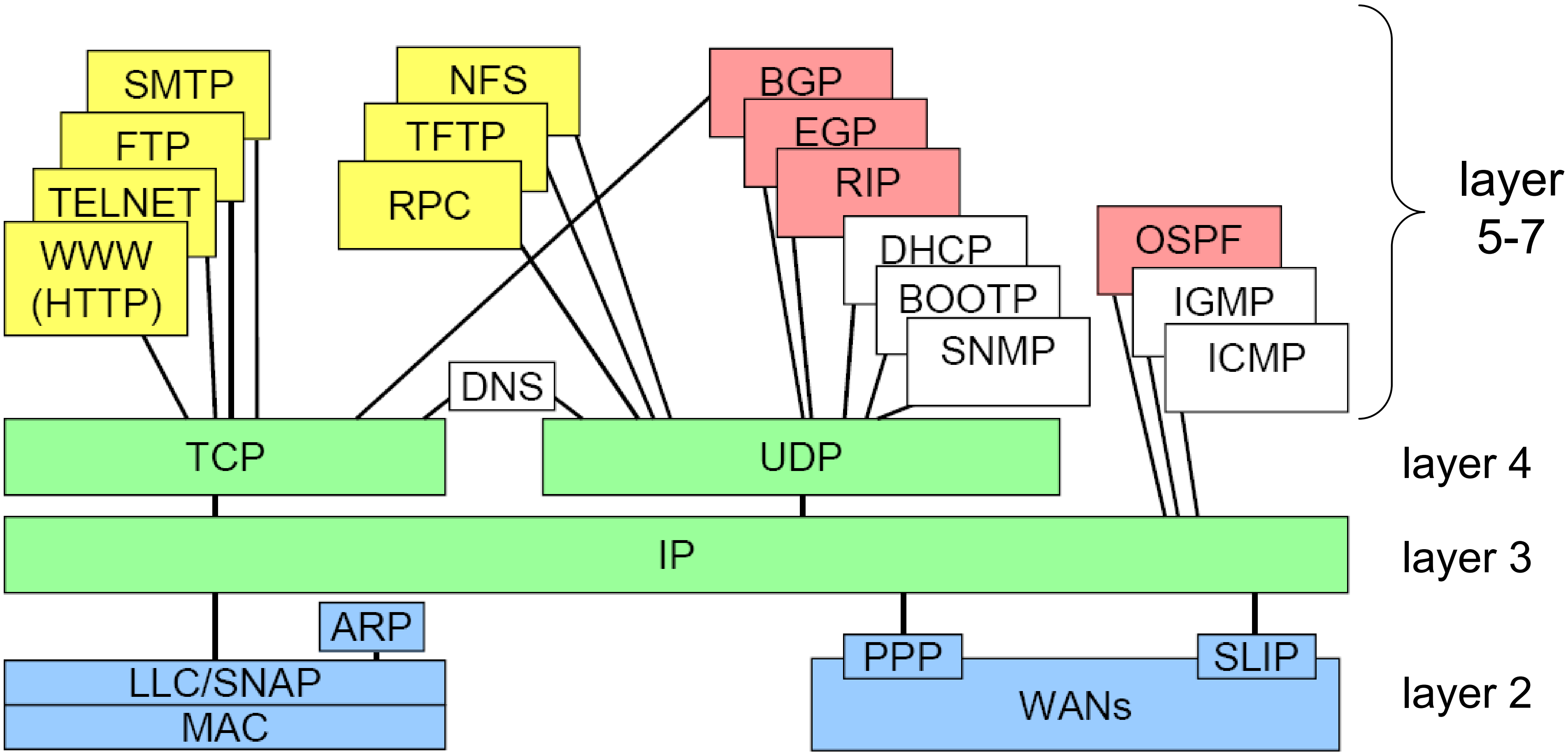
IP Networking
Internet Protocols
(TCP/IP Protocol Suite)

Contents - IP Networking - Internet Protocols

Internet Protocols Overview

Overview - TCP/IP Protocol Family



- TCP/IP refers to a whole bunch of protocols
- TCP/IP can be used on top of various layer 1/2 protocols

Overview - Protocol Examples

a) Layer 3 (Network Layer)

- IP (Internet Protocol): (connection-less) datagram protocol to transport layer 4 data units
- ICMP (Internet Control Message Protocol): protocol for exchanging control information
- ARP (Address Resolution Protocol): protocol for mapping IP addresses to respective layer 2 addresses (e.g. MAC addresses)

b) Layer 4 (Transport Layer)

- TCP (Transmission Control Protocol): connection-oriented, reliable transport protocol
- UDP (User Datagram Protocol): connection-less, unreliable transport protocol

