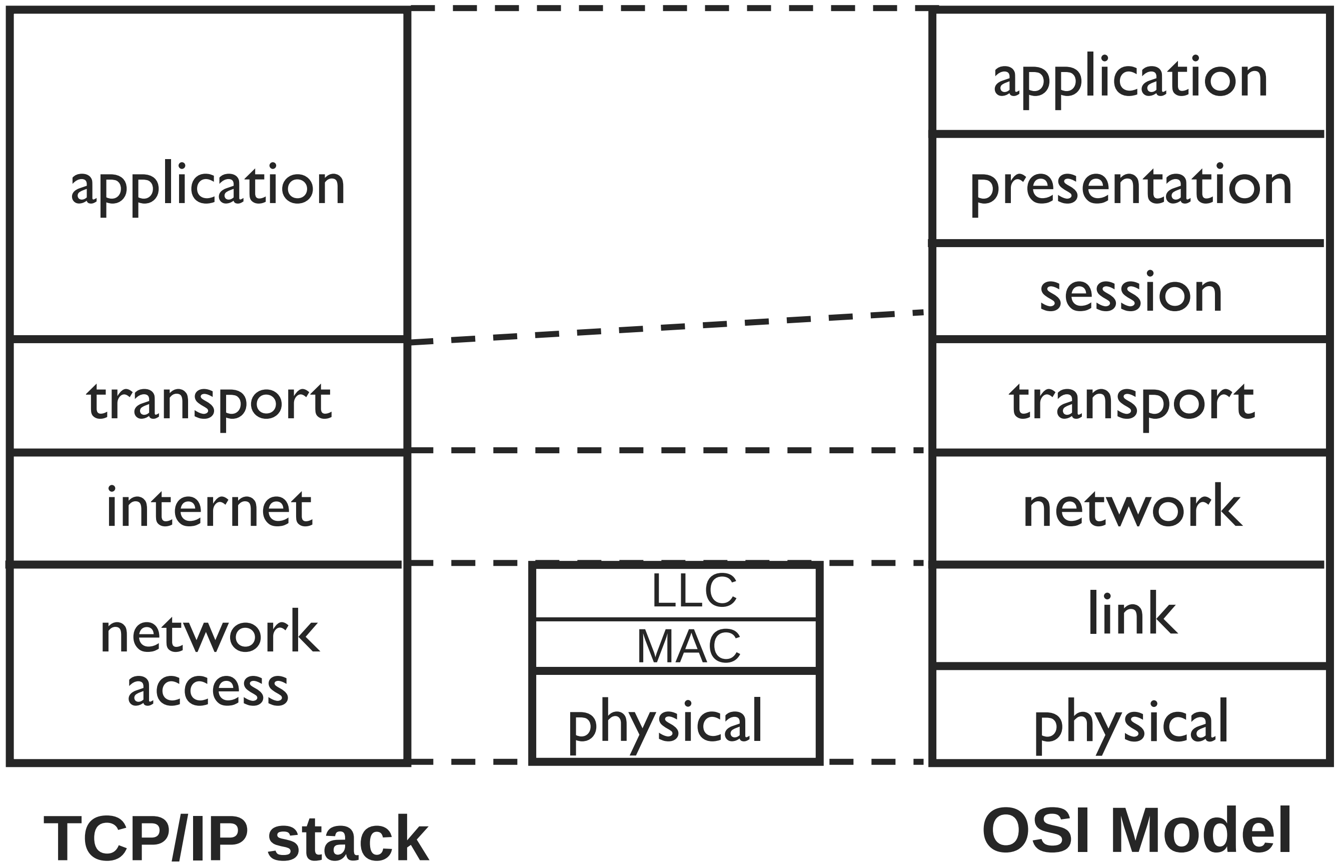
Institute of Telecommunications
Warsaw University of Technology
2017

internet technologies and standards

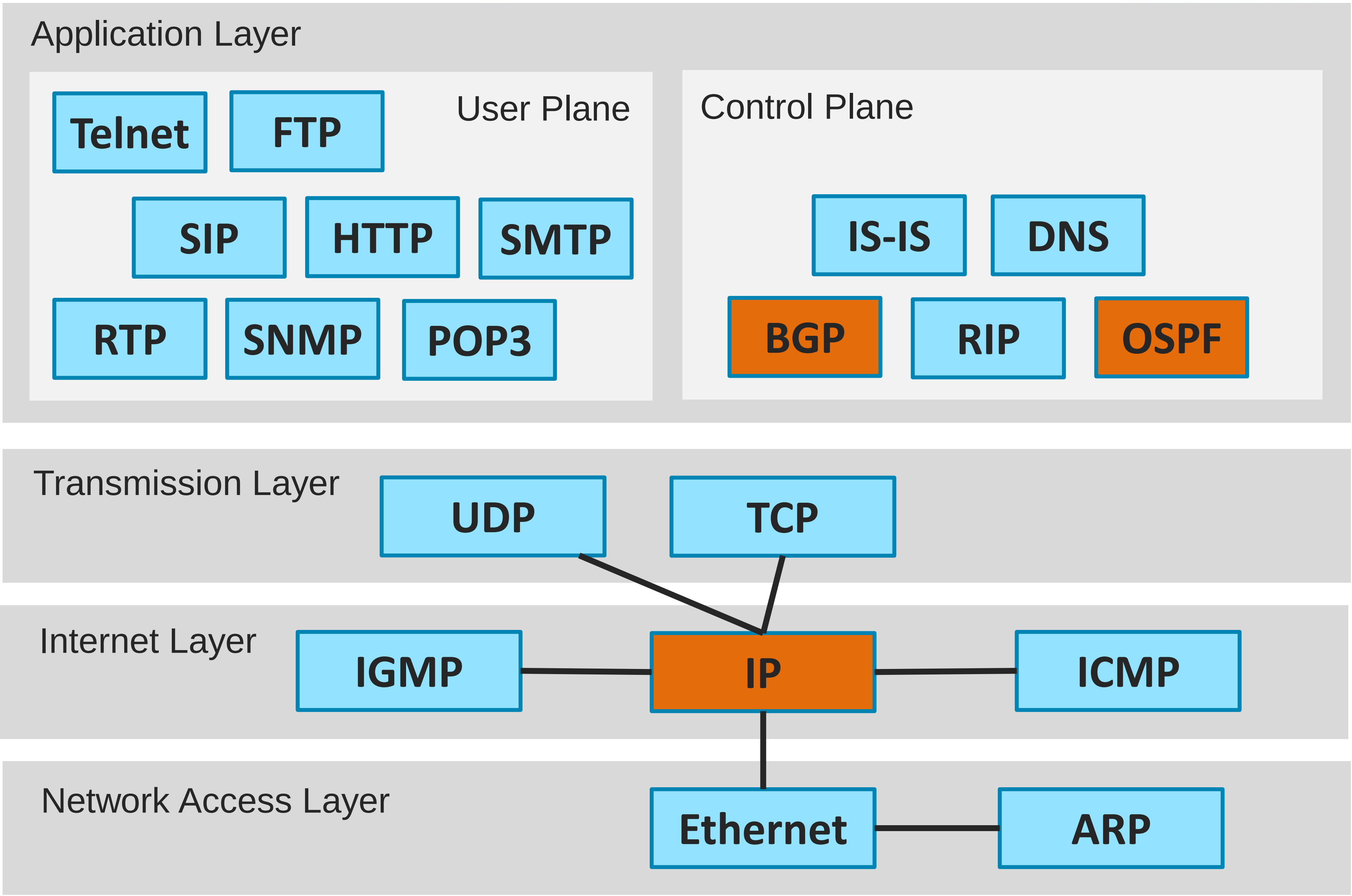
- Piotr Gajowniczek
- Andrzej Bąk

TCP/IP Stack: Network Layer

TCP/IP Stack vs OSI Reference Model



TCP/IP Protocols



IP Network Layer Services

- Encapsulation
  - *on sending side encapsulates segments into datagrams*
  - *on receiving side, delivers segments to transport layer*
- Packet switching/forwarding
  - *connectionless (datagram) packet forwarding*
    - each datagram carries destination address
    - router examines header fields in all IP datagrams passing through it
    - stateless forwarding
  - *packet transport from sending to receiving host*
  - *best effort service (default)*
- Packet fragmentation
- Addressing
  - *address formats*
  - *addressing rules etc.*
- QoS management (DiffServ/IntServ)

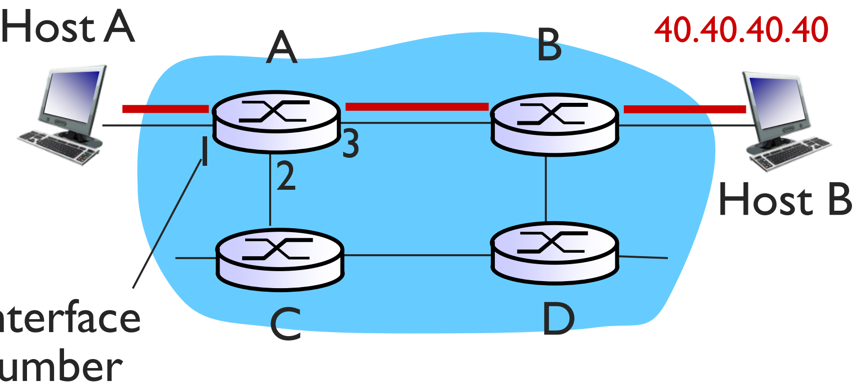
Forwarding vs Routing Function

- Forwarding: move packets from router's input to appropriate router output
- Routing: determine route taken by packets from source to dest.

analogy:

- forwarding: process of getting through single interchange
- routing: process of planning trip from source to dest.

