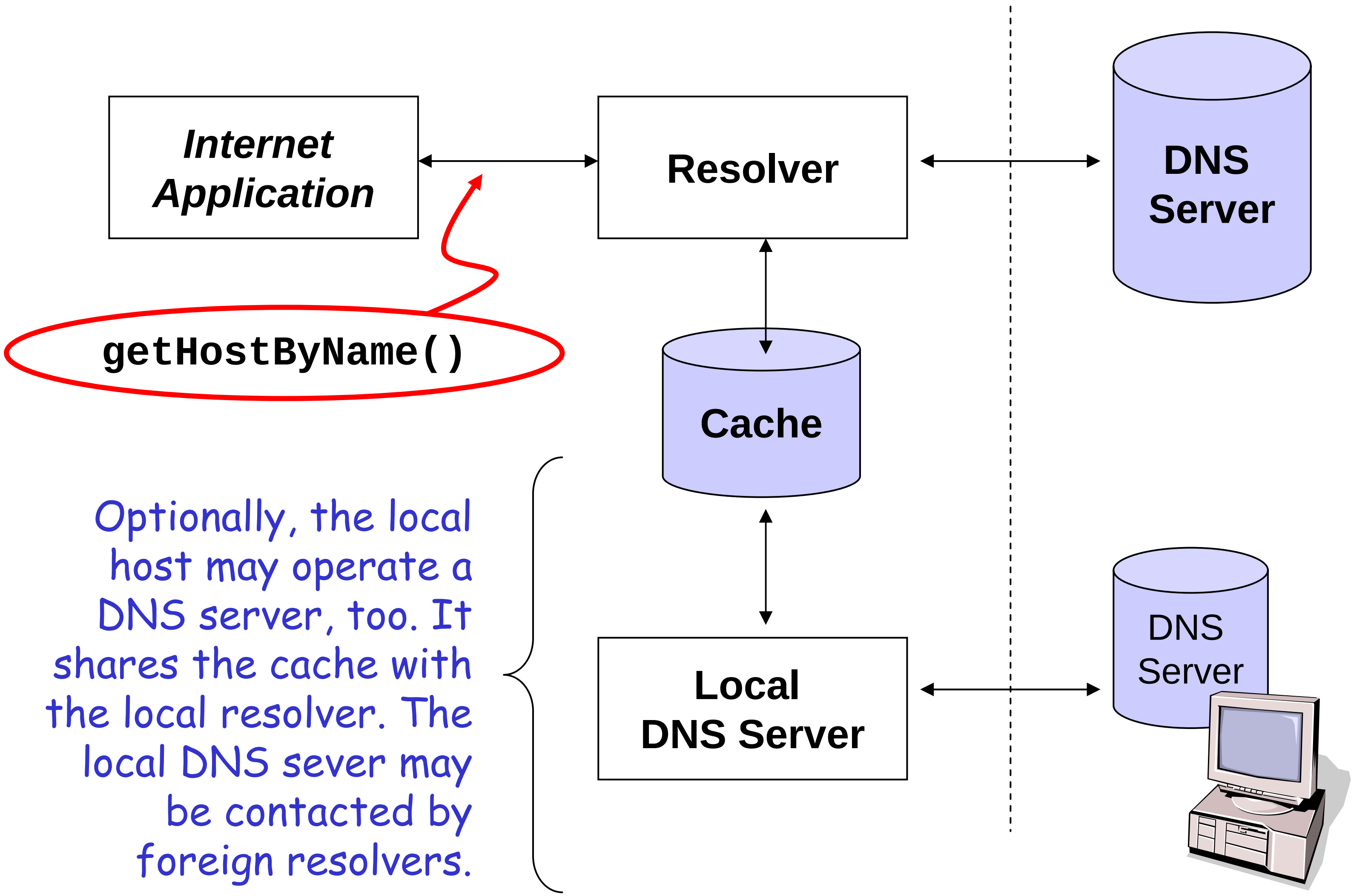
Managerial Layer,
häufige Änderungen, kurze Cache-Lebensdauer bzw. überhaupt kein Caching

Rasche Veränderlichkeit der Zuordnung eines Namens zu einer Adresse beruht u.a. auf der Mobilität der Endgeräte bzw. ihrer Benutzer.

Namensauflösung (2)

- Die Namensauflösung liefert die Adresse des durch einen Pfad (=Namen) im Namensraum bezeichneten Knotens.
- Bei einem Dateisystem ist die Addresse z.B. der Sektor einer Festplatte. DNS Namen werden zu IP-Adressen aufgelöst.
- Dazu sind zwei grundverschiedene Schritte erforderlich:
  - Zunächst muss der Wurzelknoten des Namensraums gefunden werden bzw. bekannt sein (=***closure mechanism***).
  - Davon ausgehend kann dann schrittweise (iterativ oder rekursiv) der Name aufgelöst werden.
- Das Auffinden des Wurzelknotens wird meist implizit gelöst:
  - Beispiel UNIX Dateisystem: Wurzelknoten ist im ersten i-node gespeichert. Dessen Adresse kann direkt berechnet werden.
  - Beispiel Domain Name System: Wurzelknoten tragen öffentlich bekannte IP-Adressen.

DNS – Overview



DNS Operation

- DNS operates on UDP and TCP port 53.
  - DNS messages in UDP packets are truncated to 512 bytes. This is indicated by a flag.
  - The querying resolver should then repeat its query using TCP.
- When using UDP, the querying resolver must repeat its query after some timeout to cope with packet loss.
- Zone transfers from a primary to a secondary name server are performed via TCP.
  - Primary name server are administered to contain up-to-date authorative data for the respective zone.
  - Secondary name servers are authorative for a zone, but they obtain their data from a primary server of that zone.
- Other server may use cached data obtained from the authorative servers. But the authorative server are the anchor of all DNS information.

SOA (=start of authority) Entries in the DNS

The zone we talk about.

TTL of this information.

Primary of this zone.

Send administrative inquires to root@examle.com.

```
example.com. 3600 SOA dns.example.com. root.example.com. (
                    1999022301 ; serial YYYYMMDDnn
                    86400      ; refresh ( 24 hours)
                    7200       ; retry ( 2 hours)
                    604800     ; expire ( 1 week)
                    172800 )   ; minimum ( 2 days)
```

- Resolvers and/or name servers caching this data may keep it up to <TTL> seconds before they must query the data again.
- Note that the entry has a serial number that must be increased manually when the configuration is edited. Typically, people choose to base this number on the current date.
- Refresh, retry & expire give the intervals how often the secondary server will perform a zone transfer and how often it will retry if the primary is temporarily unreachable. If the primary remains unreachable the secondary server will expire this data and not answer queries any more.

Further DNS Entries

- An A record defines a 32bit IPv4 address.
- Similarly, an AAAA record defines an IPv6 address.
- The PTR record is used for inverse queries. Such queries yield, for example, the DNS name of an IP address.
- The CNAME record give the canonical name of a resource. This is used for alias names. The name to which an inverse query resolves is the canonical name.
- The MX record denotes the mail exchanger of a domain. Thereby, mail addresses like „john@example.com“ can be resolved to a host like „www.example.com“ resolves to a host.
  - A domain may contain serveral MX records. A numerical preference value before each name allows hosts to determine the primary and fall-back mail exchanger.
- The NS record gives the authorative name servers of a domain.

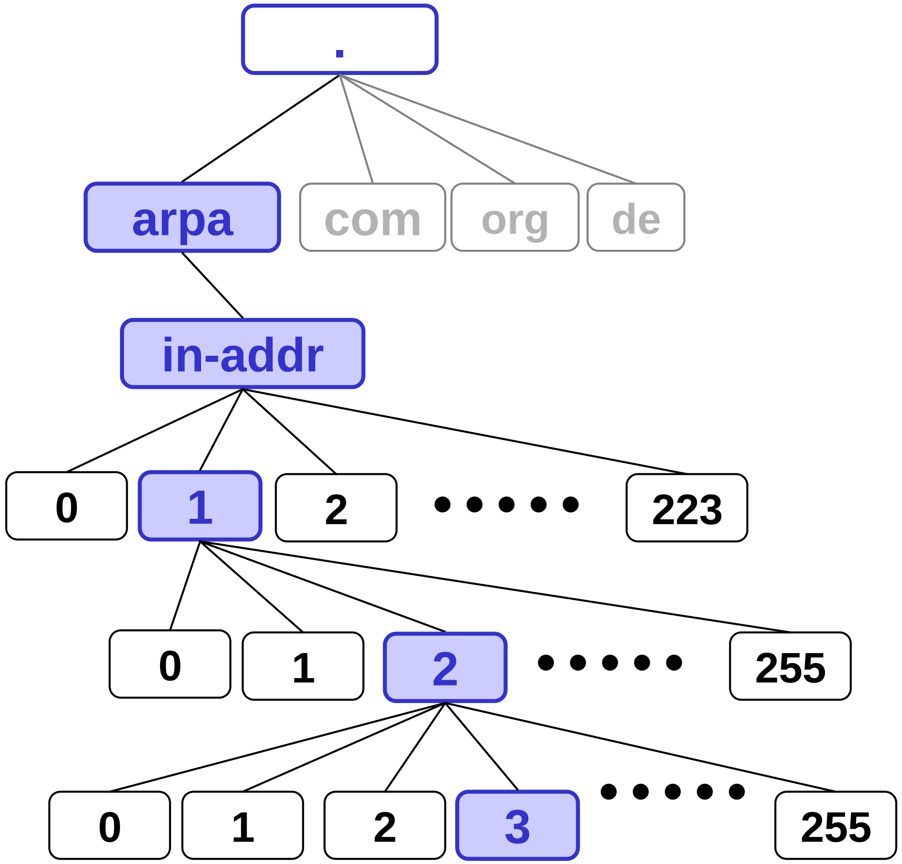
Inverse Lookup

Often, an application wants to know the DNS name of the host that tries to open a connection. Examples are

- consistency checks to block spoofed addresses, and
- human readable log entries.

To this end, the DNS maintains a special domain, the in-addr.arpa domain.

- Each host should have two entries, one in the generic or country domain and one in the in-addr.arpa domain.
- If the in-addr.arpa entry is missing, inverse lookups will fail.



3.2.1.in-addr.arpa.
stores the inverse lookup data for hosts in the network 1.2.3.x

DNS Message Format

<table>
<tr><td colspan="9">Identification</td></tr>
<tr><td>QR</td><td>Opcode</td><td>AA</td><td>TC</td><td>RD</td><td>RA</td><td>zero</td><td colspan="2">R-Code</td></tr>
<tr><td colspan="9">Number of queries</td></tr>
<tr><td colspan="9">Number of answers (=resource records)</td></tr>
<tr><td colspan="9">Number of authority resource records</td></tr>
<tr><td colspan="9">Number of additional resource records</td></tr>
</table>

16 bit

QR = Query (0) or
Answer (1)

AA = Authorative answer?

TC = Message truncated?