Overview - Protocol Examples

c) Layer 5-7

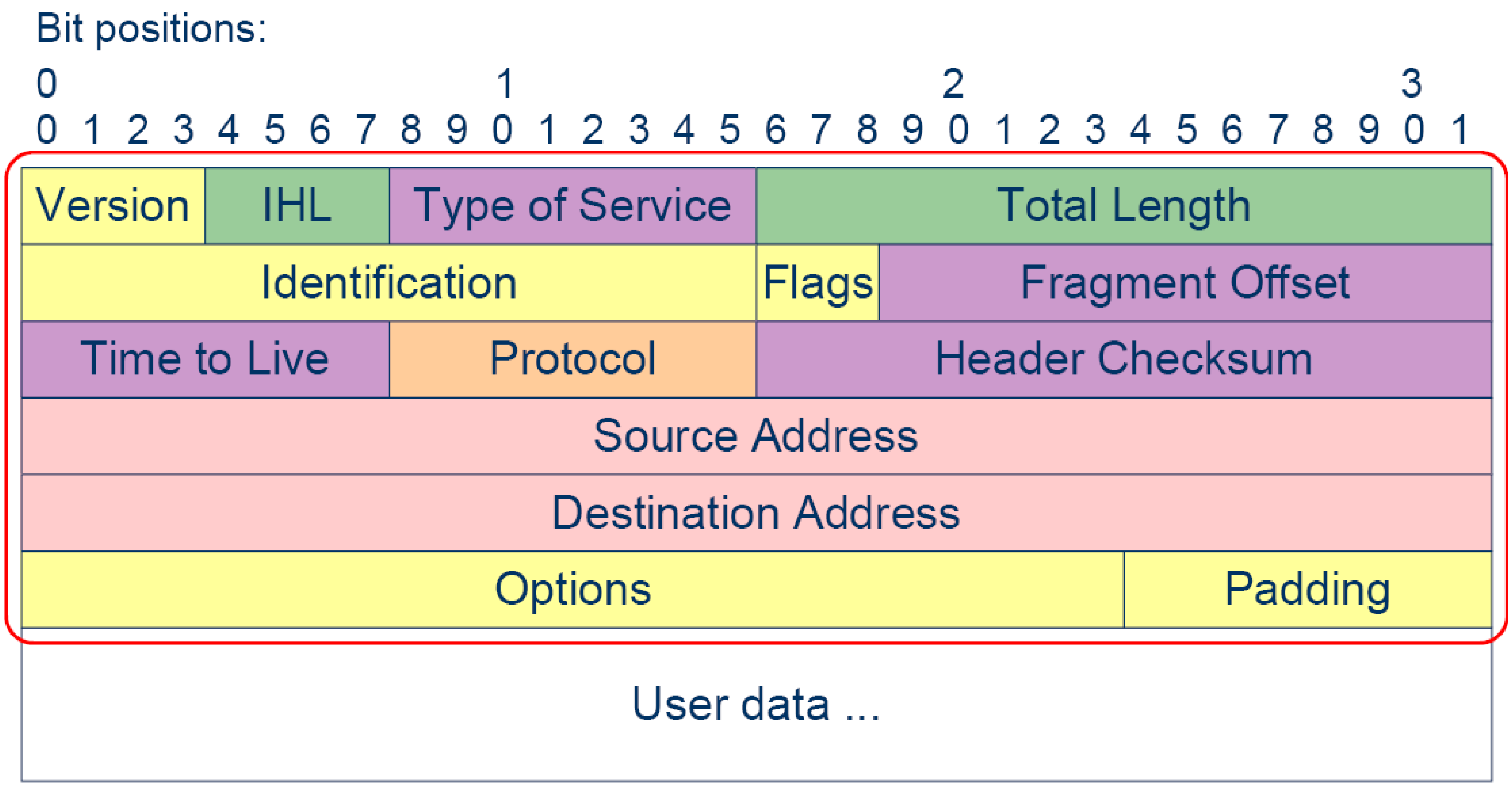
- SMTP (Simple Mail Transfer Protocol): protocol for exchanging text emails between hosts
- DNS (Domain Name Service): directory service for mapping names to addresses
- FTP (File Transfer Protocol): protocol for transferring whole files
- TELNET (Telecommunications Network): offers a virtual “terminal service”; allows to access a remote host from a TCP/IP-capable terminal (e.g. a PC), as if logged in directly at the host
- NFS (Network File System): allows accessing data on a remote system as if they would be stored on the local host (transparent file access)

Internet Protocol (IP)

IP Protocol - Characteristics

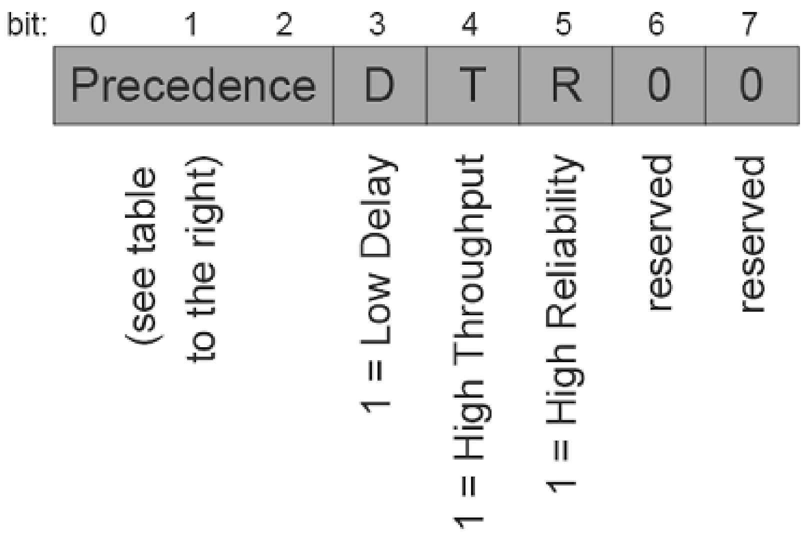
- Connection-less protocol (standardized in IETF RFC791, 1981)
- IP packet = datagram; header transmitted before datagram body
- Addressing: 32 Bit address in IPv4, 128 Bit address in IPv6
- Packet size: max. 65535 Byte, min. 20 Byte
- Transport: **Best Effort**
  - packets might be lost, duplicated or arrive in wrong order
  - no error handling: no error monitoring, no retransmission in case of errors, no flow control → error handling is left for higher layers (e.g. TCP)
- Checksum: only for header, not covering the datagram body
- Finite packet life time of packets (Time-to-Live, TTL)
- Fragmentation, i.e. splitting bigger packets into smaller ones if necessary (Segmentation and Reassembly, SAR)

IP Protocol - IPv4 Packet Format



IP Protocol - IPv4 Packet Header

- Version (4 bits)
  - Version 4: RFC791
- IHL, Internet Header Length (4 bits)
  - counts the number of 32bit words (=4 byte) of the header
  - minimum size = 5 (corresponds to a 20 byte header)
  - points to the first data word of the body
- Type of Service (8 bits) (old definition, RFC791)
  - for sevice-specific labeling of IP packets



Precedence Values

111	Network Control (inside a network)
110	Internetwork Control
101	CRITIC / ECP
100	Flash Override
011	Flash
010	Immediate
001	Priority
000	Routine

IP Protocol - IPv4 Packet Header

- Type of Service (8 bits) (new definition according to DiffServ, RFC2474)
  - new definition of the "Type of Service (TOS)" field in the IP header
  - 6-Bit DiffServ Codepoints (DSCP) instead of Precedence, D-, T- and R-fields
  - no guaranteed (but only relative) quality of service:
    - differentiated packet handling within routers
    - DSCP decides about in which queue the packets are sorted into and possibly about their discard priority
    - works only if supported by the routers (otherwise: all packets are treated equally)
  - since 2001 there is a new RFC3168:
    - bits 6-7 now used for ECN (Explicit Congestion Notification - IP Flow Control)