Datagram Forwarding Table



Routing table in A:

Destination host address / Destination address range	Out interface	Next hop
10.10.10.10 / 10.10.10.0-255	3	Router B
20.20.20.20 / 20.20.20.0-255	1	Host A
30.30.30.30 / 30.30.30.0-255	2	Router C
40.40.40.40 / 40.40.40.0-255	3	Router B
…	…	

- 4 billion IP addresses, so very large potential routing table entries

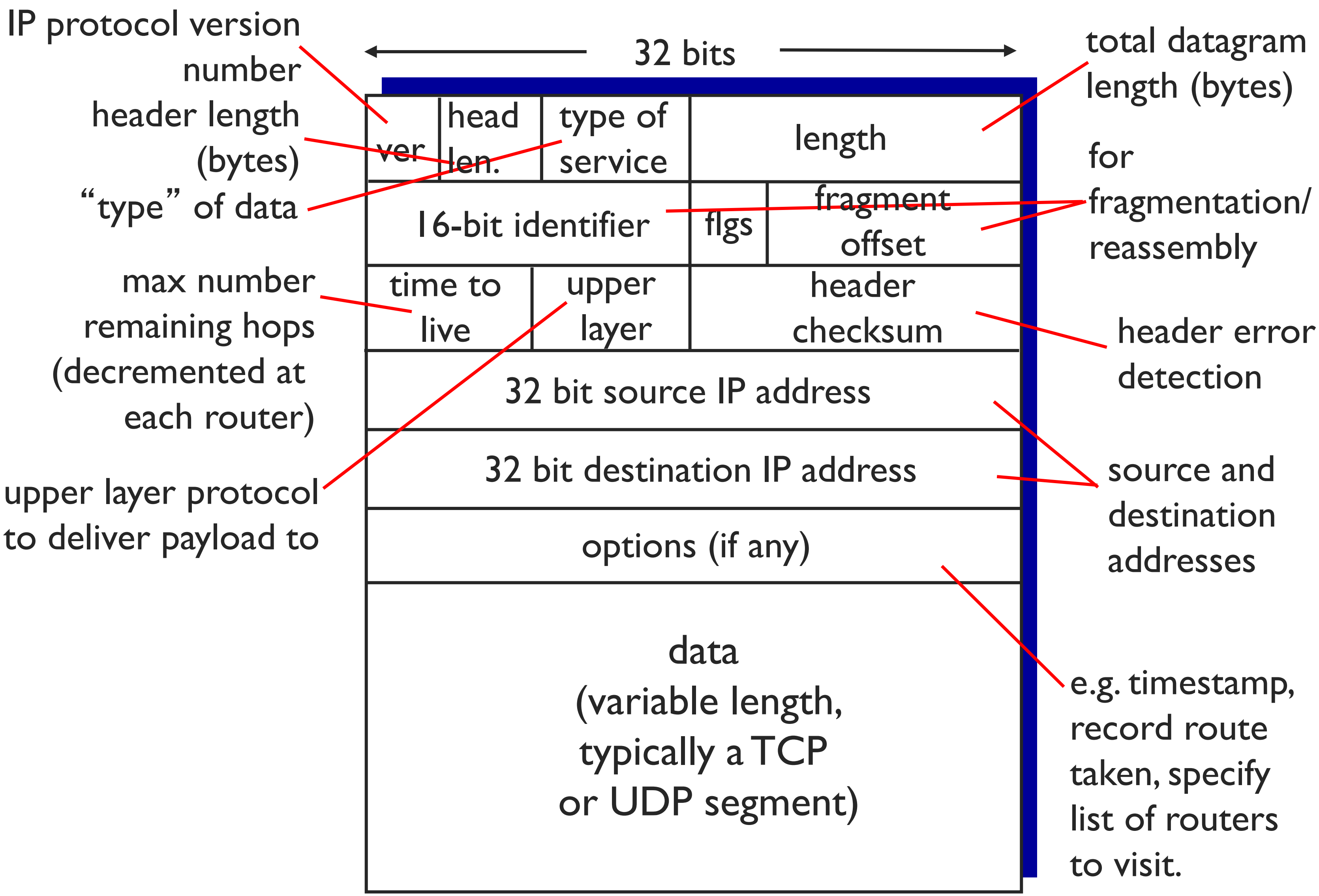
range of addresses (aggregate table entries)

Destination Address Range:

Prefix: 00101000 00101000 00101000 00000000

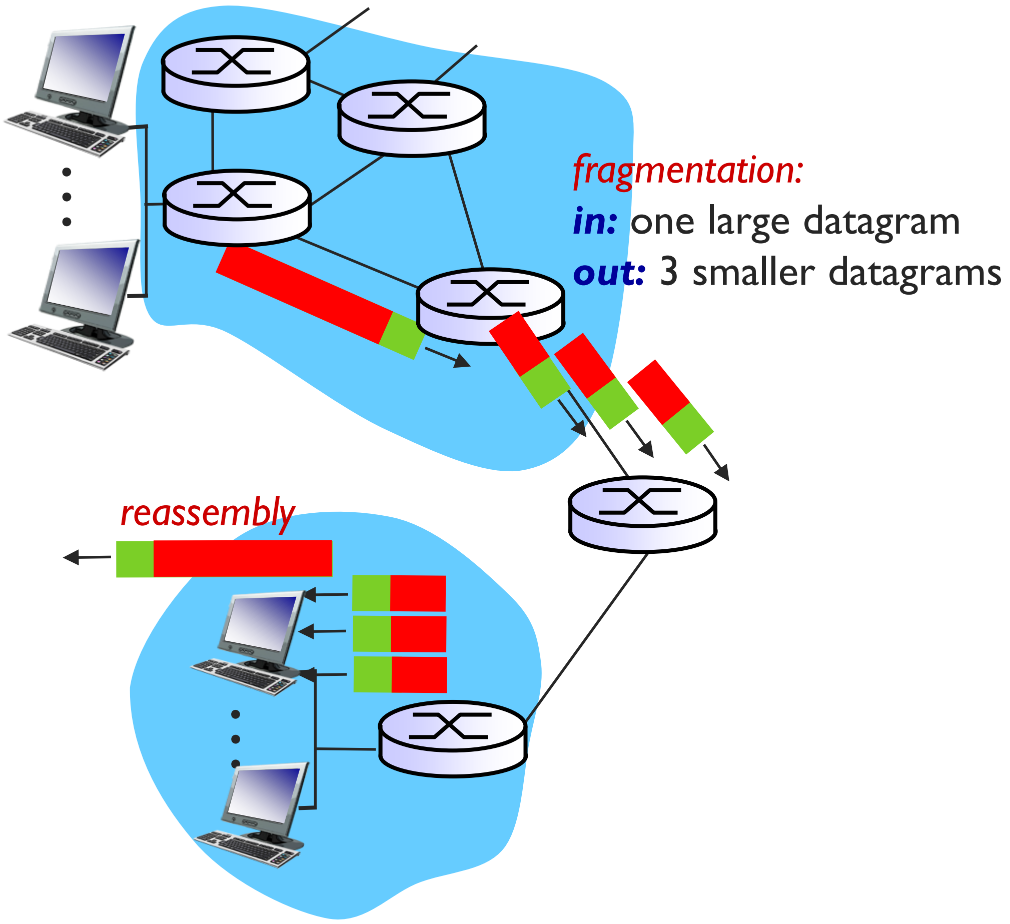
Mask: 11111111 11111111 11111111 00000000