RD = Recursion desired

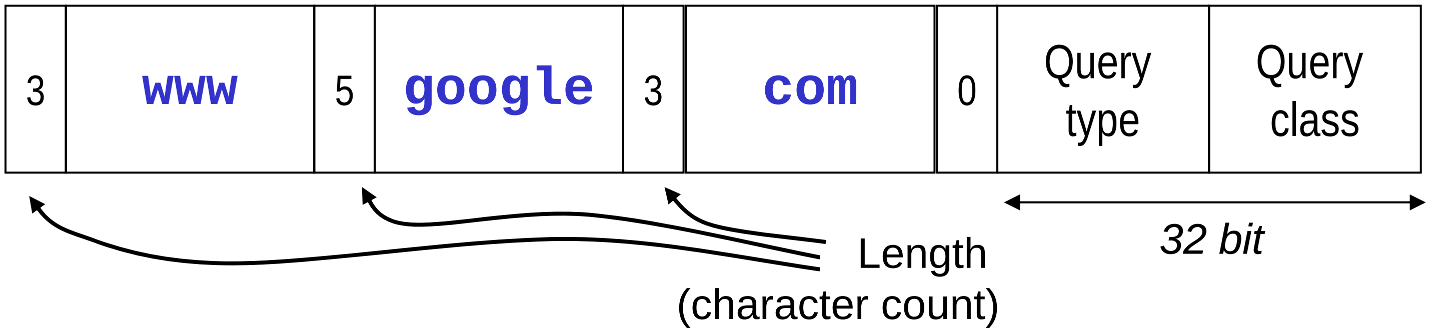
RA = Recursion available

Opcode = standard query,
inverse query, or
status request

R-Code = Error code

For queries, typically, the number of questions is one and the number of resource records (answers, authority, and additional) is zero. For answers, the queries are repeated.

Example Query



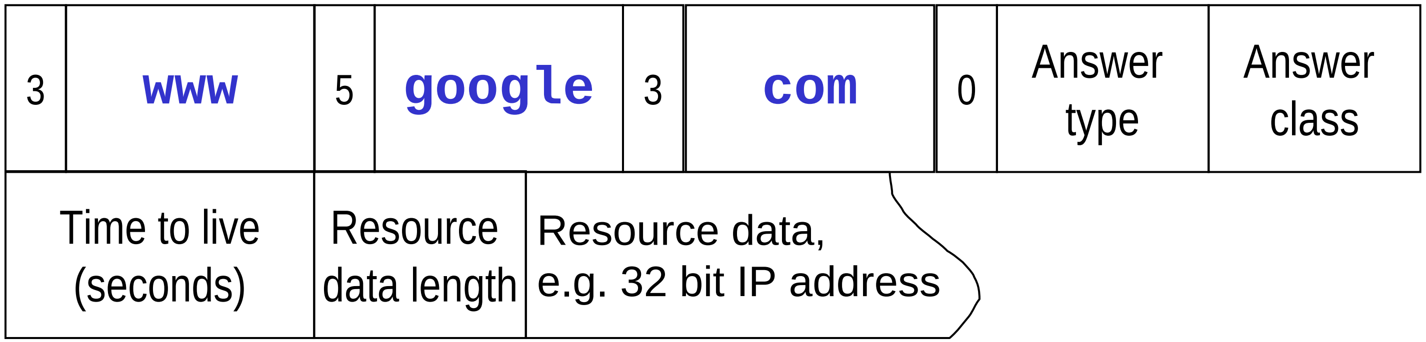
Query type

1	A	a host
2	NS	an authoritative name server address
5	CNAME	the canonical name for an alias
6	SOA	marks the start of a zone of authority
11	WKS	a well known service description
12	PTR	a domain name pointer
13	HINFO	host information
14	MINFO	mailbox or mail list information
15	MX	mail exchange

Query class

1	IN	Internet

Example Answer



- Responses repeat the query. (Note that the message ID of a response matches the ID of the request.)
- If the responder is authorative it fills the answer fields. Otherwise, it gives a pointer to an authorative server.
- It may additionally give supplementary information, e.g. about the IP addresses of the authorative servers.

A Practical Example (1)

```
host -v www.tum.de

Trying "www.tum.de"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12031
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 2, ADDITIONAL: 3

;; QUESTION SECTION:
;www.tum.de.                    IN      A

;; ANSWER SECTION:
www.tum.de.             86347   IN      CNAME   tum.www.ze.tu-muenchen.de.
tum.www.ze.tu-muenchen.de. 7147 IN      CNAME   io.ze.tum.de.
io.ze.tum.de.           7151    IN      A       129.187.39.54

;; AUTHORITY SECTION:
ze.tum.de.              7151    IN      NS      w3projns.ze.tum.de.
ze.tum.de.              7151    IN      NS      dns1.lrz-muenchen.de.

;; ADDITIONAL SECTION:
dns1.lrz-muenchen.de.   31728   IN      A       129.187.19.183
dns1.lrz-muenchen.de.   31722   IN      AAAA    2001:4ca0:0:100:0:53:1:1
w3projns.ze.tum.de.     7151    IN      A       129.187.39.1

Received 216 bytes from 141.3.10.90#53 in 3 ms
```

A Practical Example (2)

```
host -v tum.de

Trying "tum.de"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31550
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 8

;; QUESTION SECTION:
;tum.de.                        IN      MX

;; ANSWER SECTION:
tum.de.             29542   IN      MX      100 mailrelay2.lrz-muenchen.de.
tum.de.             29542   IN      MX      100 mailrelay1.lrz-muenchen.de.

;; AUTHORITY SECTION:
tum.de.             29542   IN      NS      dns1.lrz-muenchen.de.
tum.de.             29542   IN      NS      dns2.lrz-muenchen.de.
tum.de.             29542   IN      NS      dns3.lrz-muenchen.de.
tum.de.             29542   IN      NS      deneb.dfn.de.

;; ADDITIONAL SECTION:
mailrelay1.lrz-muenchen.de. 29537 IN    A       129.187.254.106
mailrelay2.lrz-muenchen.de. 29542 IN    A       129.187.254.102
dns1.lrz-muenchen.de.   29543   IN      A       129.187.19.183
dns1.lrz-muenchen.de.   29537   IN      AAAA    2001:4ca0:0:100:0:53:1:1
dns2.lrz-muenchen.de.   29543   IN      A       141.40.9.211
dns2.lrz-muenchen.de.   29537   IN      AAAA    2001:4ca0:0:100:0:53:1:2
dns3.lrz-muenchen.de.   29543   IN      A       129.187.5.2
deneb.dfn.de.           5583    IN      A       192.76.176.9

Received 324 bytes from 141.3.10.90#53 in 4 ms
```

A Practical Example (3)

```
host -v tum.de

Trying "tum.de"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64185
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;tum.de.                         IN      A

;; AUTHORITY SECTION:
tum.de.              10800   IN      SOA
                                         dns1.lrz-muenchen.de.
                                         hostmaster.lrz-muenchen.de.
                                         2006092211
                                         21600
                                         1800
                                         3600000
                                         86400

Received 89 bytes from 141.3.10.90#53 in 11 ms
```

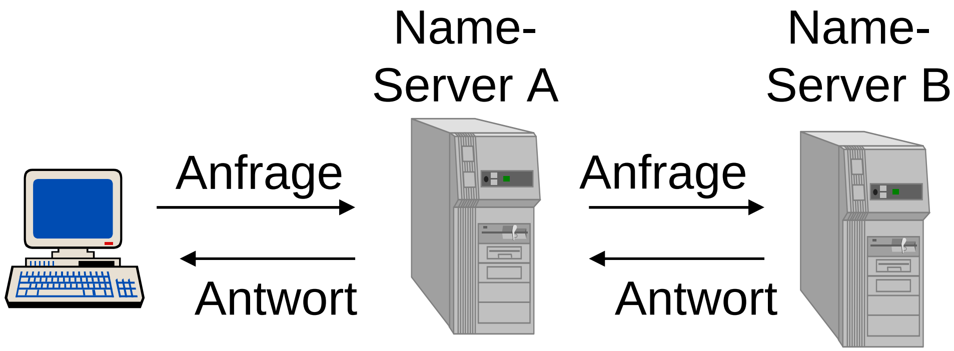
A Practical Example (4)

```
host -v 129.187.39.54

Trying "54.39.187.129.in-addr.arpa"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8983
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

;; QUESTION SECTION:
;54.39.187.129.in-addr.arpa.    IN      PTR

;; ANSWER SECTION:
54.39.187.129.in-addr.arpa. 7200 IN     PTR     io.ze.tum.de.

;; AUTHORITY SECTION:
39.187.129.in-addr.arpa. 7200   IN      NS      dns1.lrz-muenchen.de.
39.187.129.in-addr.arpa. 7200   IN      NS      w3projns.ze.tum.de.

;; ADDITIONAL SECTION:
dns1.lrz-muenchen.de.   29125   IN      A       129.187.19.183
dns1.lrz-muenchen.de.   29119   IN      AAAA    2001:4ca0:0:100:0:53:1:1
w3projns.ze.tum.de.     4548    IN      A       129.187.39.1

Received 185 bytes from 141.3.10.90#53 in 17 ms
```

Rekursive & Iterative DNS-Anfragen

Rekursive Anfrage

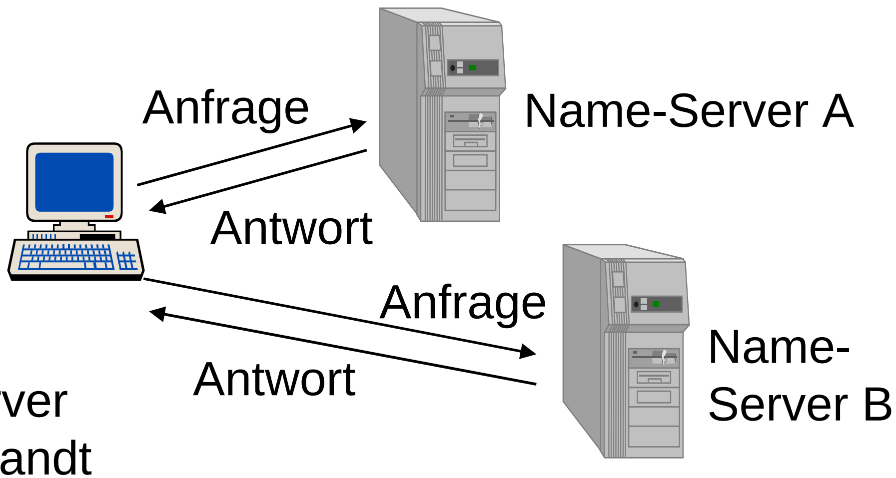
- Anfrage wird an Name-Server A gestellt.
- Falls die Antwort im Cache vorliegt, kann sie sofort beantwortet werden.



- Andernfalls löst A die Anfrage über Name-Server B auf.