IP Protocol - IPv4 Packet Header

- Total Length (16 bits)
  - length of the IP packet (in octets), including IP header and data
  - maximum length: 65535 octets (Bytes)
  - all hosts must be able to receive an IP packet of length 576 octets (512+64)
  - a sender may only transmit IP packets of more than 576 octets length if the receiver is able to handle them
- Identification (16 bits)
  - for continuous labeling of transmitted IP packets
  - unique for the next 65535 packets
  - used by the receiver to re-order and re-assemble fragmented IP packets
- Control Flags (3 bits)
  - Bit 0: must be 0
  - Bit 1: DF-Bit: 1 = don‘t fragment, 0 = may fragment
  - Bit 2: MF-Bit: 1 = more fragments, 0 = last fragment

IP Protocol - IPv4 Packet Header

- Fragment Offset (13 bits)
  - shows to which position in the original IP packet's body the first octet of the current fragment's body belongs
  - counts in 8 octet words (64 bit)
- Time to Live (TTL) (8 bits)
  - describes the maximum time an IP packet may reside in the network
  - according to RFC791, TTL shall be measured in seconds
  - in reality the following procedure is applied: when forwarding a packet the router decrements the TTL by 1, i.e. the TTL represents the remaining hop count
  - TTL is necessary to avoid infinitely cycling packets (in case of a routing loop - special problem in connection-less packet routing)

IP Protocol - IPv4 Packet Header

- Protocol (8 bits)
  - the protocol number indicates to which protocol of the higher layer (usually transport layer) the packet in the body belongs
  - protocol number examples (according to RFC1700 "assigned numbers"):

1	ICMP	EGP	Exterior Gateway Protocol
2	IGMP	ICMP	Internet Control Message Protocol
4	IP	IGMP	Internet Group Management Protocol
6	TCP	IGP	Interior Gateway Protocol
8	EGP	IGRP	Cisco Interior Gateway Routing Protocol
9	IGP	IP	Internet Protocol
17	UDP	NHRP	Next Hop Resolution Protocol
46	RSVP	RSVP	Resource Reservation Protocol
58	NHRP	TCP	Transmission Control Protocol
88	IGRP	UDP	User Datagram Protocol

IP Protocol - IPv4 Packet Header

- Header Checksum (16 bits)
  - the checksum **covers only the IP header**
  - the checksum is modified by every node that changes the header (e.g. the TTL field) (usually done by routers)
- Source Address (32 bits)
  - IP address of the host that sent the IP packet (source)
- Destination Address (32 bits)
  - IP address of the host towards which the IP packet is sent (destination)
- Padding
  - fills the IP header to the next multiple of 4 octets

IP Protocol - IPv4 Packet Header

- Options (with variable length)
  - either none (0) or multiple options
  - cases for multiple options:
    - case 1: single-octet option
    - case 2: multi-octet option → represented as follows:
      option type (1 octet) + option length (1 octet) + option value

option type:

bit:	0	1	2	3	4	5	6	7
	CF	OC		Option Number				

Copy Flag (CF):
1 = copy / 0 = do not copy
(controls the handling of options in case of fragmentation)

Option Class (OC):
0 = control
2 = error monitoring and classification
1,3 = reserved for future use