IPv4 Datagram (Packet) Format



IP Fragmentation&Reassembly

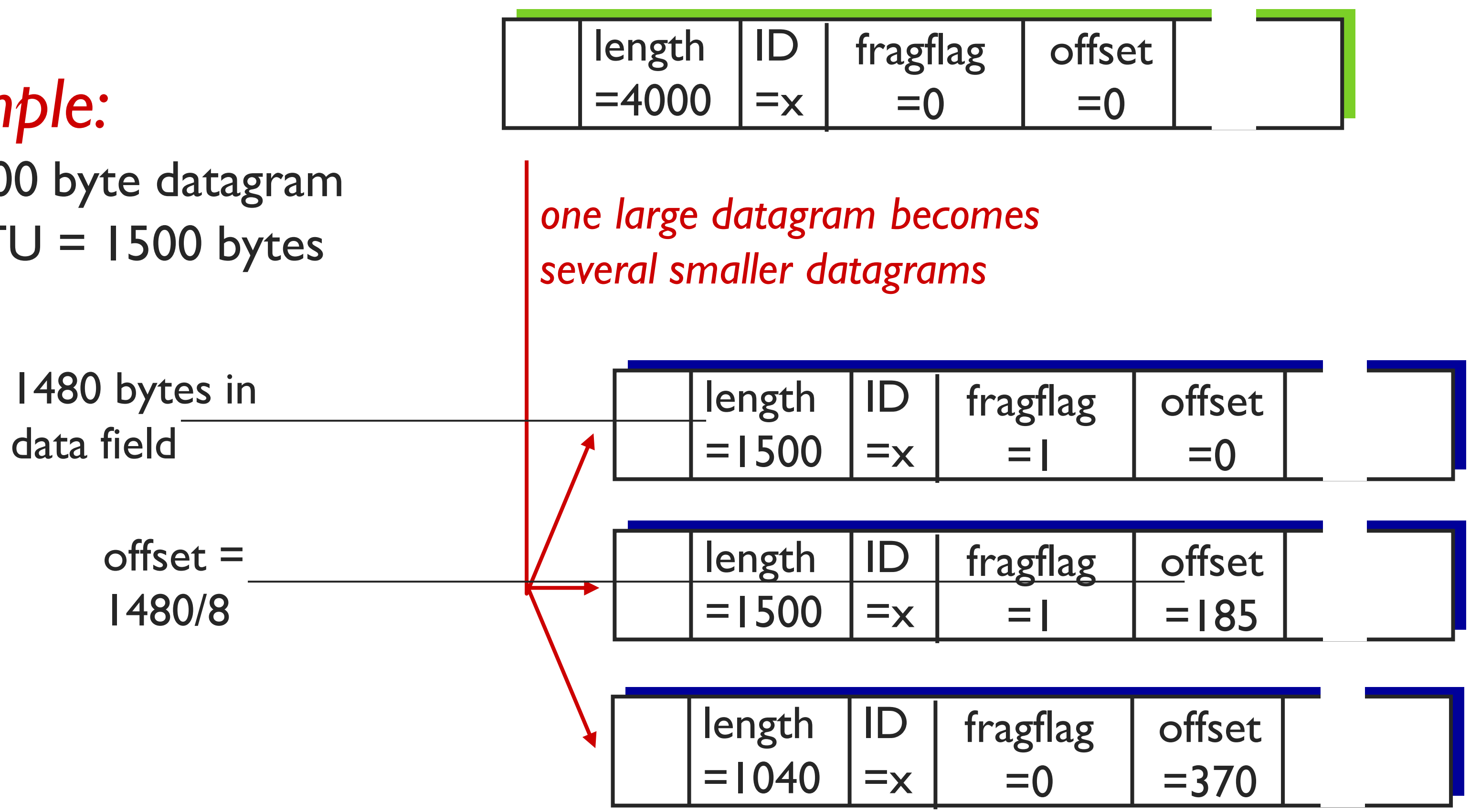
- network links have MTU - largest possible link-level frame
  - *different link types, different MTUs*
- large IP datagram divided ("fragmented") within net
  - *one datagram becomes several datagrams*
  - *"reassembled" only at final destination*
  - *IP header bits used to identify, order related fragments*



IP Fragmentation&Reassembly

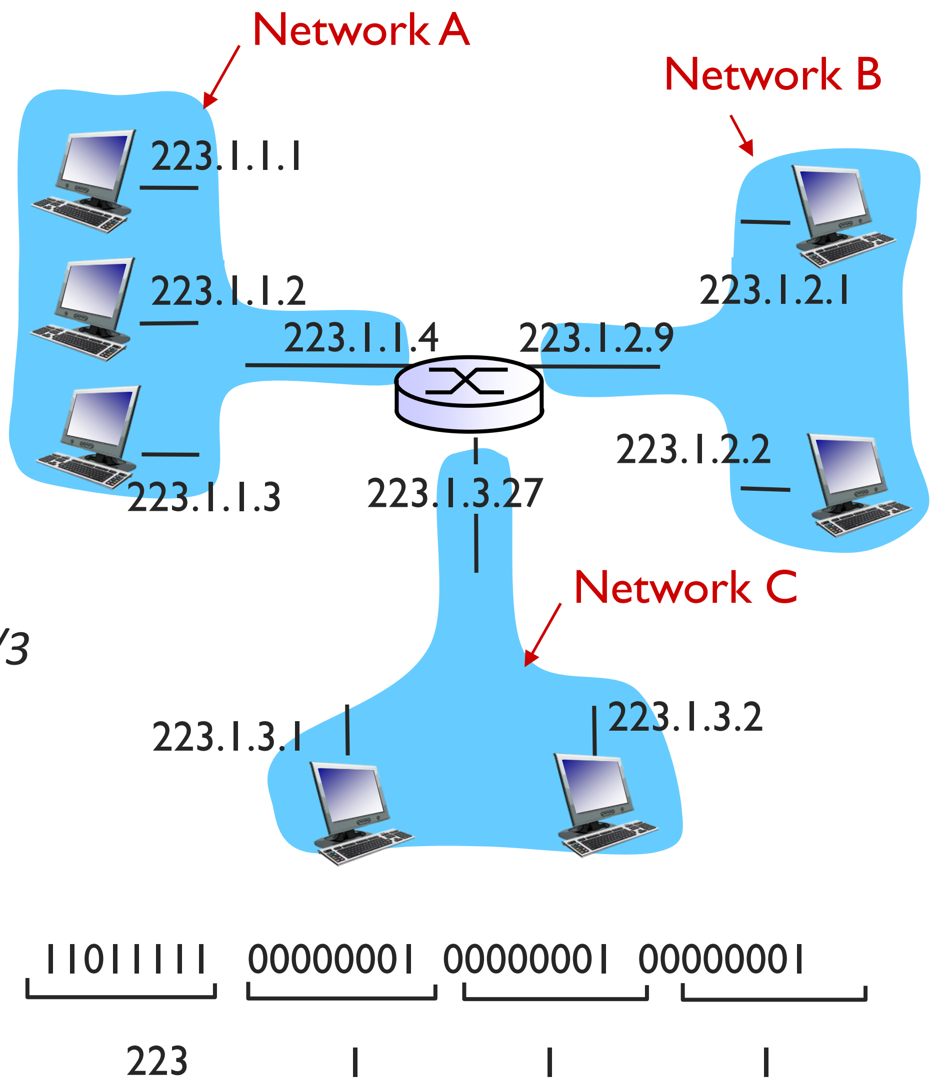
example:

- 4000 byte datagram
- MTU = 1500 bytes



IPv4 Addressing

- IP address – 32 bit identifier for host or router interface
- IP address structure:
  - *network part - high order bits*
  - *host part - low order bits*
- What's a network ?
  - *set of devices (interfaces) that can physically reach each other without intervening router*
- Initial IP spec introduced address classes
  - *class A (1-126) – 1 byte for network/3 bytes for hosts*
  - *class B (128-191) – 2 bytes for network/2 bytes for hosts*
  - *class C (192-223) – 3 bytes for network/1 byte for hosts*
  - *class D (224-239) – multicast*
  - *class E (240-255) – experimental*