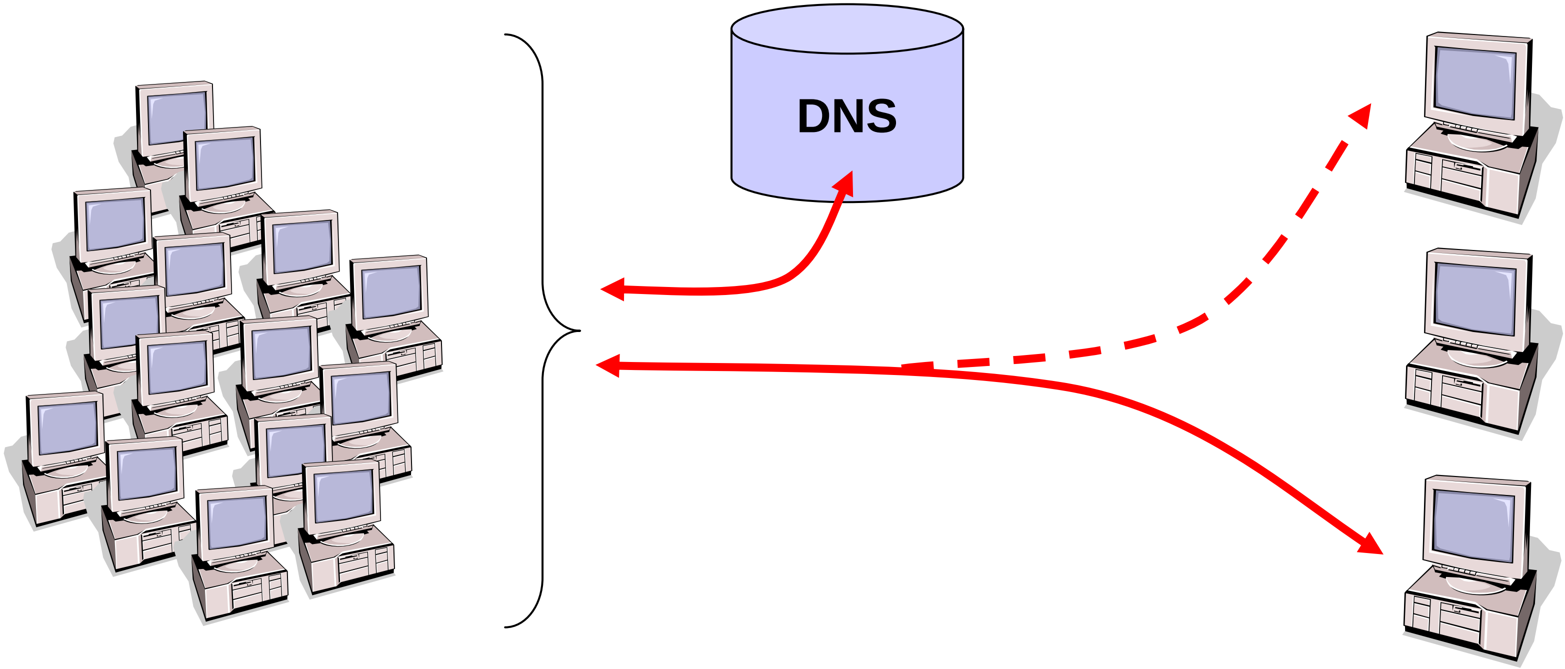
Iterative Anfragen

- Der Domainname wird schrittweise aufgelöst, indem sich z.B. das Endsystem selbst der Reihe nach durch die Hierarchie fragt.



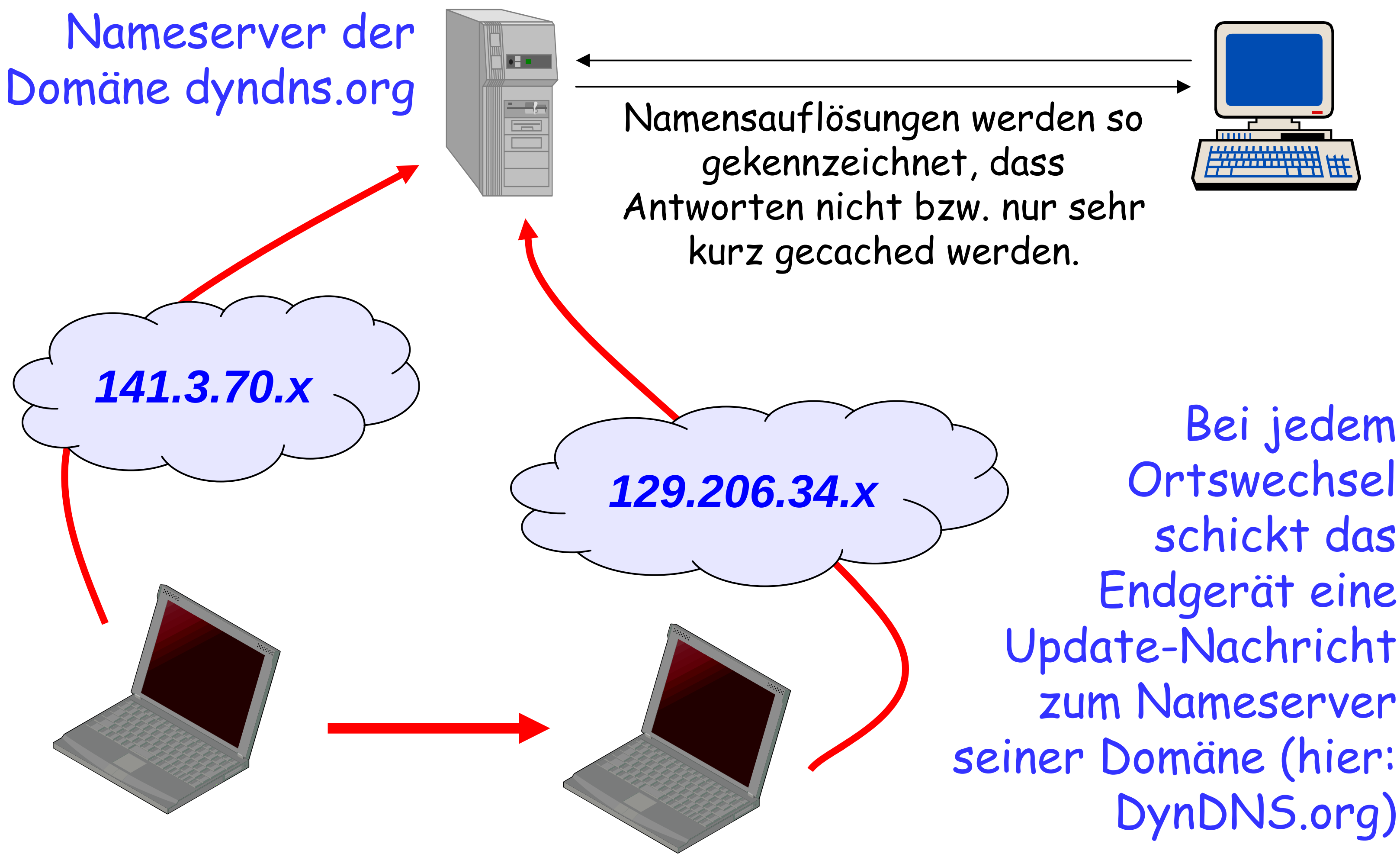
Bei jeder Stufe einer rekursiven Abfragekette kann der dortige Nameserver entscheiden, welches Verfahren angewandt werden soll.

DNS based Load Balancers

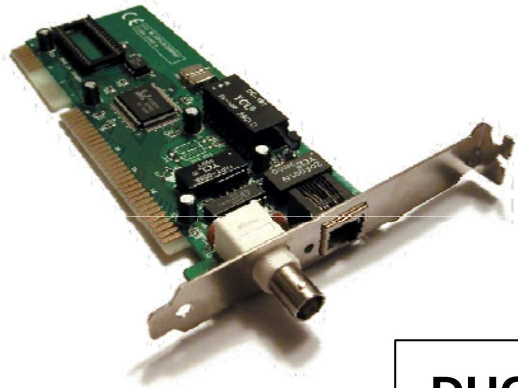
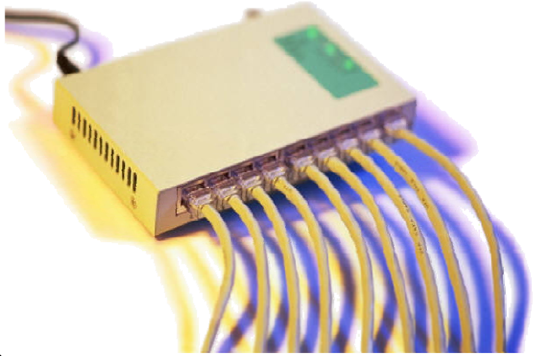


- Today, DNS is (mis-) used for various purposes.
- For example, a DNS server can respond with different IP addresses so that subsequent traffic is directed to different machines.
- Another example is that the DNS locates the host that sent the query and answers with a machine that is close to that host.

DynDNS


Nameserver der
Domäne dyndns.org
Namensauflösungen werden so gekennzeichnet, dass Antworten nicht bzw. nur sehr kurz gecached werden.
141.3.70.x
129.206.34.x
Bei jedem Ortswechsel schickt das Endgerät eine Update-Nachricht zum Nameserver seiner Domäne (hier: DynDNS.org)

Host Location & Addressing

MAC address	IP address	DNS name
		

DHCP →

← ARP

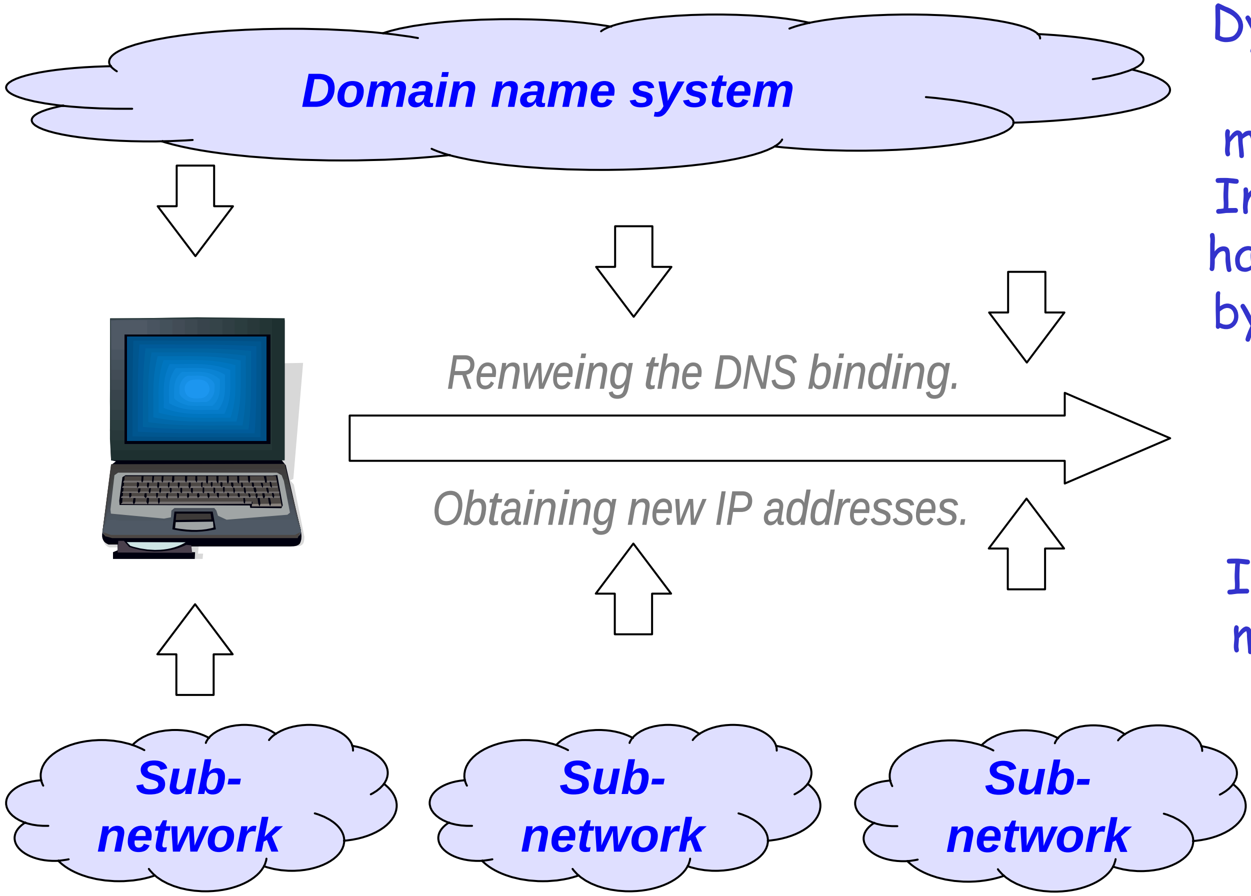
↔ DNS

Permanent, assigned by the hardware vendor.	Structural property that is determined by the network topology. NICs need to assume an IP address of their sub-network. Hence, IP addresses are host location dependent.	Legal right, assigned to a legal or natural person; manifested in a DNS database entry.
„Vendor controlled“	„Provider controlled“	„Lawyer controlled“