option length: number of octets in {option type + option length + option value}

IP Protocol - IPv4 Packet Header

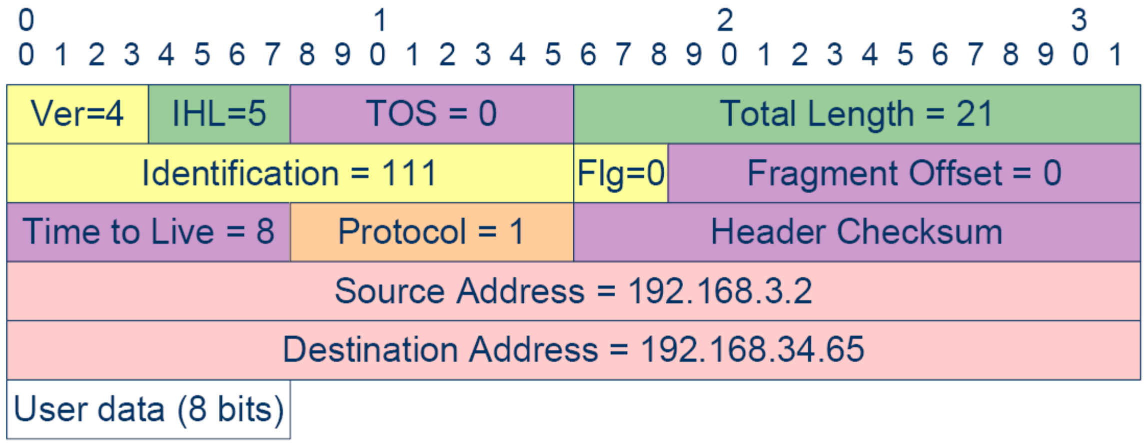
- Option Examples

Option Type

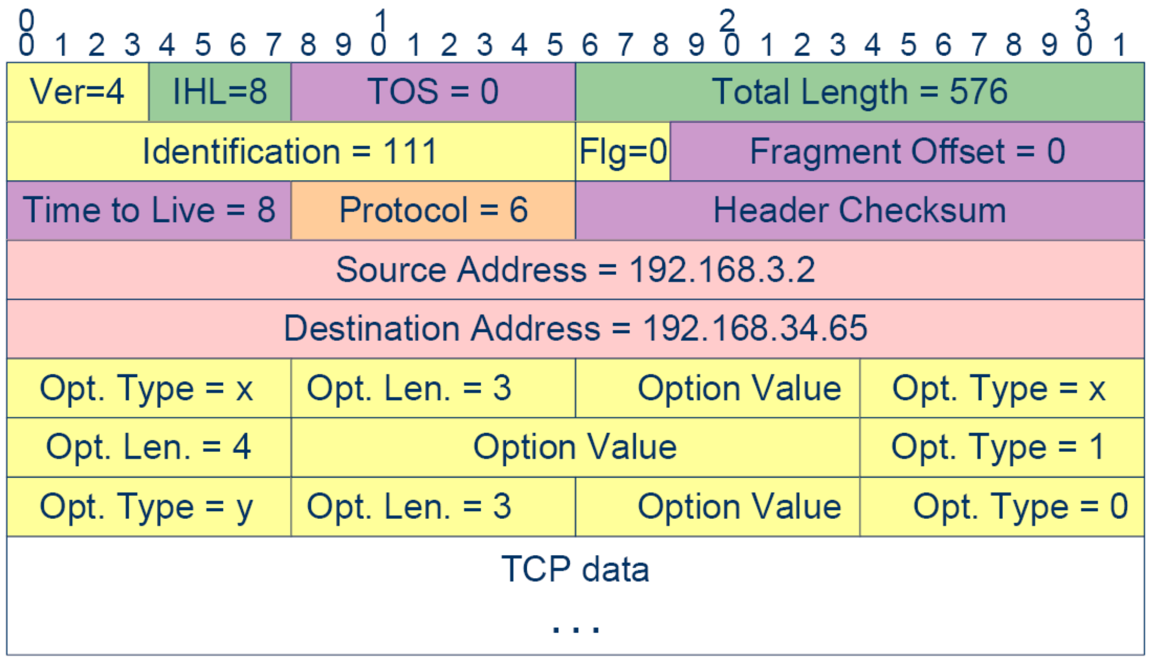
Option Class	Option Number	Length	Description
0	0	–	End of option list. Used for padding. No length octet.
0	1	–	No operation. No length octet.
0	2	11	Security and handling restrictions (US DoD)
0	3	var.	Loose source route. Request routing along the specified routers.
0	7	var.	Record Route. Collect the addresses of routers along the path.
0	8	4	Strem identifier (SATNET, obsolete)
0	9	var.	Strict source route.
0	11	4	MTU probe. Used for path MTU discovery.
0	12	4	MTU probe reply. Used for path MTU discovery.
0	20	4	Router alert. Request router to process end-to-end packet contents. (RFC2113)
2	4	var.	Record timestamps along a route.
2	18	var.	Traceroute. To find routers along a path.

IP Protocol - IPv4 Packet Header

- Option Examples
  - IP packet with only one octet payload in the body and without options



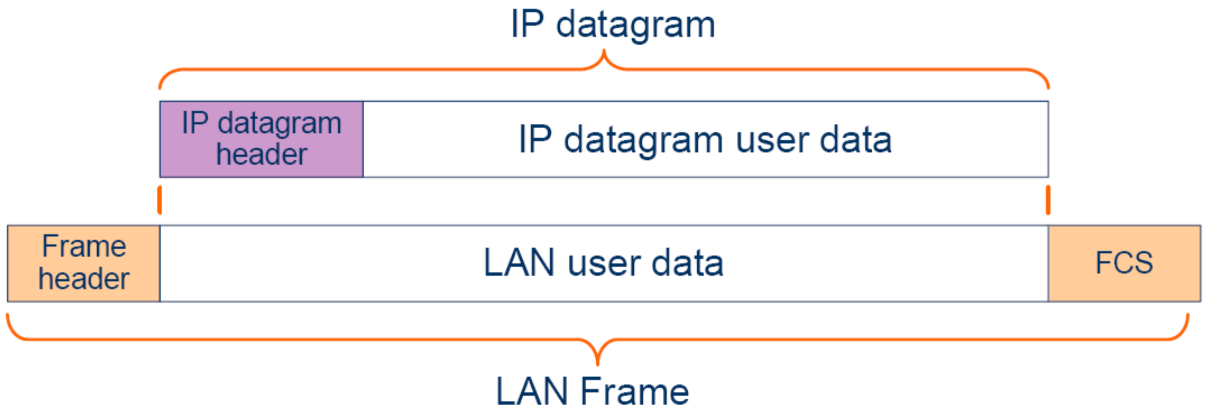
  - IP packet with TCP payload in the body and 4 options



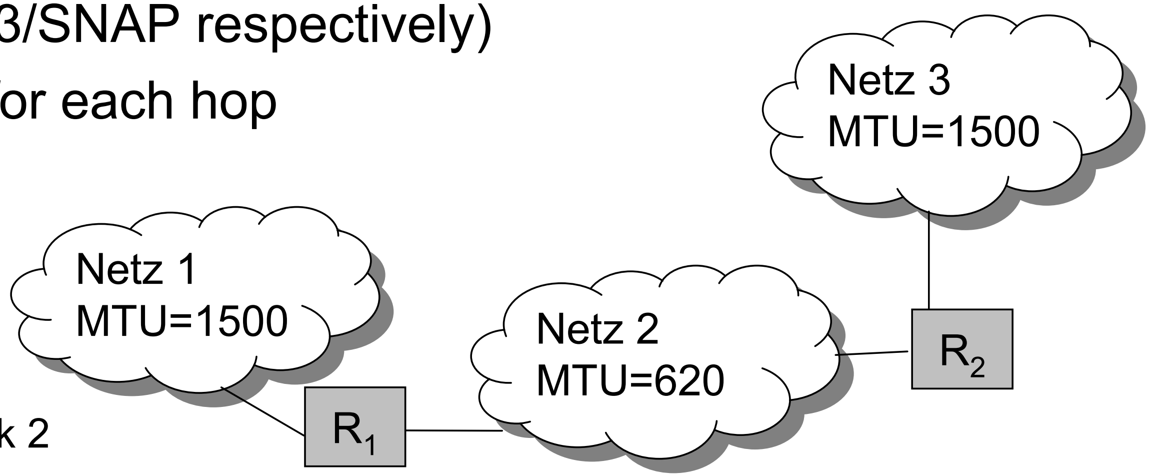
IP Protocol - Fragmentation (SAR)

Necessity of fragmentation

- a layer 2 network (e.g. a Ethernet LAN) treats an IP packet as payload within a layer 2 frame:



- problem: the maximum size of the payload of a layer 2 frame (Maximum Transfer Unit, MTU) limits the size of the IP packet, e.g. 1500 octets for Ethernet (1492 octets for IEEE802.3/SNAP respectively)
- the MTU can be different for each hop
  example:



router R_1 cannot send IP packets with 1500 octet length into network 2

IP Protocol - Fragmentation (SAR)