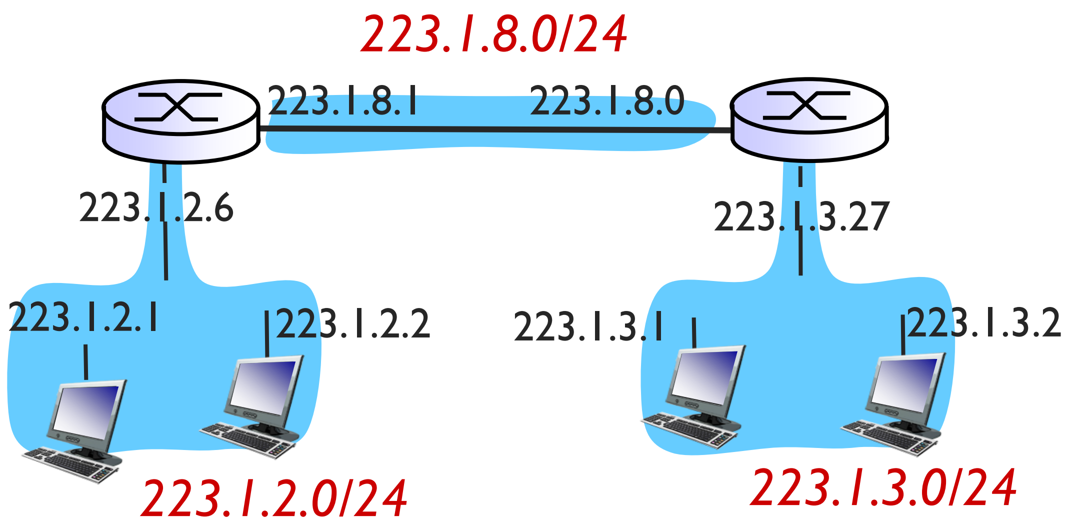


IPv4 Addressing

- Unicast address (routable except private and link local)
  - *first octet 1-126 or 128-223 (classes A+B+C without 0.0.0.0/8 i 127.0.0.0/8)*
  - *number of available addresses: 3 724 541 952*
  - *private addresses:*
    - 10.0.0.0/8
    - 172.16.0.0/12
    - 192.168.0.0/16
  - *link local (auto-generated for over the LAN communication)*
    - 169.254.0.0/16
- Multicast addresses
  - *first octet 224-239 (Class D)*
  - *number of available addresses: 268 435 456*
- Experimental addresses (not routable)
  - *first octet 240-255 (Class E)*
  - *number of available addresses : 268 435 456*
- 0.0.0.0/8
  - *hosts on "this" network – can be used as a source address in initialization procedure by which the host learns its full IP address*
- 127.0.0.0/8
  - *loopback addresses*

Subneting/VLSM

- How many IP networks?
  - *Need one IP network per layer 2 network (including point-to-point links)*
- Inefficient use of IP address space
  - *Class A, B, C networks usually bigger than physical network size*
- Solution: subneting
  - *Divide IP network into smaller pieces (subnets)*
    - Add mask that defines the network and host numbers
  - *More efficient use of IP address space*
  - *VLSM (Variable Length Subnet Masking) – allows to subent IP network to subnets of diffrent size*



Classless InterDomain Routing

- Classless InterDomain Routing (CIDR)
  - *Network portion of address of arbitrary length*
    - address mask is associated with address prefix (and propagated by the routing protocols)
  - *Allows for address aggregation (supernetting)*
  - *Address format: a.b.c.d/x, where x is # bits in subnet portion of address*

subnet part | host part

11001000 00010111 00010000 00000000

200.23.16.0/23

Forwarding with CIDR: Longest Prefix Matching

longest prefix matching

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

examples:

DA: 11001000 00010111 00010110 10100001 — which interface?

DA: 11001000 00010111 00011000 10101010 — which interface?

IP addresses: how to get one?

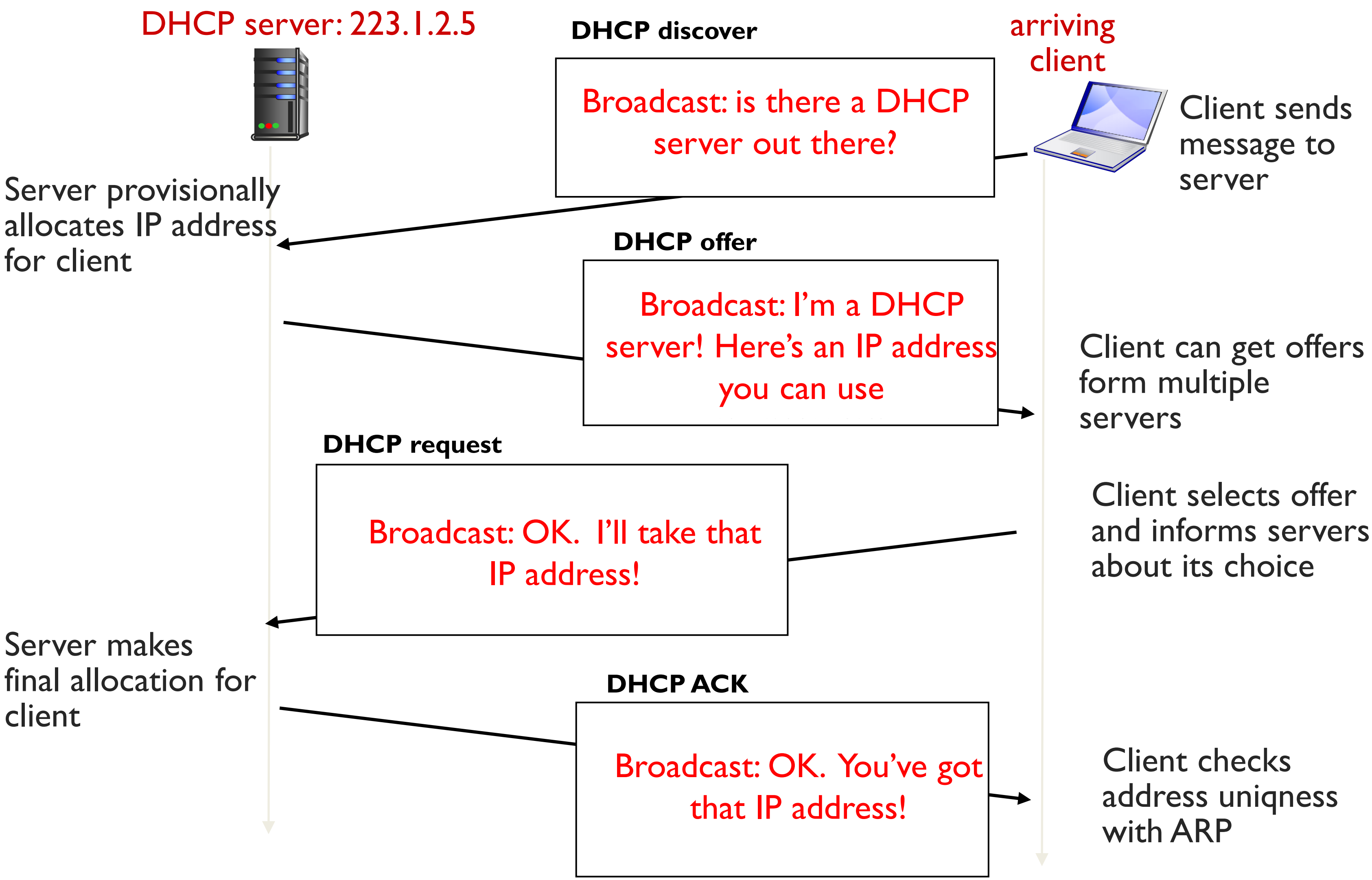
- Q: How does a host get IP address?

- Hard-coded by system admin in a file
  - *Windows: control-panel->network->configuration->tcp/ip->properties*
  - *UNIX: /etc/rc.config*
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server
  - *"plug-and-play"*

DHCP: Dynamic Host Configuration Protocol

- Goal: allow host to dynamically obtain its IP address from network server when it joins network
  - *simplifies network admistration tasks*
  - *allows reuse of addresses (only hold address while connected/"on")*
  - *support for mobile users who want to join network*
- DHCP overview:
  - *host broadcasts "DHCP discover" msg*
  - *DHCP server responds with "DHCP offer" msg*
  - *host requests IP address: "DHCP request" msg*
  - *DHCP server sends address: "DHCP ack" msg*