Host Location & Addressing

- MAC addresses (=hardware addresses) are permanently bound to the network interface card (NIC).
  - Example: IEEE 802 addresses such as Ethernet, Bluetooth, or WiFi addresses. Vendors buy address blocks from the IEEE registration authority.
- IP addresses are bound to the host location in the network.
  - They belong to the sub-network and are assigned by the respective network administrator (manually or via DHCP). Hosts determine the binding via the address resolution protocol (ARP).
  - Network address assignment is a hierarchical administrative process: Network administrators obtain address block from their Internet service provider (ISP). In the end, all address blocks are obtained from the Internet Assigned Numbers Authority (IANA).
- Domain names are belong to natural or legal persons and are temporarily bound to IP addresses via the domain name system (DNS).
  - Typically, the binding of a DNS name to an IP address is stable over many months or even years. The TTL of the DNS record determines how fast a binding can be changed.
  - Sometimes, the binding is short-lived to accommodate dial-up hosts with dynamically assigned addresses.
  - Domain name assignment is a hierarchical administrative process. Top level domains are assigned by IANA.

Mobile Hosts



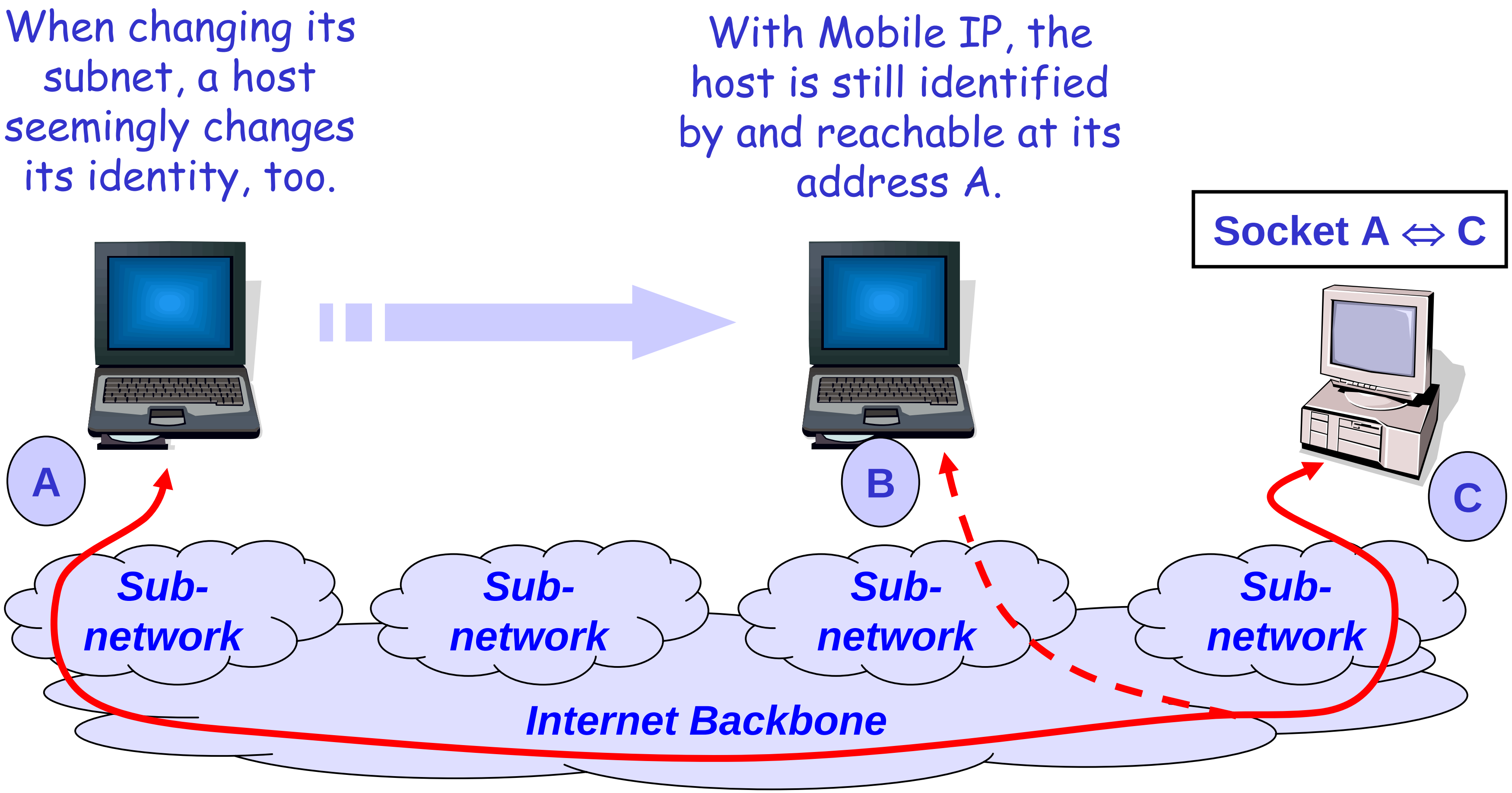
Dynamic DNS protocols can reflect a hosts movement through the Internet. However, the handover time is limited by the DNS cache TTL. Moreover, existing connections break.

In each subnet, a host must acquire a new IP address via DHCP.

Hence, DynDNS is only suitable for dial-up hosts, not mobile hosts!

Note: Here, we talk about macro-mobility, that is hosts changing sub-networks.
Micro-mobility, that is host mobility within a subnet should to be handled by layer 2 mechanisms.

Mobile IP

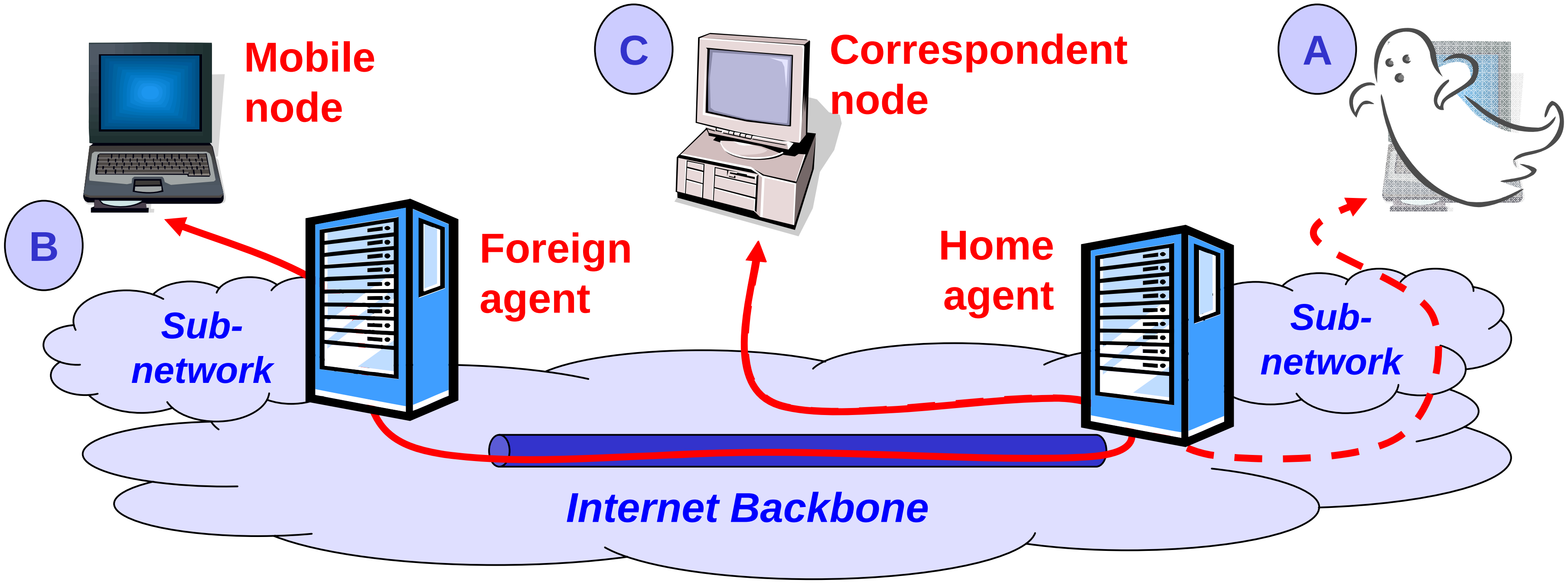


Mobile IP occupies a different point in the design space than DynDNS. It keeps the double role of IP addresses as locator and identifier.

Mobile IP – Terminology

- Mobile node – A node that can „move around in the Internet“.
  - A mobile node belongs to one network in the Internet, its home network.
  - It can detach from this network and connect to other networks while keeping its IP address.
- Home agent – Router in the home network that …
  - maintains the binding between the mobile node‘s home address and its care of address and
  - tunnels all packets destined to the mobile node to its current care of address.
- Foreign agent – Router in the foreign network that …
  - provides a care of address to the mobile nodes,
  - updates the binding of the mobile node to its care of address at the node’s home agent, and
  - serves as tunnel end-point for traffic to the mobile node.
- Care of address – The mobile node’s IP address in the foreign network.
- Correspondent node – The node that connects to a mobile node.