Two alternatives for fragmentation

a) introduction of a special SAR protocol that performs a link-by-link segmentation and re-assembly (SAR) of the IP packets

   problem: IP would only work on links that apply the SAR protocol

b) the IP protocol adapts itself to the MTU of the next hop

   possible cases:

   - per-network SAR ("Intranet SAR")
     benefit: always permits to use the highest possible IP packet length
     drawback: requires higher computation effort in the nodes
   - end-to-end SAR ("Internet SAR"), re-assembly only at the destination

     benefit: simplicity

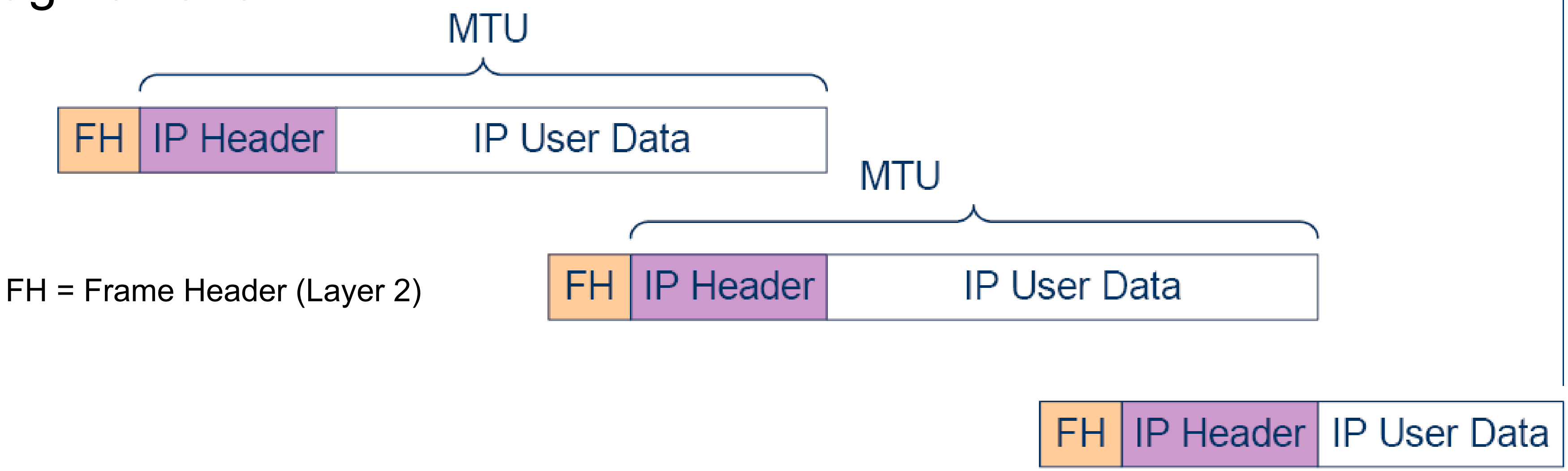
     drawback: worse efficiency (allows only to use the minimum MTU along the route)

IP Protocol - Fragmentation (SAR)

Fragmentation mode of operation

- original IP packet

- fragmentation



remarks:
- multiple fragmentations are possible
- the fragments may have different sizes

IP Protocol - Fragmentation (SAR)

Fragmentation mode of operation

- each fragment has its own IP header
  - the identification (ID) of the original IP packet is copied into the fragment header → used to detect the fragments that belong together during re-assembly
  - the options are copied only if the copy flag (CF) is set to 1
- fragment offset
  - the fragment offset is an 8 octet (64 bit) counter within the IP header
  - it shows the position of the fragment body's first octet in the body of the original IP packet
- more fragment flag (MF)
  - MF = 0 marks the last fragment
  - MF = 1 indicates that more fragments are following

IP Protocol - Fragmentation (SAR)

Fragmentation mode of operation

- Don't Fragment Flag (DF)
  - DF = 0: fragmentation is allowed
  - DF = 1: fragmentation is not allowed
  - routers send corresponding ICMP messages back to the sender; this might be used to test the maximum IP packet size along the route to the destination; with that, segmentation and re-assembly at the receiver can be avoided
- Note: the length field in the IP header shows the length of the fragment; not the length of the original IP packet!
  - the original IP packet's length is known only after receiving the last fragment (MF=0) and after complete re-assembly
  - the recipients have to reserve buffer memory for re-assembling

IP Protocol - Fragmentation (SAR)

Fragmentation mode of operation - example MTU=620

- original IP packet (length 1420 octets):

IP Header	IP User Data, 1400 octets

- first fragment:

IP Header	IP User Data, 600 octets	MF=1	FO=0	data offset=0

- second fragment:

IP Header	IP User Data, 600 octets	MF=1	FO=75	data offset=600

- third (last) fragment:

IP Header	Data, 200 oct.	MF=0	FO=150	data offset=1200

FO Fragment Offset
MF More Fragments

IP Protocol - Fragmentation (SAR)

Fragmentation mode of operation

- maximum payload size in the fragments:

$$L_{FD} = \text{MTU} - L_{IH}$$

- number of fragments

$$n_F = \lceil (L_D - L_{IH}) / L_{FD} \rceil$$

L_D	length of the IP packet
L_{FD}	length of the fragment payload
L_{IH}	length of the IP header
MTU	Maximum Transfer Unit
n_F	number of fragments