DHCP Client-Server Scenario



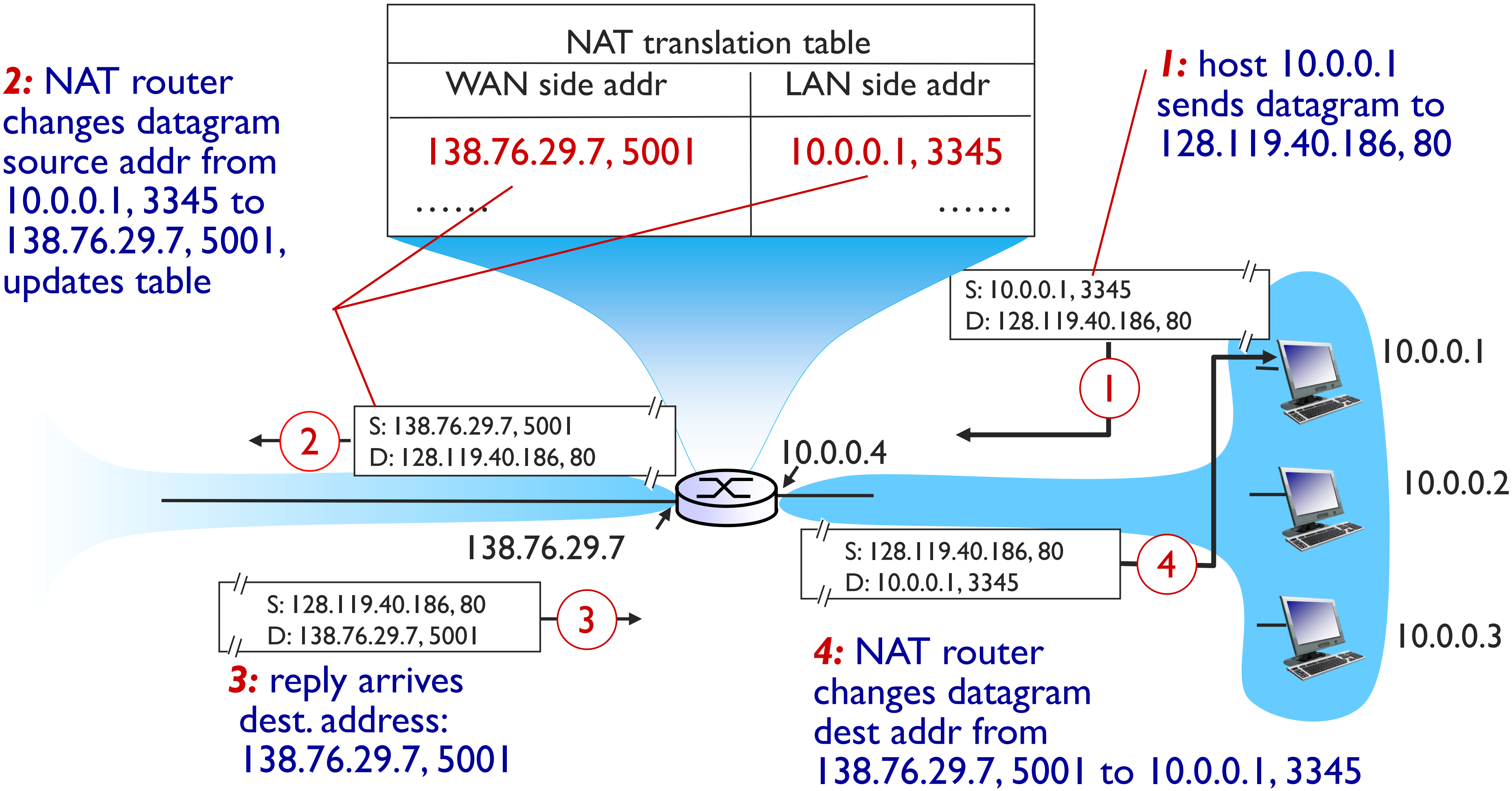
DHCP: more than IP addresses

- DHCP can return more than just allocated IP address on subnet:
  - *address of first-hop router for client (gateway)*
  - *name and IP address of DNS sever*
  - *network mask (indicating network versus host portion of address)*
  - *lease time*
  - *..... more*

NAPT: network address port translation

- NAT function
    - Maps private addresses and port numbers to single public address and different port numbers
    - Allows host in private network to access public Internet
- Motivation for NAT
    - *Solves the problem of IP address shortage*
        - better usage of IP address space
        - range of addresses not needed from ISP: just one IP address for all devices
    - *simplified administration of private network*
        - can change addresses of devices in local network without notifying outside world
        - can change ISP without changing addresses of devices in local network
    - *network security*
        - devices inside local net not explicitly addressable, visible by outside world

NAPT: network address port translation



NAPT: network address port translation

- 16-bit port-number field:
  - *65,000 simultaneous connections with a single LAN-side 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 app designers, e.g., P2P applications
  - *address shortage should instead be solved by IPv6*

Why Do We Need IPv6?

- The main motivation is shortage of 32-bit address space
  - *IPv4 address are by now completely allocated*
- Uneven distribution of the IPv4 addresses around the World (60% being allocated to US)
- Early distribution methods allocated addresses inefficiently
  - *Some organizations get address blocks much larger than they needed, the reallocation of these addresses is practically not possible*
  - *Addresses allocated not hierarchically – large routing tables*
- Emergence of IoT technology creates new demand for IP addresses
  - *Potentially billions of new devices that must be connected to the network*
- Additional motivation:
  - *Improve protocol efficiency (packet forwarding)*
  - *Restoration of the end-to-end network model (elimination of NAT)*

New Features in IPv6

- Extended address space
  - *Can handle all imaginary needs for IP addresses*
  - *Support for hierarchical structure of IP address space to optimize global routing*
- Stateless auto-configuration
  - *Hosts can auto generate IP addresses from the network prefix – simplifies the management of IP devices*
  - *Support for device mobility*
- New simplified header
  - *Header format helps speedup processing/forwarding*
- Improved support for QoS and extensions
  - *Header changes to facilitate QoS*
  - *Header changes to support protocol extensions (more open protocol)*

IPv6 vs. IPv4 Header

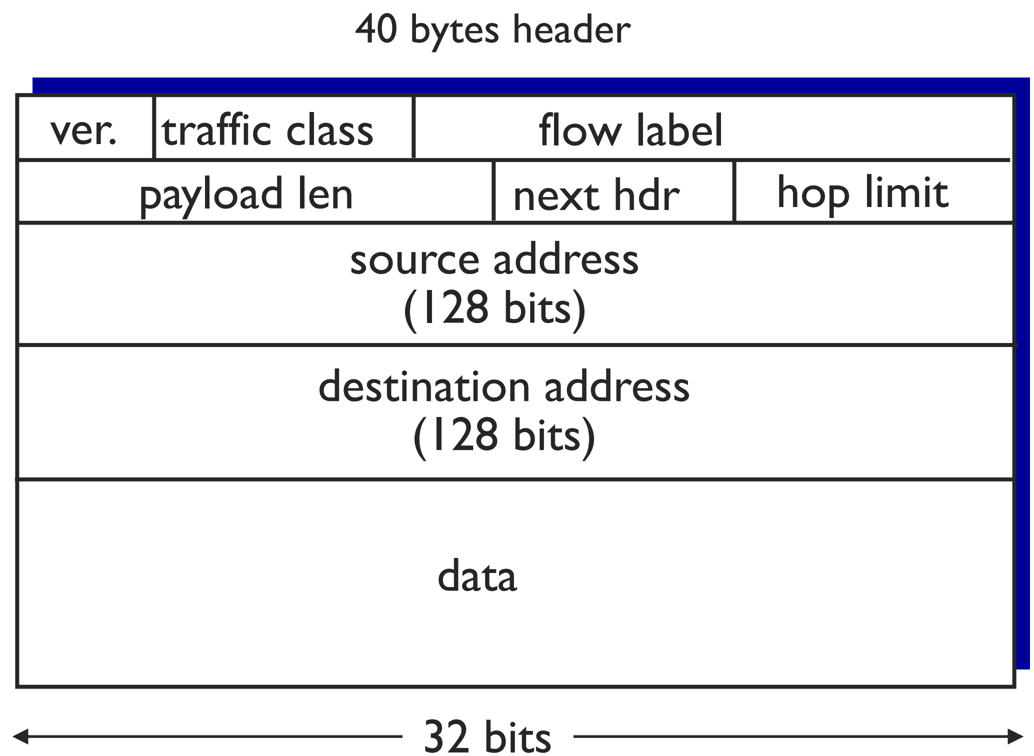
- Removed fields
  - *Header length*
    - the header is now fixed length
  - *Identification, Flags, Fragment Offset*
    - were used to implement fragmentation inside the network
    - routers now do not support fragmentation
    - the fragmentation is handled by hosts via extension header
  - *Header Checksum*
    - was eliminated to improve packet processing time
    - the checksumming is commonly done at layer 2 or 4 (no need for this feature on the network layer)

IPv6 vs. IPv4 Header

- New field
  - *Flow label*
    - identification of traffic flow for QoS purpose
- Common fields
  - *Hop limit*
    - used basically in the same way as TTL in IPv4
  - *Traffic Class*
    - similar to Type of Service/DS Field in IPv4
  - *Version*
    - the same in both protocols
  - *Next header*
    - carries upper layer protocol type or extension type
    - Similar to Protocol Type in IPv4

IPv6 datagram format

- version: IPv6 (bits 0110)
- traffic class: 6-bits Differentiated services (classify packets). 2-bits for ECN: congestion control.
- payload length: length of the data field
- next header: identify type of the extension header (if any)
- hop limit: as TTL in IPv4
- source address: 128 bit IPv6 address
- destination address: 128 bit IPv6 address
- flow Label: identify datagrams in same "flow". Real time services – all packet of that flow should stay on the same path



IPv6 address notation

- IPv6 addreses are presented as 8 hexadecimal blocks seperated by colons:

  0dfc:0000:0000:0000:0217:cbff:fe8c:0000

- Simplified IPv6 address notation:
  - omitting leading zeros within blocks

    dfc:0:0:0:217:cbff:fe8c:0

  - double colons (::) in place of a series of zeros

    dfc::217:cbff:fe8c:0

  - only one double colons allowed:

    ~~dfc::0217:cbff:fe8c::~~

    - Why? The double use of :: makes it unclear how many zeros were in each 0 string originally.