Mobile IP – Overview



Binding Maintenance:

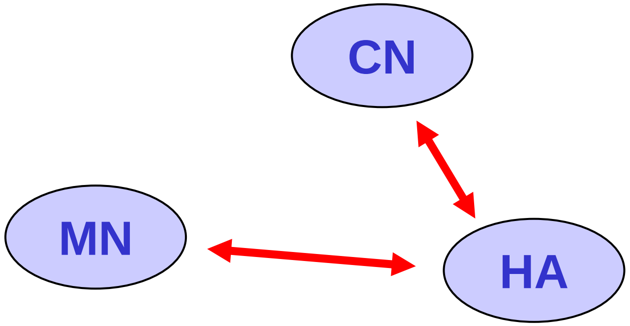
- Upon arriving in subnet B the mobile node requests an address from the network's mobile IP agent. This agent serves as foreign agent for the mobile node.
- The foreign agent informs the mobile node's home agent about the new binding.

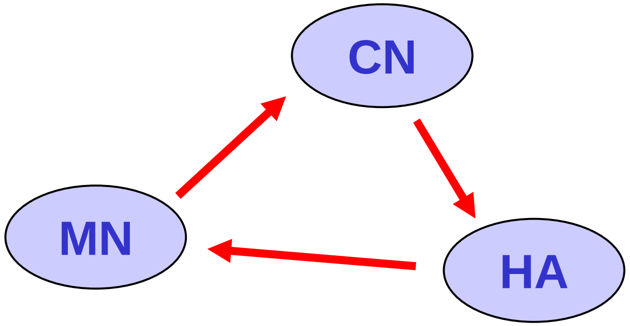
Communication:

- When a C communicates to the mobile node it sends its packets to A.
- The home agent does not forward the packet in its local subnet, but tunnels them to the mobile node's current foreign agent.
- The foreign agent forwards the packets in its local subnet.

Triangle Routing

Reverse tunneling	Triangle routing



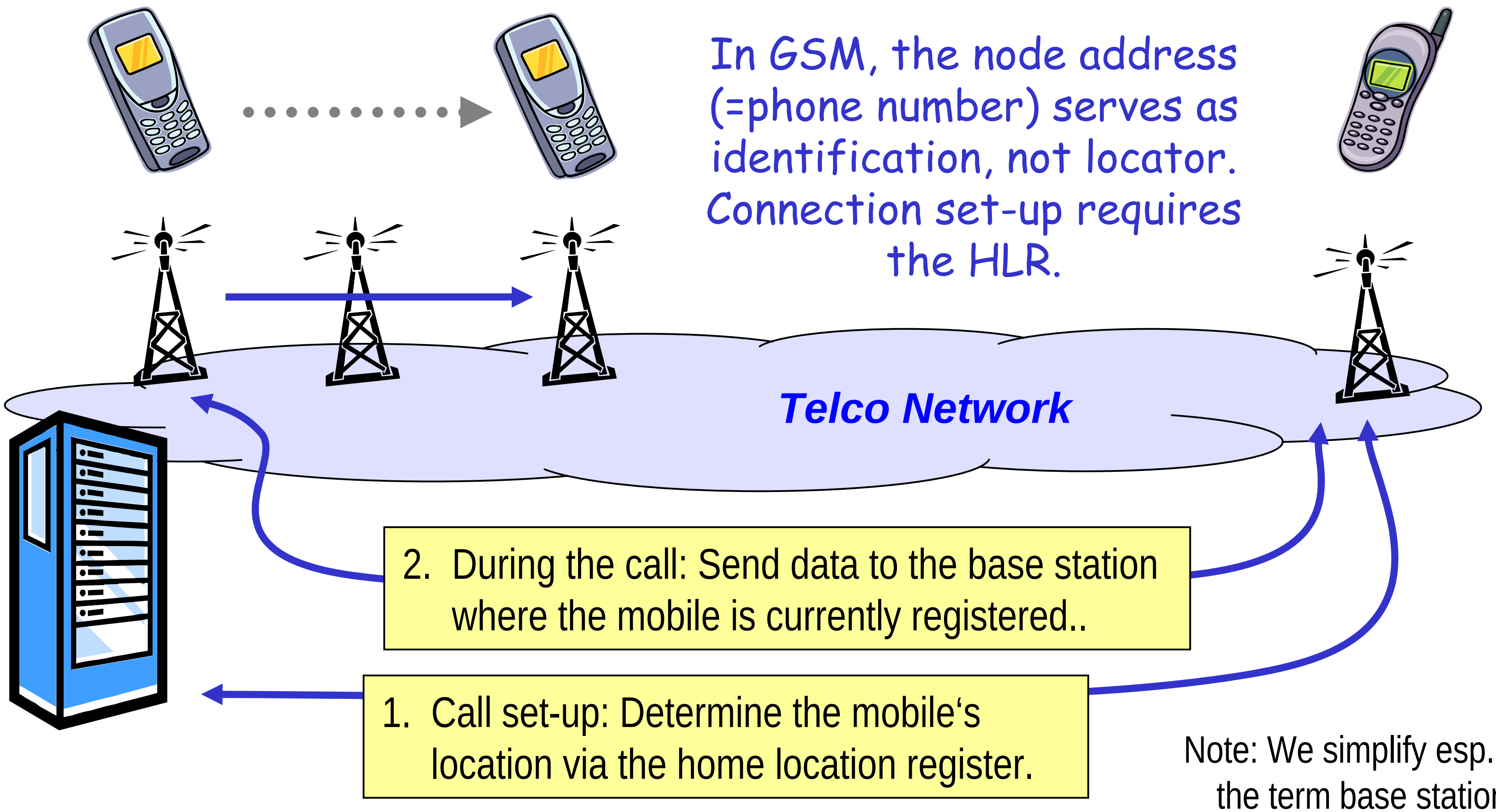


- Both the home agent and the foreign agent tunnel the traffic from/to the correspondent node.
- Traffic leaving a subnet always bears a topologically consistent source address.

- Traffic to the mobile node is tunneled, but traffic from the mobile node is sent directly to the correspondent node.
- Triangle routing creates more overhead than regular routing, but less overhead than reverse tunneling.

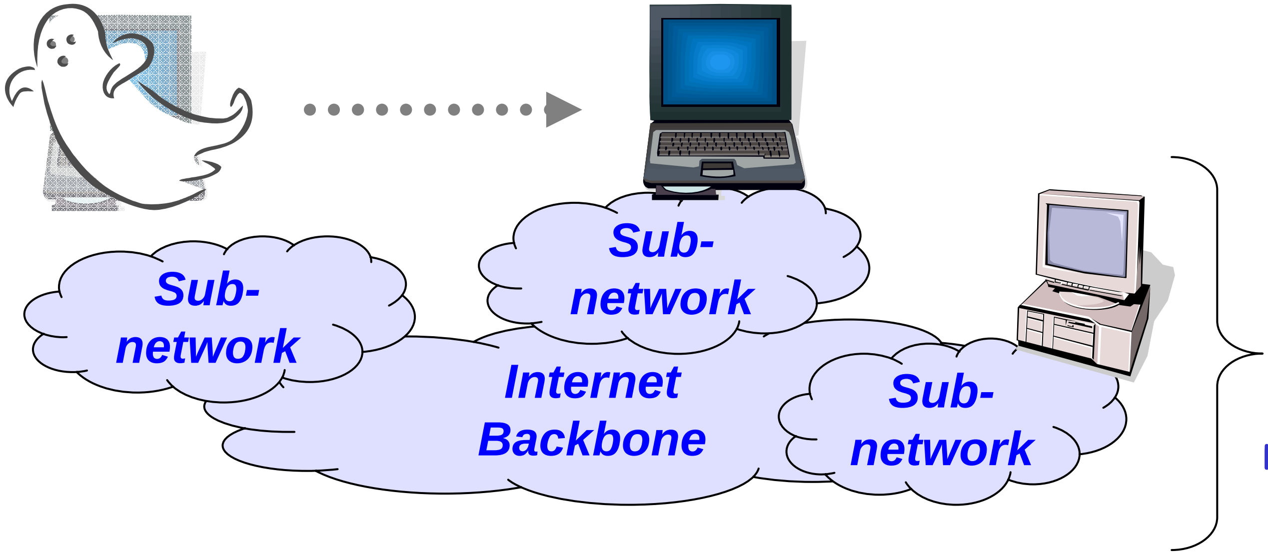
Compare to GSM …

3. Hand over: When the mobile moves, the base station forwards the data to the new location.



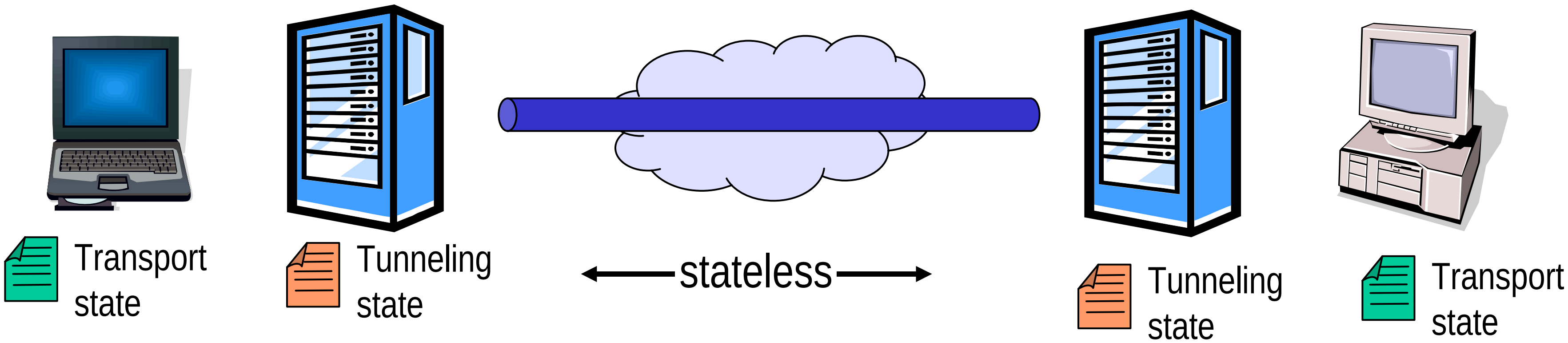
Note: We simplify esp. wrt. the term base station.