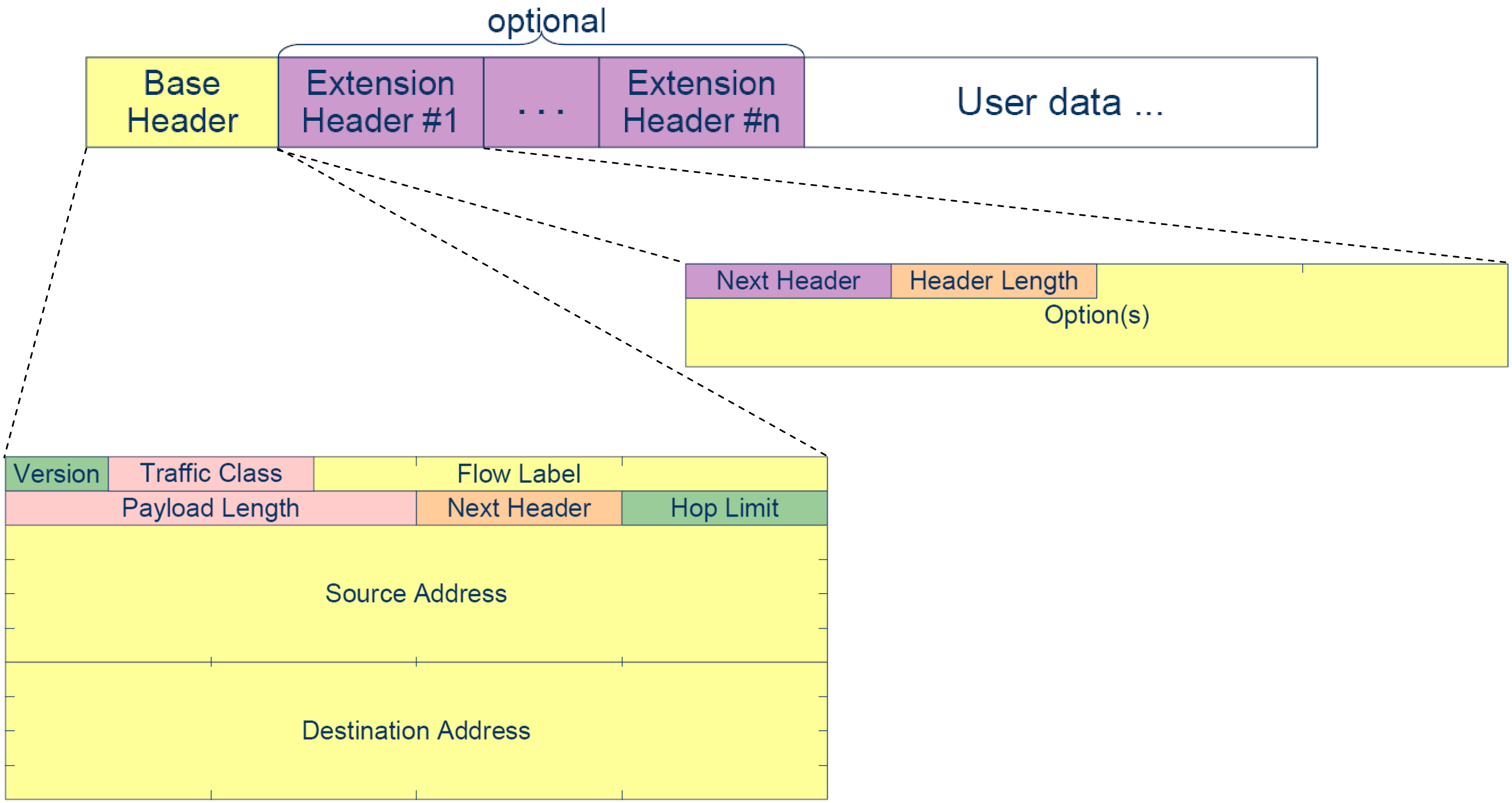
IP Protocol Version 6 (IPv6)

- IP Version 6 (IPv6) is the successor of IPv4
  - also known as IPnG (IP next Generation)
  - Version 5 was used for the (experimental) stream protocol (ST)
  - IPv6 is defined in the RFCs 1883, 2460, 2462-2464, 2373-2375, 2526
- Motivation for IPv6
  - bigger address range
  - enhanced QoS mechanisms on network layer
  - enhanced (end-to-end) security mechanisms on network layer
  - auto-configuration
- Completely new header format
  - longer addresses
  - new options

IP Protocol Version 6 - IPv6 Addresses

- 128 bit (16 octet) address length (RFC1887)
  - allows more than 1024 addresses per m² of earth's surface
  - allows structured address design
- Address types
  - unicast addresses
    - globally aggregated or locally assigned (according to link, location, ...)
  - anycast addresses
    - packets are delivered to the address (within the anycast address space) that is accessible on the shortest path
  - multicast addresses
- Representation of IPv6 addresses:
  - 8x16bit integers, represented as 4 hex numbers
    example: 108:0000:0000:0000:00AB:0801:200E:3AB2
  - short form 1: 1080:0:0:0:AB:0801:200E:3AB2
  - short form 2: 1080::AB:0801:200E:3AB2
    ("::" means as many "0" fields as necessary)
  - prefixes are written as decimal numbers
    example: 1090:2ADC:3300::/40 for a 40 bit prefix