- representing IPv4 address as IPv6 address

  0:0:0:0:0:ffff:192.1.56.10

Extension Headers

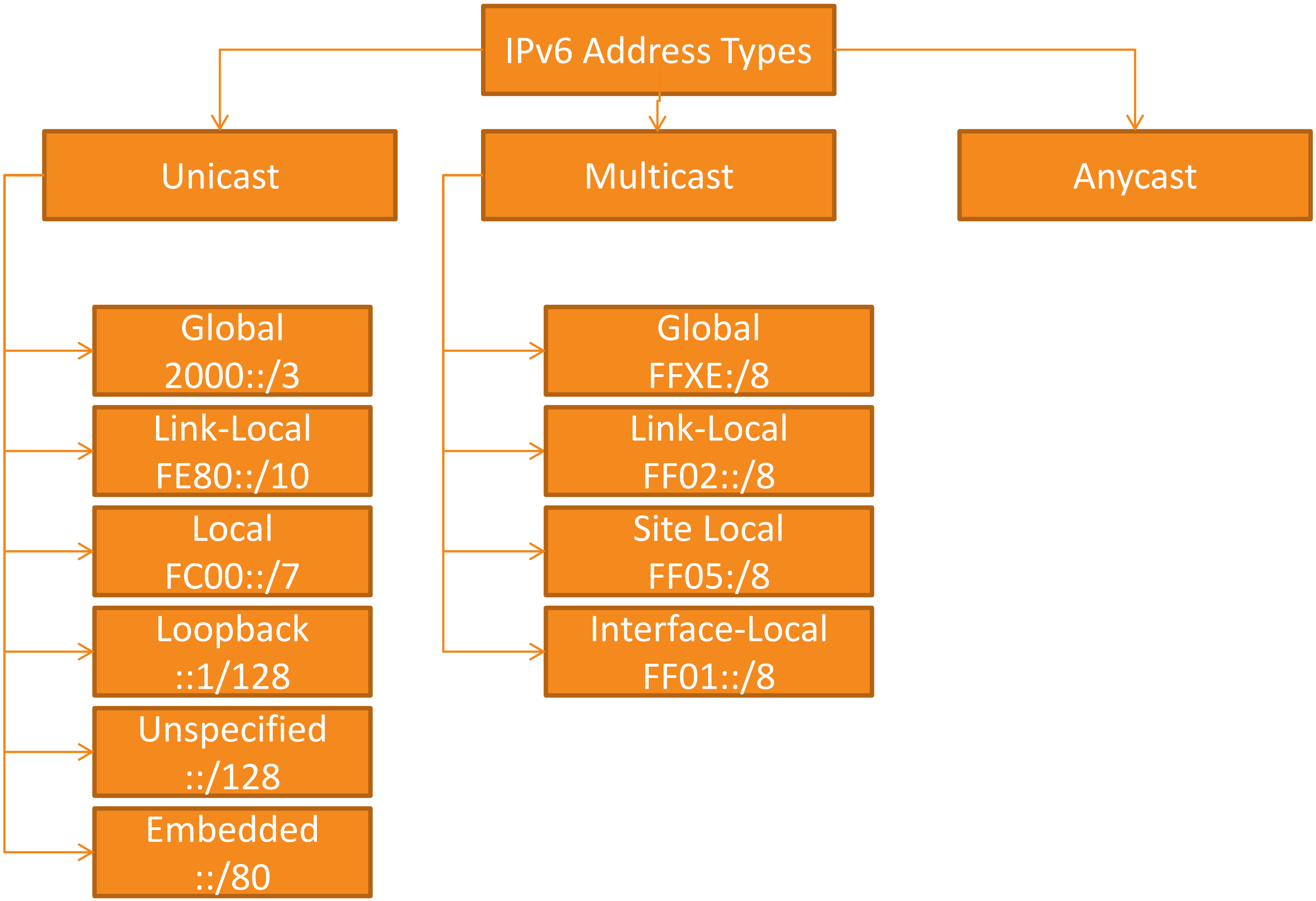
- The IPv4 header can be extended from 20 to 60 bytes in order to specify options like Security Option, Source Routing or Timestamp.
  - *The options in IPv4 are practically not used because the IPv4 hardware have to pass packets containing options to the main processor (software processing)*
- The IPv6 provides options via extension headers.
  - *The extension headers are placed between the IP header and the upper layer protocol header*
  - *There can be zero, one or more extension headers*
  - *The extension headers are processed by the node indicated in the destination address field*
    - Intermediate nodes can switch packets in hardware without need to process options
    - One exception to this rule is Hop-by-Hop Options header that is processed by all nodes

Extension Headers

IPv6 Header Next Header = 0 (hop-by-hop)	Hop-by-hop Extension Next Header = 44 (Fragment)	Fragment Extension Next Header = 6 (TCP)	TCP Header + data

- The type of the first extension header is indicated by the Next Header field in IP header, the extension headers contains Next Header field that identifies consecutive headers etc.
- Extension headers are placed in a packet in the fixed order
    - *Hop-by-Hop Options (Next Header=0)*
    - *Destination Options (to be processed by destinations indicated in the Routing header) (Next Header= 60)*
    - *Routing (Next Header= 43)*
    - *Fragment (Next Header= 44)*
    - *Authentication (Next Header= 51)*
    - *Encapsulating Security Payload (Next Header= 50)*
    - *Destination Options (to be processed by final destination) (Next Header= 60)*

IPv6 Address Types



Global Unicast Address

Global routing prefix n bits	Subnet ID 64-n bits	Interface ID 64 bits
Identifies site (address range allocated to the site)	Identifies link with site	Identifies interface on the link

- The global unicast address is identified by binary prefix 001 (2000::/3)
- The global routing prefix identifies the address range allocated to a site
  - *The prefixes are allocated by international registry service*
  - *Subnet ID is allocated by site administrator for each link within site*
- EUI-64 format assumes 64 bit for Interface ID
- Recommended prefix length is 48 bits
  - *… leaving 16 bit Subnet ID – over 65k subnets per site*
  - *The Interface ID must be unique on each link (can be determined by autoconfiguration)*

Link-Local Address

FE80::/10	0	Interface ID
10 bits	54 bits	64 bits

Identifies address as link-local

Identifies interface on the link

- The link-local address is identified by binary prefix 1111 1110 10 (FE80::/10)
- The link-local addresses are used within single link and are not routable
  - *The link-local addresses are generated by auto-configuration*
  - *The link-local addresses means „the host on this link”*
  - *The link-local addresses are used for auto-configuration of global address, neighbour/router discovery, for communication over LAN networks*

Local IPv6 Address

FC00::/7 8 bits	Global ID 40 bits	Subnet ID 16 bits	Interface ID 64 bits
Identifies address as local	Makes the local address globally unique. Used for address filtering on network boundary	Identifies link with site	Identifies interface on the link

- The local address is identified by binary prefix 1111 110 (FC00::/7)
    - *1111 1101 (FD00::/8) – address administrated locally*
    - *1111 1100 (FC00::/8) – reserved for future use*
- The local addresses are globally unique but they should never be routed in the Internet (uniqueness assures routing security in case of misconfiguration)
- The local addresses plays the role of private address to be used by the site administrator for communication over private networks

Special IPv6 Addresses

- Unspecified address (allzero address) - ::/128
  - *Indicates the lack of IP address, can be used as source address during the initial host configuration*
- Loopback address - ::1/128
  - *Used as destination address in order to send packet internally within the host (from one process to the other). Its meaning is „this host”.*

Special IPv6 Addresses

- IPv4 Embedded Addresses
  - *IPv4 compatybile IPv6 address (depreciated): for tunneling IPv6 packets over IPv4 infrastructure*