Mobile IP and the End-to-End Principle

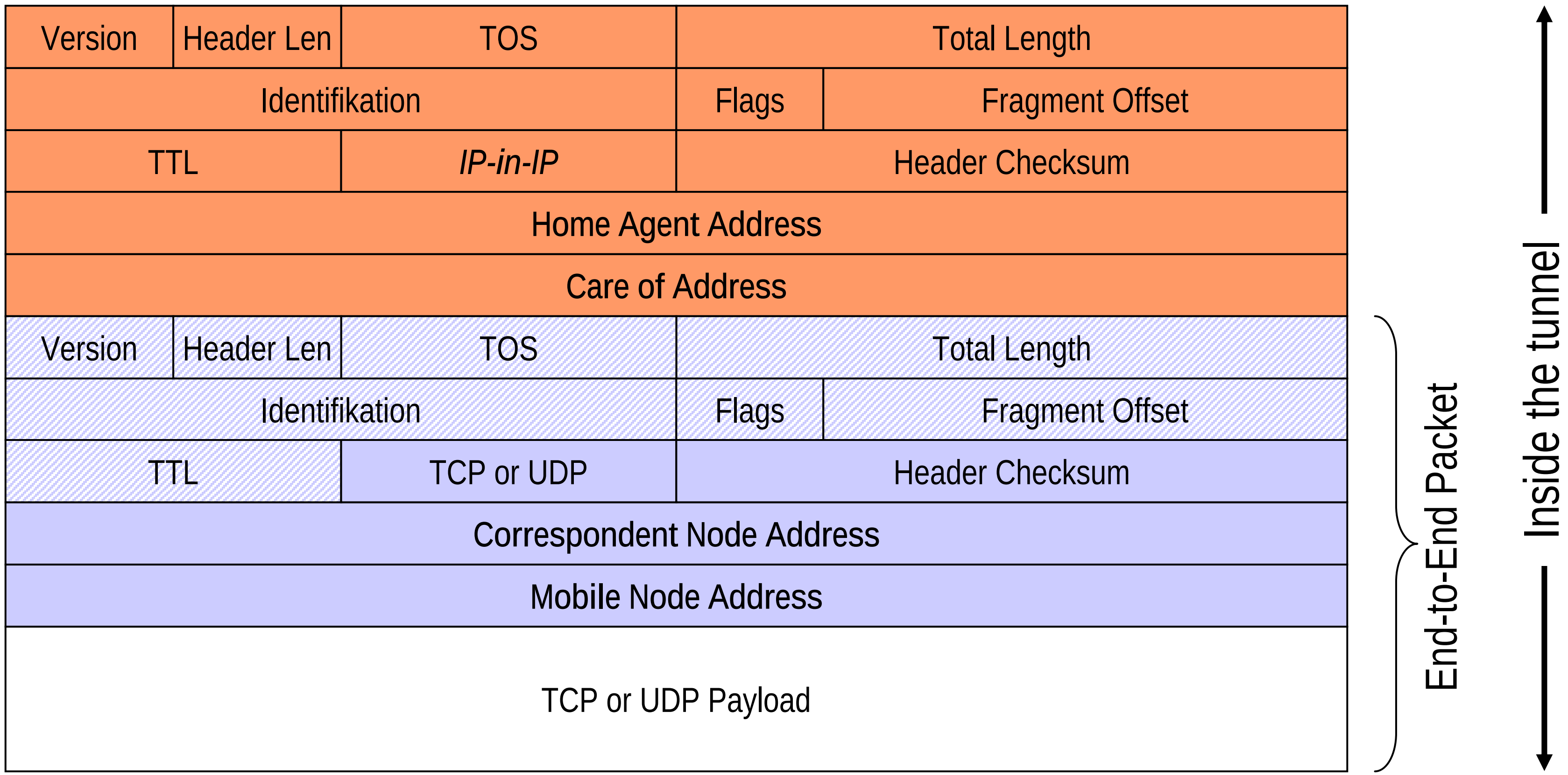


Internet Routing needs topologically assigned addresses to ensure aggregation of routing information.

Home agent and foreign agent hide the actual traffic source / destination inside a tunnel. The agents must keep the respective state. When an agent fails, the routing fails.



IP Tunneling



IP header for tunneling

Original IP header

Header fields that can be reused in the tunneling header

RFC 2003

ICMP Router Discovery (RFC 1256)

Router Advertisment

Type = 9	Code = 0	Checksum
Num Addrs	Addr Len	Lifetime
Address [0]		
Preference [0]		
Address [1]		
Preference [1]		

• • • • • • • • • • • • • • • •

Router Solicitation

Type = 10	Code = 0	Checksum
reserved		

ICMP router discovery is an alternative to DHCP. Router advertisements are broadcast regularly (~30min) in the subnet. Upon booting a host may broadcast a router solicitation.

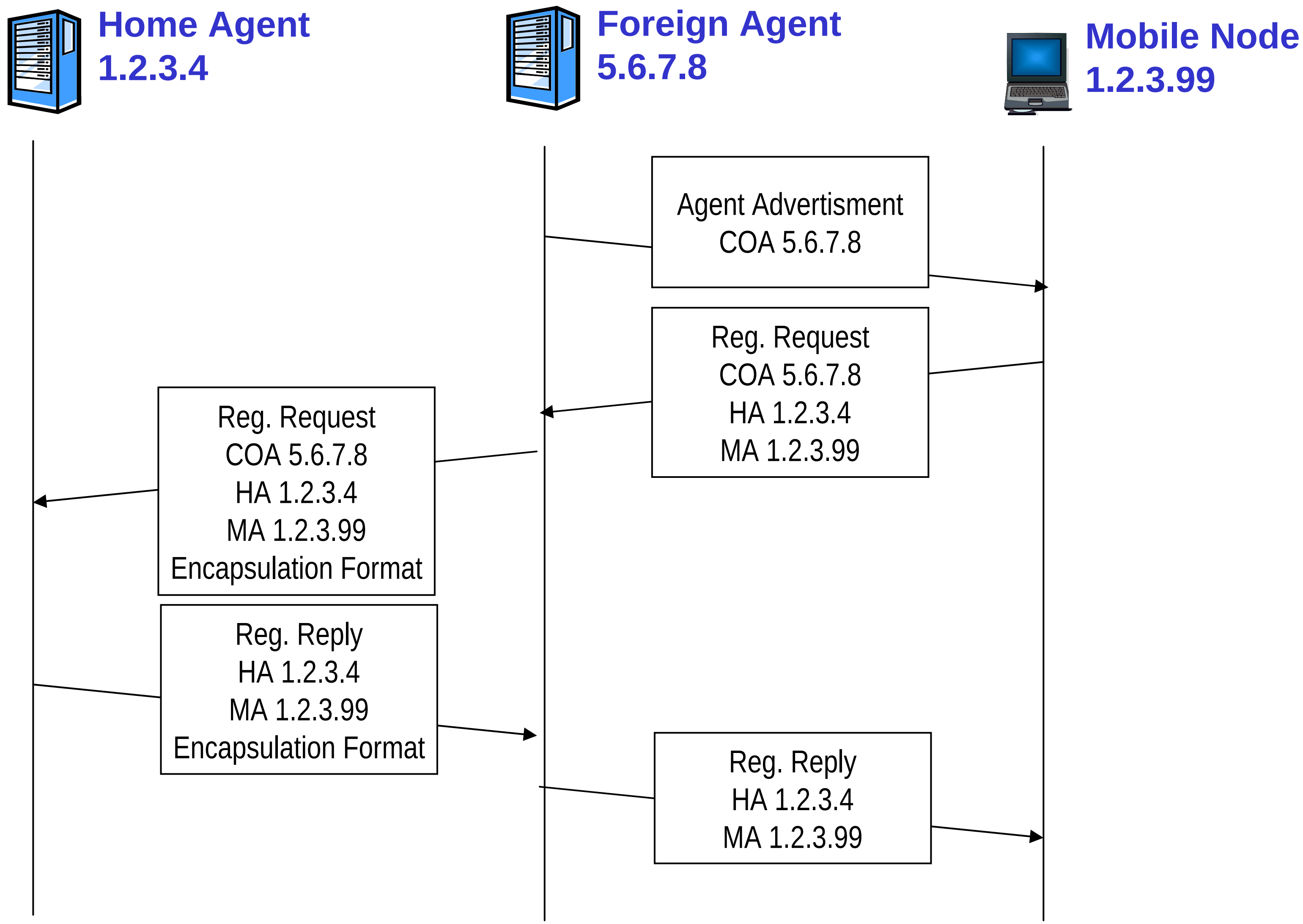
Mobile Agent Advertisment (RFC 2002)

Type = 16	Length	Sequence Number	
Lifetime		Flags	reserved
Care of Address [0]			
Care of Address [1]			

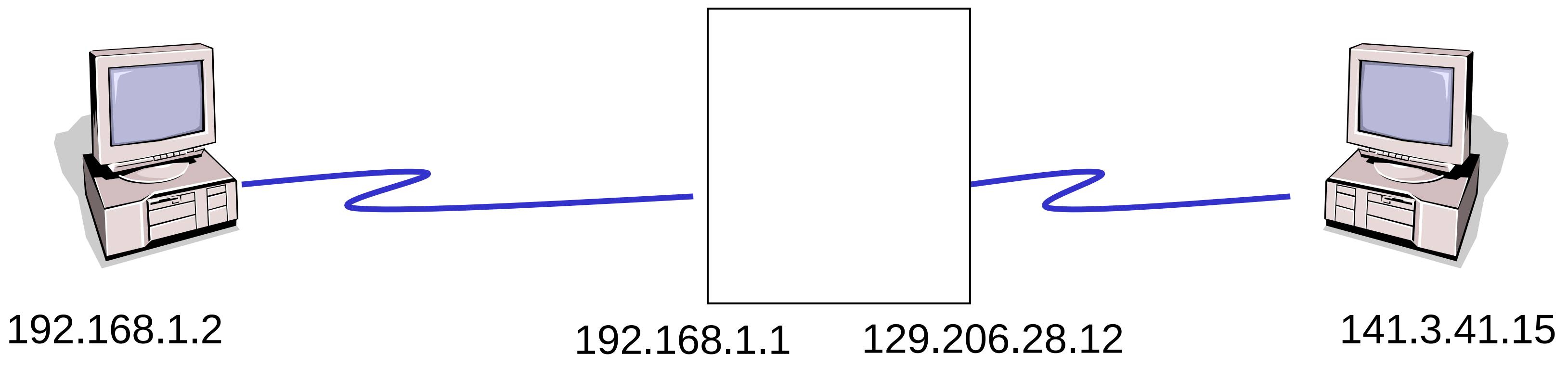
● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

- Extension of router advertisment, carried in the same message.
- Regularly broadcast so that mobile node
  - Learns the agent's address, and
  - can detect if it moved to a new subnet.
- Flags:
  - R = Must register with foreign agent, i.e. no co-location allowed
  - B = Agent busy, i.e. it will not accept new requests, but already bound mobile nodes will still be served
  - H, F = Agent is a home agent or foreign agent
  - M, G, V = Agent supports minimal encapsulation, generic route encapsulation, Van Jacobson header compression