IP Protocol Version 6 - IPv6 Packet Format



IP Protocol Version 6 - IPv4 vs. IPv6 Packet Format

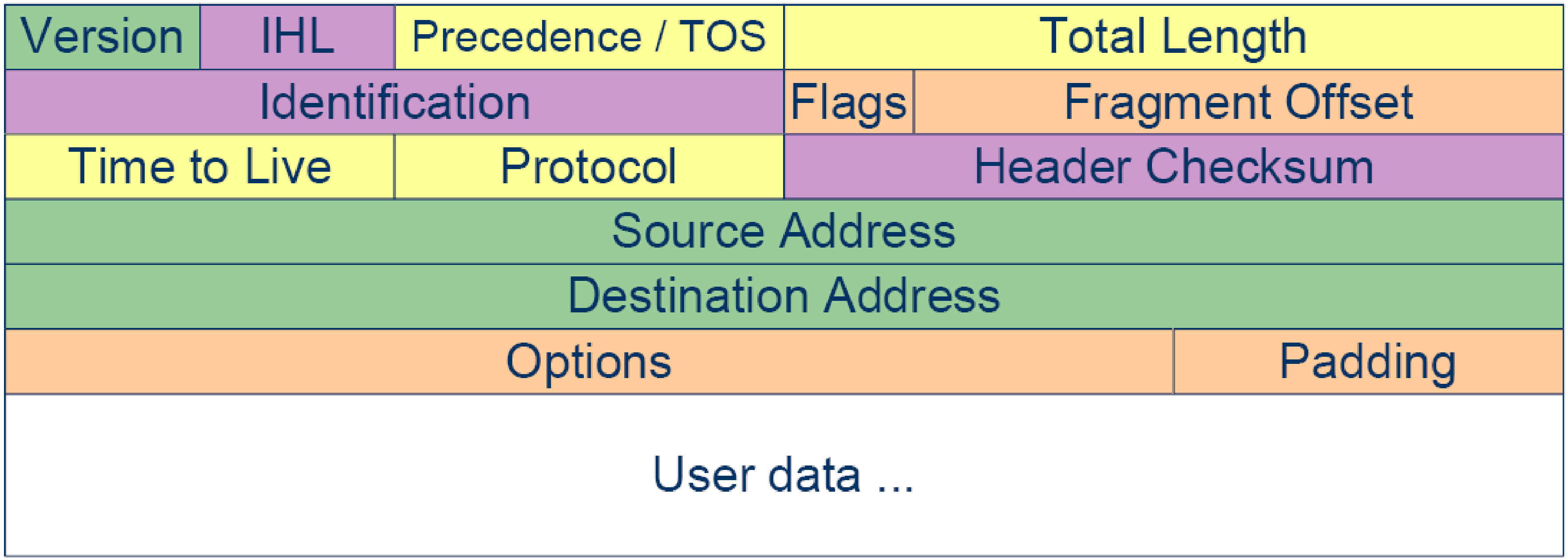


Illustration of the differences between IPv6 and IPv4 headers:

- field still used in IPv6
- field missing in IPv6
- field changed in IPv6 (new meaning)
- field moved to extension header in IPv6

IP Protocol Version 6 - Reasons pro/contra IPv6

- Reasons for introducing IPv6:
  - solution for the address shortage
  - support of flow labels (for labeling and differentiated handling of packet flows)
  - full support for IPv6-in-IPv4 tunneling and IPv4/IPv6 interworking (on the other hand IPv4 over IPv6 support is not defined or required yet)
- Reasons against introducing IPv6:
  - modifications to all routers and end systems required
  - modifications also required in application software
  - IPv4 is well established and approved
  - enormous overhead due to long IPv6 addresses (esp. for short packets)
  - the problem of address shortage is not as critical as previously expected (because of NAT, etc.)
  - IPv6 does not support "Label Stacking" (the flow label field cannot be used for MPLS)

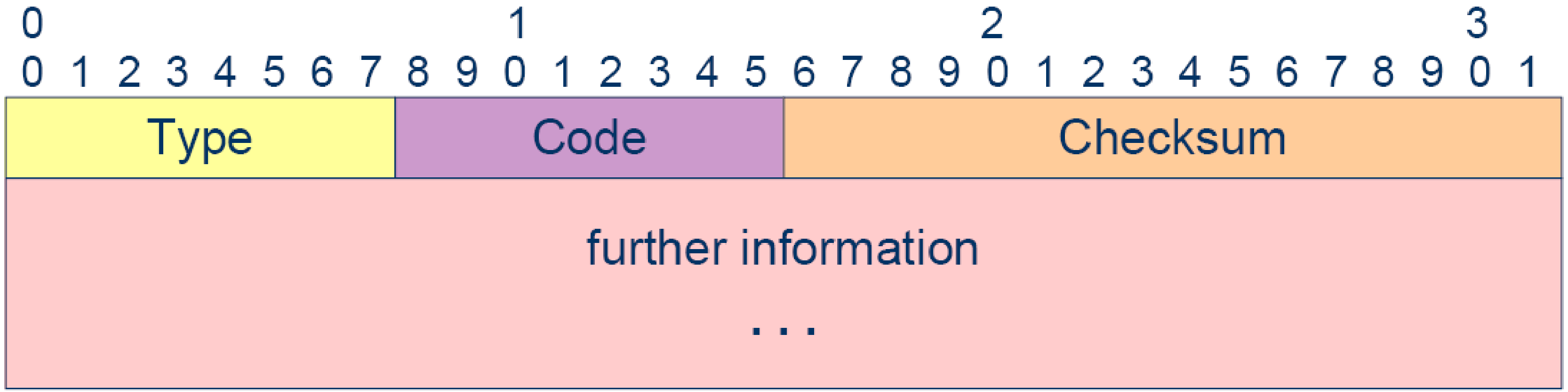
Internet Control Message Protocol (ICMP)

ICMP Protocol - Overview

- ICMP is a companion protocol to IP
  - is part of each IP protocol implementation
  - uses the payload field of IP (ICMP encapsulated in the IP packet payload)
  - uses protocol number 1
- ICMP is used to report problems during the forwarding of IP packets
  - the ICMP message is addressed to the source IP address of the IP frame
  - (the corresponding application in) the source can evaluate the ICMP message
  - die ICMP error message contains the header of the IP packet that created the error + the first 8 octets of its payload
  - error correction is not provided
- no error reports for erroneous ICMP packets themselves to avoid avalanches of error messages

ICMP Protocol - ICMP Format

- general ICMP frame format
  - 4 octets common part of all ICMP messages
  - depending on the message type further information is appended



- the same checksum algorithm as for IP packets is used
  - the checksum covers only the ICMP message
- the type field defines the content and format of the ICMP message
- the code field specifies details of the ICMP message

ICMP Protocol - Reachability Test via Echo Request/Reply

- "Echo Request/Reply" is used for reachability testing
  - the recipient of the echo request responds with an echo reply
  - optionally data from the echo request is copied in the echo reply
- Used in the “ping“ application:
  - ping sends an ICMP echo request
  - and waits for a corresponding echo reply with fitting identifier and sequence number

0	8	16	31
Type (8 / 0)	Code (0)	Checksum	
Identifier		Sequence Number	
Optional data . . .			