0 80 bits	0 16 bits	IPv4 address 32 bits

  - *IPv4 mapped IPv6 address: represents IPv4 node as IPv6 address*

0 80 bits	FFFF 16 bits	IPv4 address 32 bits

Multicast Address

FF	Flags	Scope	Group ID
8 bits	4 bits	4 bits	112 bits

Identifies address as multicast

Identifies the address scope

Identifies a group of interfaces

- The multicast address is identified by binary prefix 1111 1111 (FF00::/8)
- The multicast address can be assigned to a group of interfaces, all group members will receive the packets send to the multicast address
- Flags: 00PT
  - *P=1: Multicast address based on network prefix (RFC3306)*
  - *T=0: Well-known multicast address (permanently assigned), T=1: temporary multicast address*
- Scope: used to limit the range of the multicast transmission
  - *1: interface (node) local scope (similar as loopback in unicast transmission)*
  - *2: link local scope*
  - *5: site local scope*
  - *E: global scope*

Well-known Multicast addresses

- Interface (node)-local scope
  - *FF01::1 - All-nodes address*
  - *FF01::2 - All-routers address*
- Link-local scope
  - *FF02::1 - All-nodes address*
  - *FF02::2 - All-routers address*
  - *FF02::5 - All-OSPF routers address*
  - *FF02::6 - All-OSPF DR routers address*
  - *FF02::1:2 - All DHCP servers address*
  - *FF02::1:FFXX:XXX - Solicited-node address*
- Site-local scope
  - *FF05::2 - All-routers address*
  - *FF05::1:3 - All DHCP servers address*

Solicited-Node Multicast Address

FF	0	2	0:0:0:0:1:FF	XX:XXXX
8 bits	4 bits	4 bits	88 bits	24 bits

24 bit taken from unicast/anycast address

- The solicited-node multicast address is identified by prefix FF02:0:0:0:0:1:FF00::/104)
  - *The solicited-node multicast address is formed by appending last 24 bits of IPv6 unicast/anycast address to the solicited-node multicast prefix*
- The node must join every solicited-node multicast address generated for every unicast /anycast address assigned to the node
  - *The packets send to the solicited-node multicast address are received only by this node (not all nodes on the link)*
- It is used in Neighbour Discovery, ARP procedure

Dynamic Allocation of Multicast Addresses (RFC3306)

FF	Flags	Scope	Reserved	Plen	Network prefix	Group ID
8 bits	4 bits	4 bits	8 bits	8 bits	64 bits	32 bits

Identifies address as multicast

Identifies the address scope

Identifies a group of interfaces

- Extended multicast address format: FF30::/12 – any-source multicast (ASM)
  - *Flags: P=1 and T=1*
  - *Reserved – not used, set to 0*
  - *Plen – the length of the network prefix*
  - *Network prefix (max 64 bit, padded with zeros)*
    - Identifies the domain that allocated this address
    - Makes the address globally unique
  - *32 bit group ID*
    - maps to the MAC address
- Source-specific multicast address (SSM): FF3x::/96
  - *Flags: P=1 and T=1, Scope =x, Plen =0, Network prefix = 0*
  - *Source IP address identifies the owner of the multicast address*

ICMPv6

- Significantly extended comparing to ICMPv4
  - *New control messages*
  - *Multicast listener discovery (replaces IGMP)*
  - *Address resolution protocol (replaces ARP)*
  - *Neighbour discovery protocol*
  - *Path MTU discovery*
  - *…*
- ICMPv6 introduces two message classes
  - *ICMP error messages*
  - *ICMP information messages*

ICMPv6 Message Format

Type 1 Byte	Code 1 Byte	Checksum 2 Bytes	Message Body Variable

- Type – determines the message class and format
  - *0-127 – error messages*
  - *128-255 – information messages*
- Code – depends on the Type and provides additional information about the message
- Checksum – calculated for ICMP header and parts of the IPv6 header
- Message body – content depends on the message Type and Code

ICMPv6 Error Messages

Type	Usage
1=Destination unreachable	Packet cannot be delivered because the destination does not exists or cannot be reached due to some administrative rules
2=Packet too big	The packet length exceedes the link MTU
3=Time exceeded	The hop limit was eceeded or the fragmented packet was not feasembled in assumed time
4=Parameter problem	The packet header contains unknown fields e.g. in packet or extension header

Destination Unreachable Message

- The packet due to some reason cannot be delivered to the destination
  - *The ICMP message is send to the source address of the invoking packet*
- The code field contains the reason why the packet could not be delivered
  - *0 - no route to destination*
    - router has no entry in the routing table for the destination address
  - *1 - communication with dest. administratively prohibited*
    - this type of message can for example be sent by firewall that is configured to filter out the packet
  - *2 – beyond scope of source address*
    - the scope of source and destination address is not the same e.g. the destination address is global while the source address is link-local
  - *3 - address unreachable*
    - Cannot resolve layer 2 address
  - *4 - port unreachable*
    - No listener for given port number at target host
  - *5 - Source address failed ingress/egress policy*
    - the packet cannot be delivered due to the ingress and egress policy, packet was filtered out by the source or destination host
  - *6 - reject route to destination*
    - the route to destination of type reject

ICMPv6 Information Messages

Type	
128=Echo request	RFC4443 Ping command
129=Echo replay	
130=Multicast listener query	RFC2710 MLD – Multicast Listener Discovery
131=Multicast listener reoport	
132=Multicast listener done	
133=Router solicitation	RFC2461 NDP – Neighbor Discovery Protocol
134=Router advertisement	
135=Neighbour solicitation	
136=Neighbour advertisement	
137=Redirect	
138=Router renumbering	RFC2894
…	

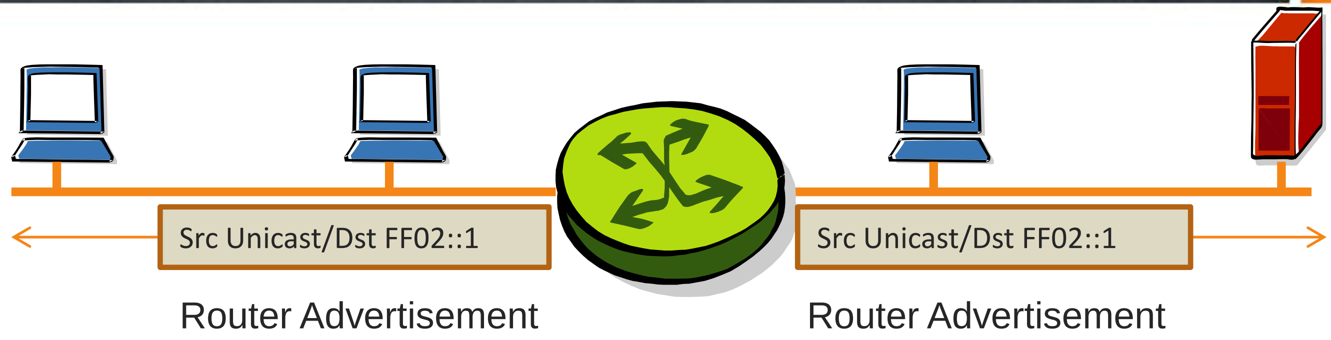
NDP Messages

Name	Meaning	Usage
Neighbour Solicitation	Who has IP address X?	For *Duplicated Address Detection* (DAD), *Address Resolution,* and *Neighbour Unreachability Detection* (NUD)
Neighbour Advertisement	I have it! (+ MAC address)	Response to Neighbour Solicitation
Router Solicitation	What's my prefix?	For global-address auto-configuration
Router Advertisement	The prefix is Y	Response to Router Solicitation
Redirect Message	There is a better route to Z	For finding router that can forward the packet

Neighbor Discovery Protocol (NDP)

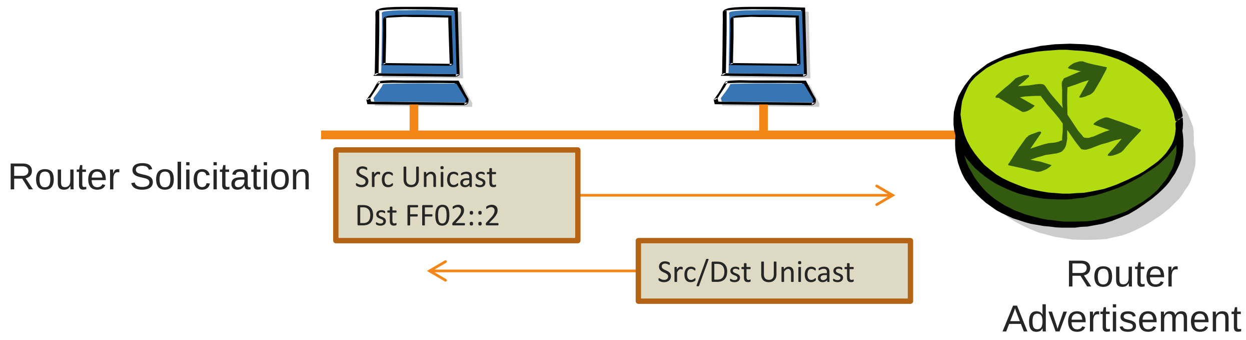
- The Neighbor Discovery Protocol provides the following functionality:
  - *Neighbour Discovery*
  - *Router Discovery*
  - *Duplicate IP Address Detection (DAD)*
  - *Neighbour Unreachability Detection (NUD)*
  - *Address Resolution Protocol (ARP)*
  - *Auto-configuration of IPv6 addresses*
  - *Redirection*