Mobile Node Registration



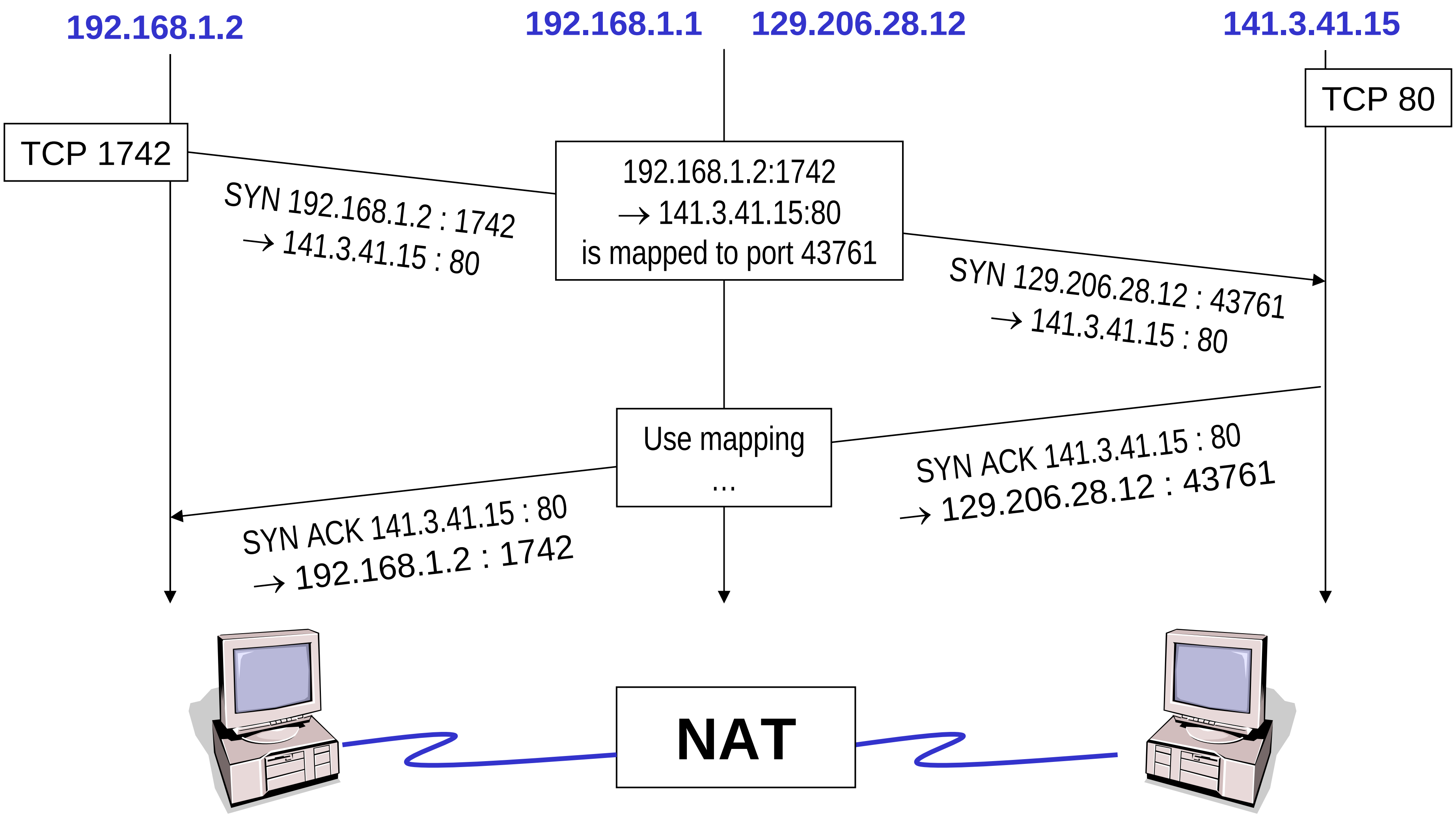
Network Address (and Port) Translation



- NAT (or NAPT) maps private IP addresses (and ports) to public addresses.
- NAPT can hide a private subnet behind only one public address.
- Often, NAPT is used to secure a private network (→stateful firewall).
- NAT keeps the port numbers, i.e. requires one public address per internal host. NAPT translates <address:port> pairs.
- NAPT is often called NAT, too. Thus, normally devices do NAPT even when they are called NAT.

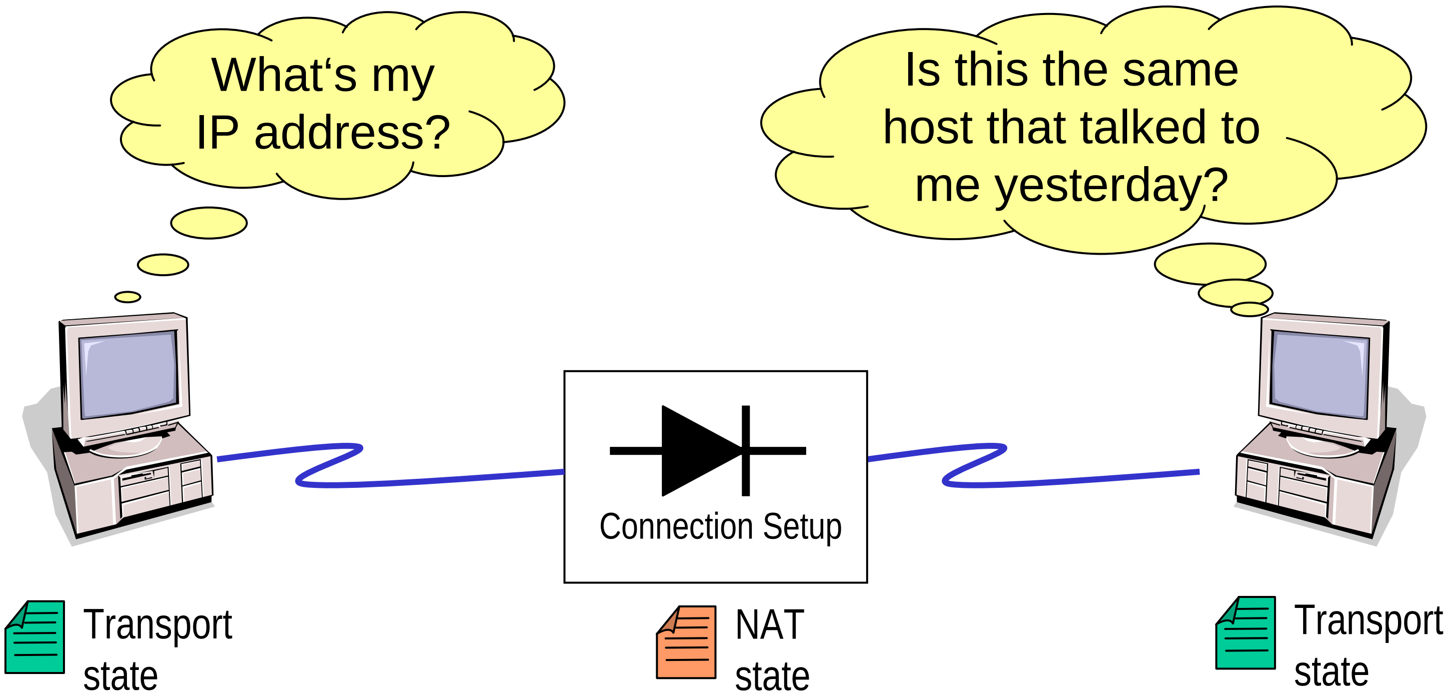
RFC 2663

NAT Example



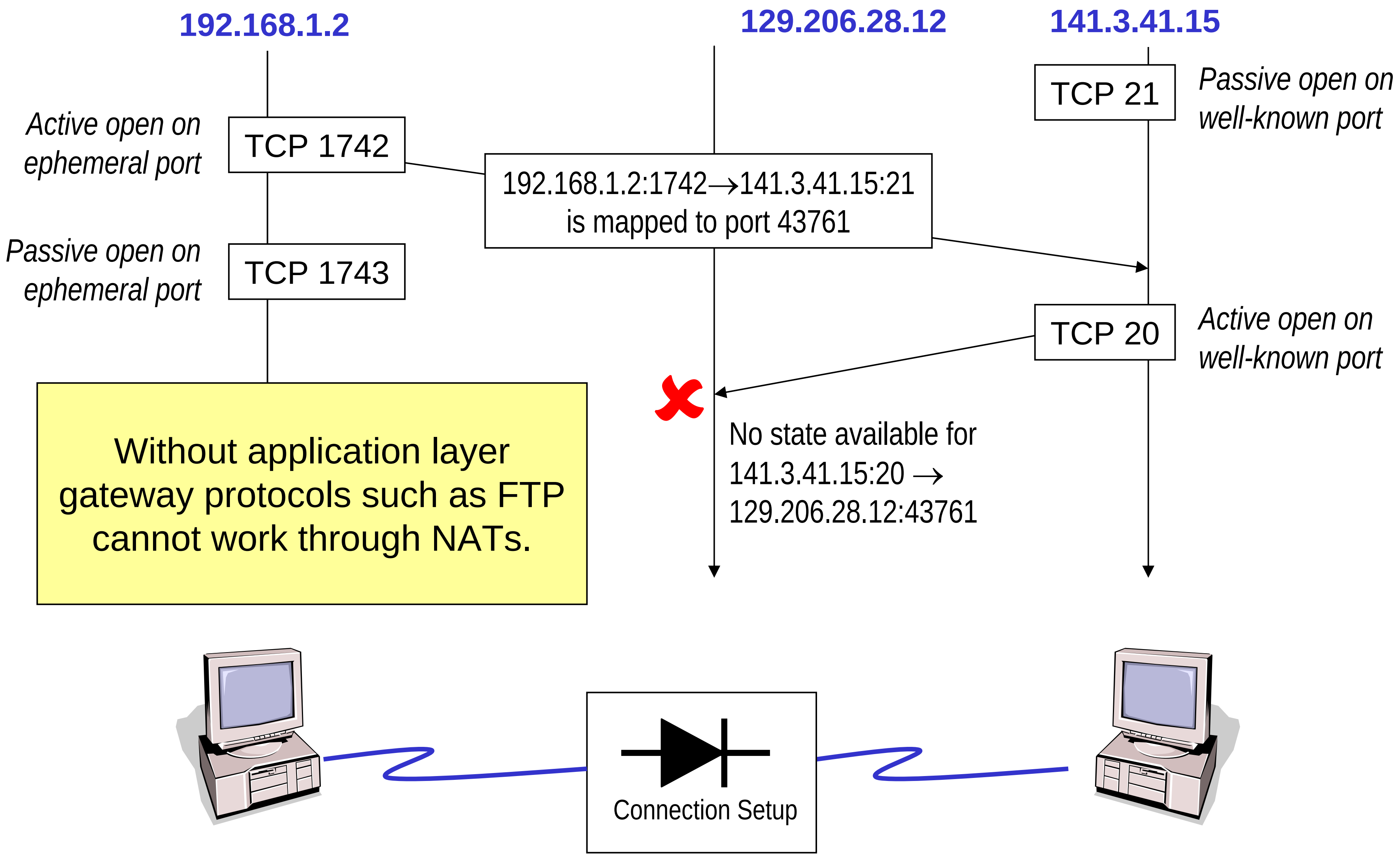
192.168.1.2
192.168.1.1
129.206.28.12
141.3.41.15
TCP 1742
TCP 80
SYN 192.168.1.2 : 1742
→ 141.3.41.15 : 80
192.168.1.2:1742
→ 141.3.41.15:80
is mapped to port 43761
SYN 129.206.28.12 : 43761
→ 141.3.41.15 : 80
Use mapping
...
SYN ACK 141.3.41.15 : 80
→ 129.206.28.12 : 43761
SYN ACK 141.3.41.15 : 80
→ 192.168.1.2 : 1742
NAT

NAT Problems



- NAT box needs to keep state; if the NAT dies the transport connection breaks.
- The NAT‘ed host cannot know its IP address.
- All transport connections must originate from the NAT‘ed host. (Bug or feature!?!)
- Correspondent nodes cannot recognize returning NAT‘ed hosts by their address. (This is true for dynamically assigned addresses, too.)