Identifier: used to denote the application process; created by the source (as port number); the destination replies to this port

Sequence Number: continuous numbering of echo request messages by the source; the destination uses the same number in its echo reply

Data Field: the source fills this field with random data; the destination does not change this field in its replies

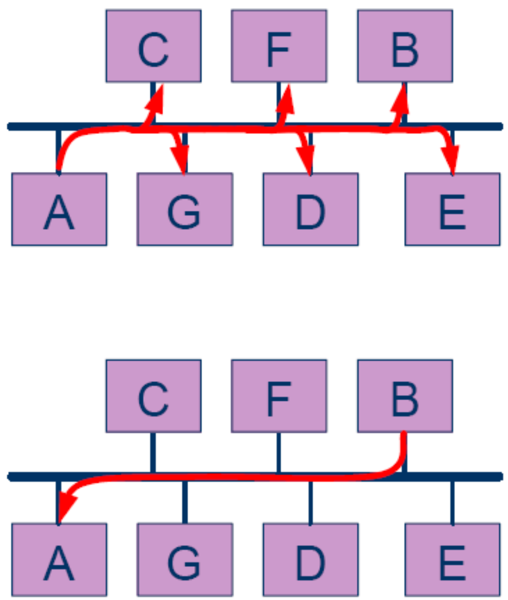
Address Resolution Protocol (ARP) and Reverse ARP (RARP)

ARP Protocol

- Layer 2 addresses are used in local networks, e.g. MAC addresses in Ethernet LANs
- Issue: a source host wants to transmit to a destination (located in the same local network), but it only knows its layer 3 address (IP address); to get its layer 2 address a special mechanism is required (Address Resolution Problem)
- Example:

ARP:

determine the layer 2 address (MAC address), that belongs to a certain IP address in the local network



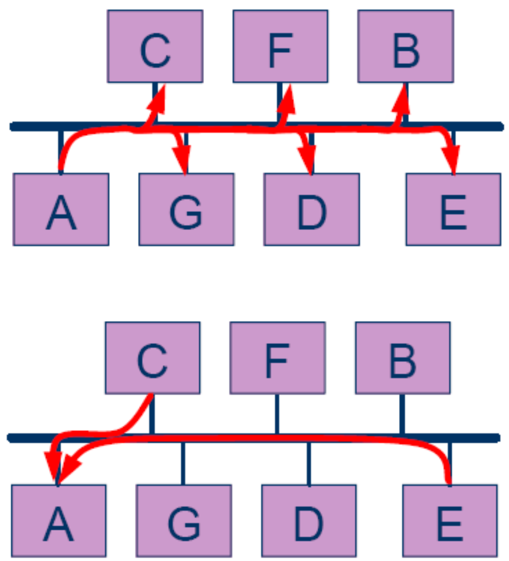
- A wants to transmit a packet to a host with IP address IP_B
- A sends an ARP broadcast to all hosts in the local network; the broadcast contains its own IP and MAC address as well as the IP address of B (IP_B)
- B realizes (due to IP_B) that it is addressed, updates its ARP Cache with A's data and sends a unicast reply back to A
- A now knows B's MAC address and updates its ARP Cache accordingly

RARP Protocol

- The RARP protocol is used to determine the IP address that belongs to a layer 2 address (MAC address) of a host
- Used during the boot phase of certain hosts (e.g. "diskless" hosts) and requires a RARP server in the local network
- Mode of operation:
  - the host sends a RARP request using a layer 2 broadcast address; it contains the layer 2 address of the requesting host
  - the RARP server sends a unicast reply to the layer 2 address of the requesting host; it contains the requested IP address of the host
- Example:

RARP:

determine the IP address that belongs to a certain layer 2 address in the local network



- A just booted and sends a RARP broadcast into the local network
- the RARP servers C and E know the layer 2 address of A and send a unicast reply to A that contains the requested IP address

ARP/RARP Packet Format

<table>
<tr><th>0</th><th>8</th><th>16</th><th>31</th></tr>
<tr><td colspan="2">Hardware Type</td><td colspan="2">Protocol Type</td></tr>
<tr><td>HLEN</td><td>PLEN</td><td colspan="2">Operation</td></tr>
<tr><td colspan="4">Sender HA (octets 0–3)</td></tr>
<tr><td colspan="2">Sender HA (octets 4–5)</td><td colspan="2">Sender IP (octets 0–1)</td></tr>
<tr><td colspan="2">Sender IP (octets 2–3)</td><td colspan="2">Target HA (octets 0–1)</td></tr>
<tr><td colspan="4">Target HA (octets 2–5)</td></tr>
<tr><td colspan="4">Target IP</td></tr>
</table>

RARP Ethertype = 8035_{16}

ARP Ethertype = 0806_{16}

Protocol Type: 0800_{16} = IPv4

Operation:
- 1 ARP Request
- 2 ARP Response
- 3 RARP Request
- 4 RARP Response

ARP	Address Resolution Protocol
HA	Hardware Address
HLEN	HA Length
PLEN	Protocol Address Length
RARP	Reverse ARP

ARP/RARP Packet Format - Header Fields

Hardware Type =	**Hardware (HW) (16),** Ethernet V2 = 0001, IEEE 802 = 0006
Protocol Type =	**Protocol Type (16)** = Protokolltyp des verwendeten Protokolls der Schicht3, z.B. IP = 2048 Dezimal = 0800 Hex = 0x800
HLEN =	**HW(Hardware)-Length (8):** Länge der Hardware-Adresse in Bytes, z.B. für Ethernet und alle IEEE 802-Netze 48 Bit = 6 Bytes, HW = 0x6.
PLEN =	**SW(Software)-Adress-Length (8Bit)=** Protocol-Adress-Length = Länge des verwendeten Protokolls der Schicht 3 in Bytes, IP 32 Bit = 4 Bytes, SW-Length = 0x4.
Operation =	**Optionscode (oder Operationscode)** = Art des ARP **1 = ARP-Request,** **2 = ARP-Reply,** **3 = RARP-Request,** **4 = RARP-Reply.**
Sender HA =	**HW (Hardware)-Source-Adress (48)** = Hardware (MAC)-Adresse = physikalische Adresse des den ARP sendenden Gerätes.
Sender IP =	**SW (Software)-Source-Adress (32)** = Software o. logische Adresse (z.B. IP) des sendenden Grätes.
Target HA =	**HW (Hardware)-Destination-Adress(48)** = Hardware (MAC)-Adresse des Ziel-Gerätes, beim Beginn wird der ARP-Request an alle als Broadcast mit 48 Einsen als HW-Dest –Adress gesendet.
Target IP =	**SW (Software)-Destination-Adress (32)** = Software oder logische Adresse (z.B. IP) des Zielgerätes.