Router Discovery



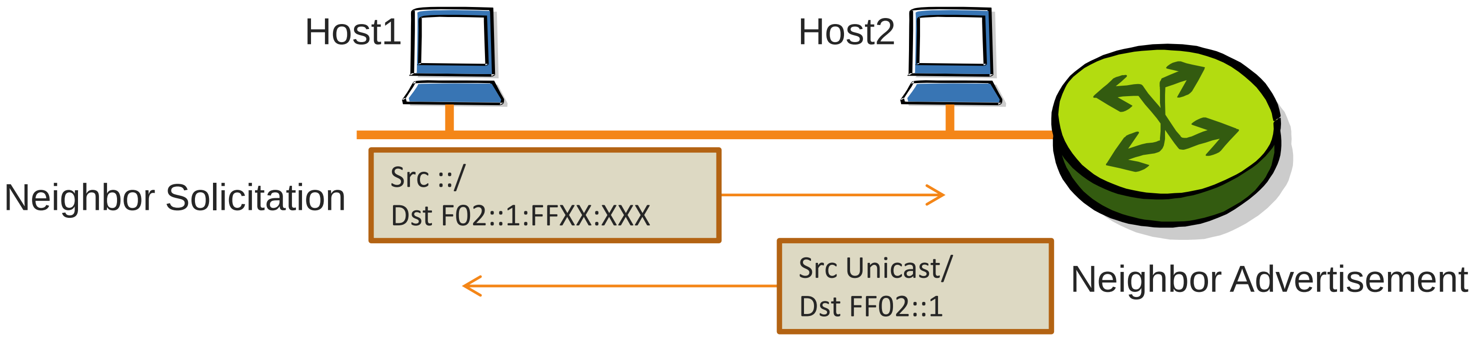
- Router sends out Router Advertisement messages at regular intervals
  - *The message is send to all nodes multicast address: FF02::1*
  - *Hop limit is always set to 255 (packets with lower hop limits are ignored)*
  - *The message contains the following parameters*
    - Default hop limit – used to configure the default hop count for all nodes on the link
    - Flags
      - M - statefull configuration for IP prefix (DHCP)
      - O – statefull configuration for other parameters then IP prefix
      - H = home address flag
    - Router lifetime – specifies the amount of time the router is used as default router (zero otherwise)
    - Reachable time - specifies the amount of time the router is reachable
    - Retrans time - specifies time between neighbor solicitation messages, used in NUD and ARP
    - Options: link-layer address, MTU size, prefix information

Router Discovery



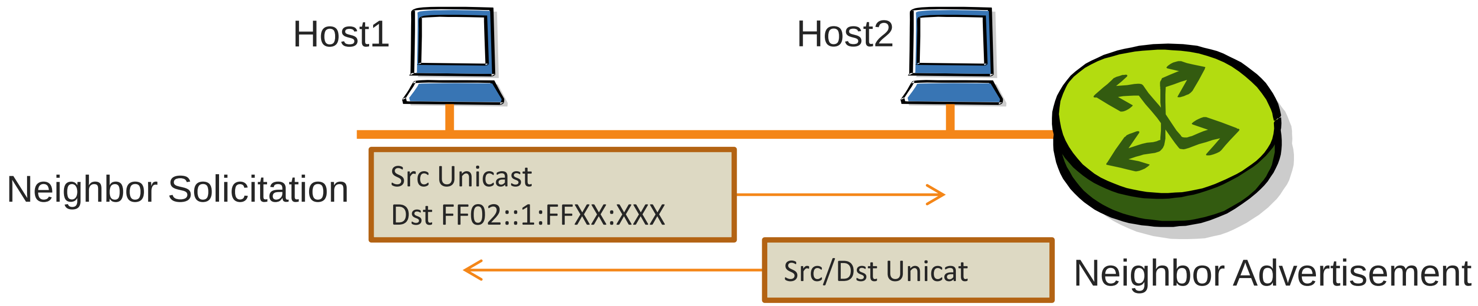
- Host can request Router Advertisement message (outside regular intervals) by sending Router Solicitation message
    - *The message is send to all routers multicast address: FF02::2*
    - *Hop limit is always set to 255 (packets with lower hop limits are ignored)*
    - *The message contains the link-layer address of the host (only in case the IP address is known to the host)*
- The Router Advertisement message is send back to the host (to source address of the Solicitation message)

Duplicated Address Detection (DAD)



- DAD is usually used during stateless auto-configuration to verify if the IP address is in use
- The host sends out Neighbor Solicitation message
  - *The message is send to solicited-node multicast address with the unspecified source address*
  - *The address being verified is send in target address field in the solicitation message*
- If the address is in use the host identified by the node solicitation multicast address responds with Neighbor Advertisement message
  - *The response is send to allnode multicast address*

Link Layer Address Resolution (ARP)



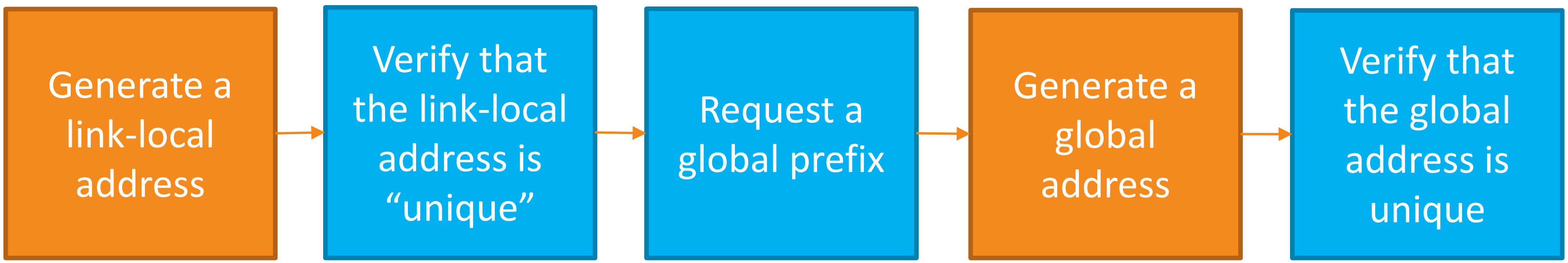
- The host that wants to resolve the link layer address sends out Neighbor Solicitation message
  - *The message is send to solicited-node multicast address: FF02::1:FFXX:XXX with the source address of the sending host*
  - *The message contains link-layer address of sending host*
- The target host (identified by the node solicitation multicast address) responds with Neighbor Advertisement message
  - *The message contains link-layer address of responding host*

IPv6 address assignment

- **Full 128-bits assignment (stateful)**
  - *Manually or,*
  - *Dynamically by DHCPv6*

- **Stateless autoconfiguration (SLAAC)**
  - *EUI-64 address format (Prefix + interface id)*
    - First 64-bits for a network prefix and
    - Last 64-bits for a Interface Id
  - *Network prefix can be assigned:*
    - Manually or,
    - Dynamically by NDP/ICMPv6
  - *An Interface Id is automatically formed using the MAC physical address*

SLAAC (Autoconfiguration) Procedure



SLAAC (GLOBAL Unicast)

(network prefix)

We are given a 64-bits network prefix - say 2000::/64

2000	0000	0000	0000				

and a 48-bits Ethernet physical address - say 01-23-45-67-89-AB

(MAC address)

01	23	45	67	89	AB

We insert FF-FE in the middle

(EUI-64)
Extended Unique Identifier

01	23	45	FF	FE	67	89	AB

Set to 1 (or flip) the 7th bit of the first byte

(Interface Id)

03	23	45	FF	FE	67	89	AB

Join the 64-bits prefix with the 64-bits interface id

2000	0000	0000	0000	0323	45FF	FE67	89AB

Which can be rewritten as 2000::0323:45FF:FE67:89AB

SLAAC (Link Local)

Link-local prefix (10-bits)	Zero (54-bits)			Interface Id (64-bits)			
FE80	0000	0000	0000	0323	45FF	FE67	89AB

OR

Link-local prefix (10-bits)	Zero (54-bits)			Randomly generated interface Id (64-bits)			
FE80	0000	0000	0000	0245	1527	3822	AB43

SLAAC Procedure