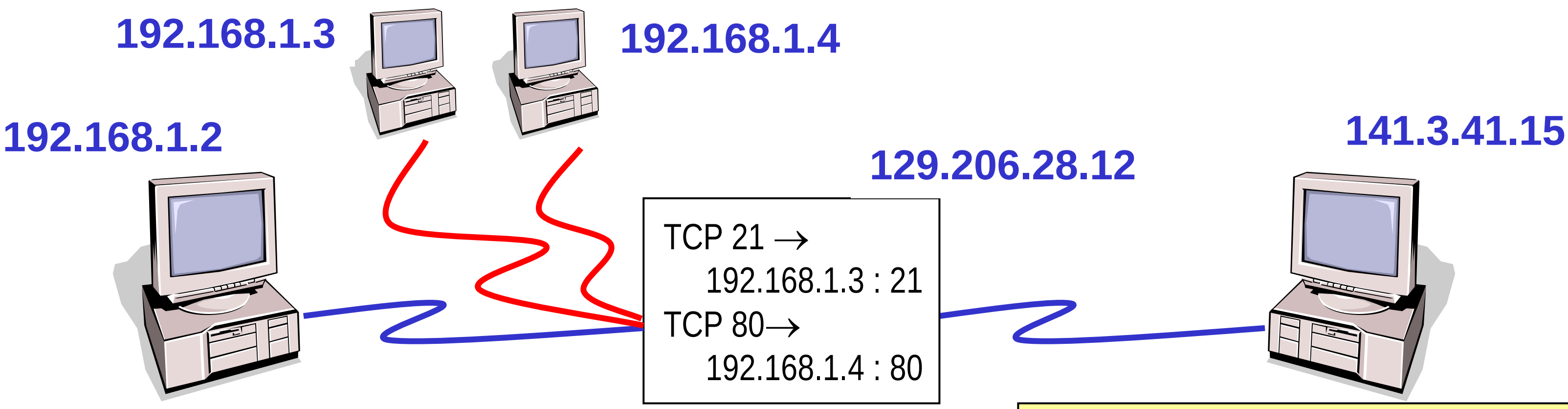
Application Layer Gateway (1)



Application Layer Gateway (2)

- Application Layer Gateways (ALG) use deep packet inspection to obtain required information to set up the NAT state.
  – Example: NAT reads the FTP control channel to learn which ports the clients opens passively.
- Deep packet inspection is tedious to implement because the NAT must understand the application protocol.
- Sometimes, the NAT must revert to an educated guess about the application protocol.
  – Example: FTP servers bind to a well-known port. But any other port would do in principle, and any other application might be bound to the FTP port. The NAT-ALG needs to know (or guess) the actual configuration to work correctly.

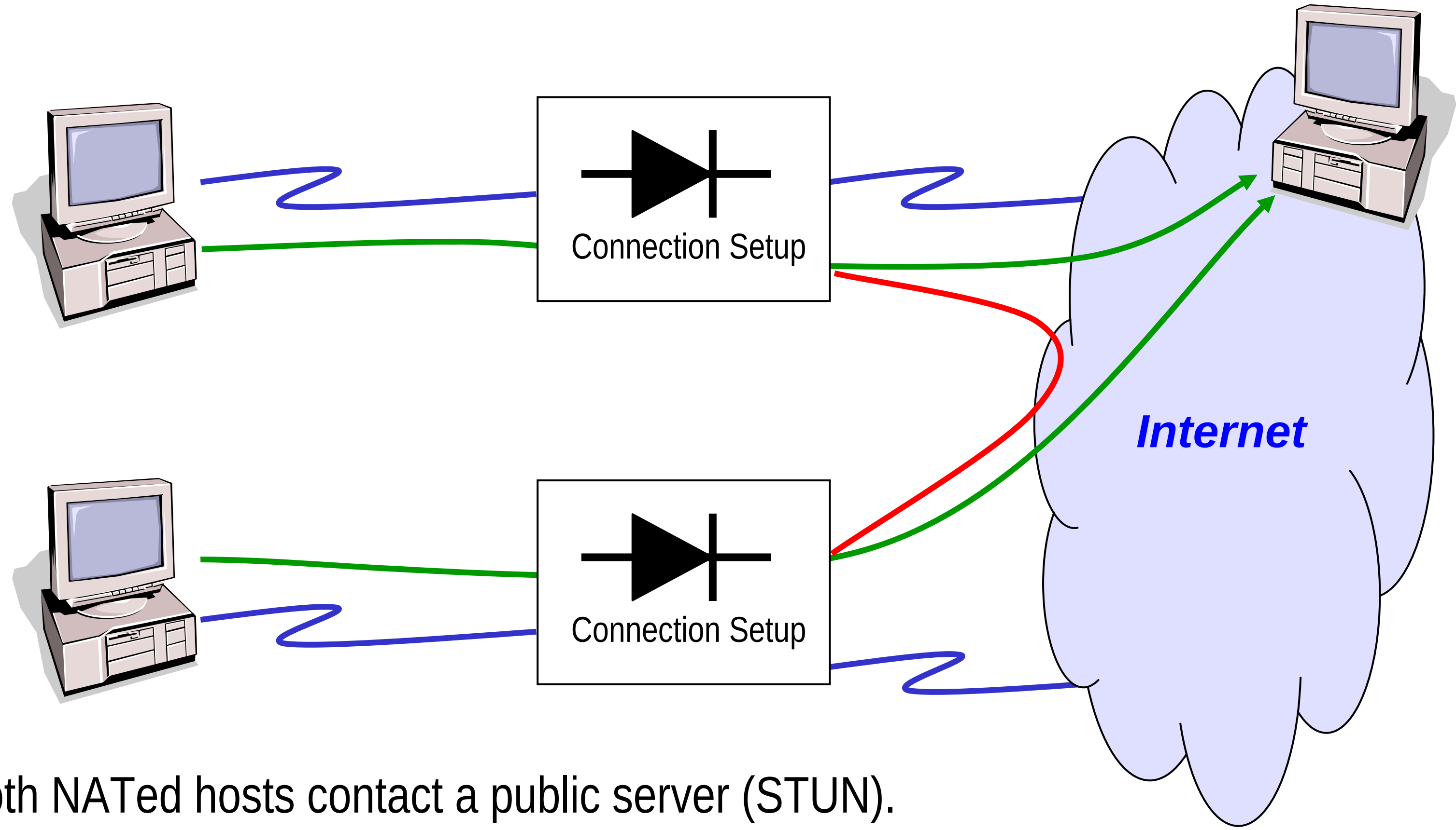
Demilitarized Zones



Outside hosts can 'ftp 129.206.28.12' even though the address is not the actual host running the FTP server.

- NAT state can be learned from the traffic.
  - TCP SYN for connection establishment
  - UDP packets
  - Deep packet inspection
- NAT state can be configured manually, too.
  - Example: All incoming traffic to the well-known FTP server port is sent to one machine in the local network. And similarly for incoming Mail, HTTP, etc. traffic.
- Such static NAT configurations are often called DMZ (=demilitarized zone) configuration because they have similar properties to firewalled DMZ.

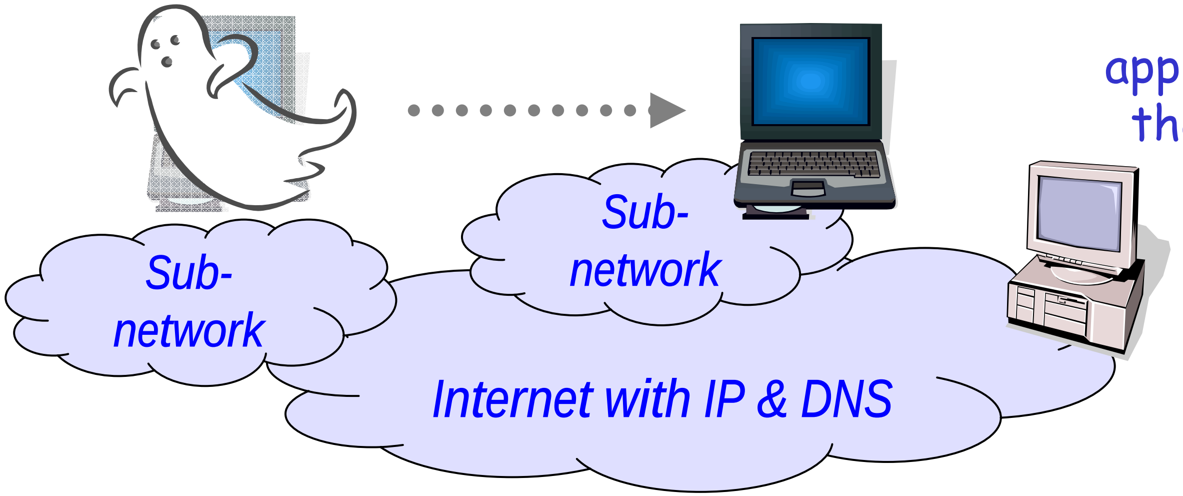
Connecting Two NAT‘ed Hosts



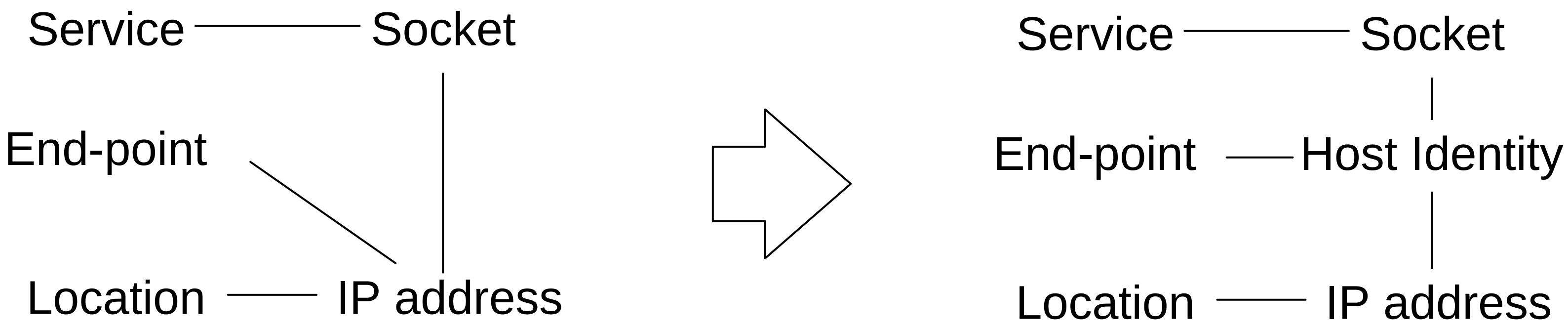
1. Both NATed hosts contact a public server (STUN).
2. The STUN tells the address and port pair it sees.
3. The NATed hosts can use this information to contact each other through the NAT.

Note: This does not work with all NATs because it requires the NAT not to consider the STUN address as part of its state.

Host Identity Protocol (RFC 4423)



How can an application trust in the identity of a host?



Classically, IP addresses were long-lived and the IP address identified a host.

With the Host Identity Protocol (HIP) transport end-points are not bound to the IP address anymore, but to cryptographically secure identifiers.

Questions?

Thomas Fuhrmann

Department of Informatics
Self-Organizing Systems Group
c/o I8 Network Architectures and Services
Technical University Munich, Germany

fuhrmann@net.in.tum.de