6 Networking and Routing

6.1 Tasks and service classes

The task of the **network layer (layer 3 in the ISO/OSI model)** is to transport packets from the sender (source) to the receiver (destination). It represents the "delivery system" of the communication network. The basic prerequisite for this is a precise knowledge of the network topography.

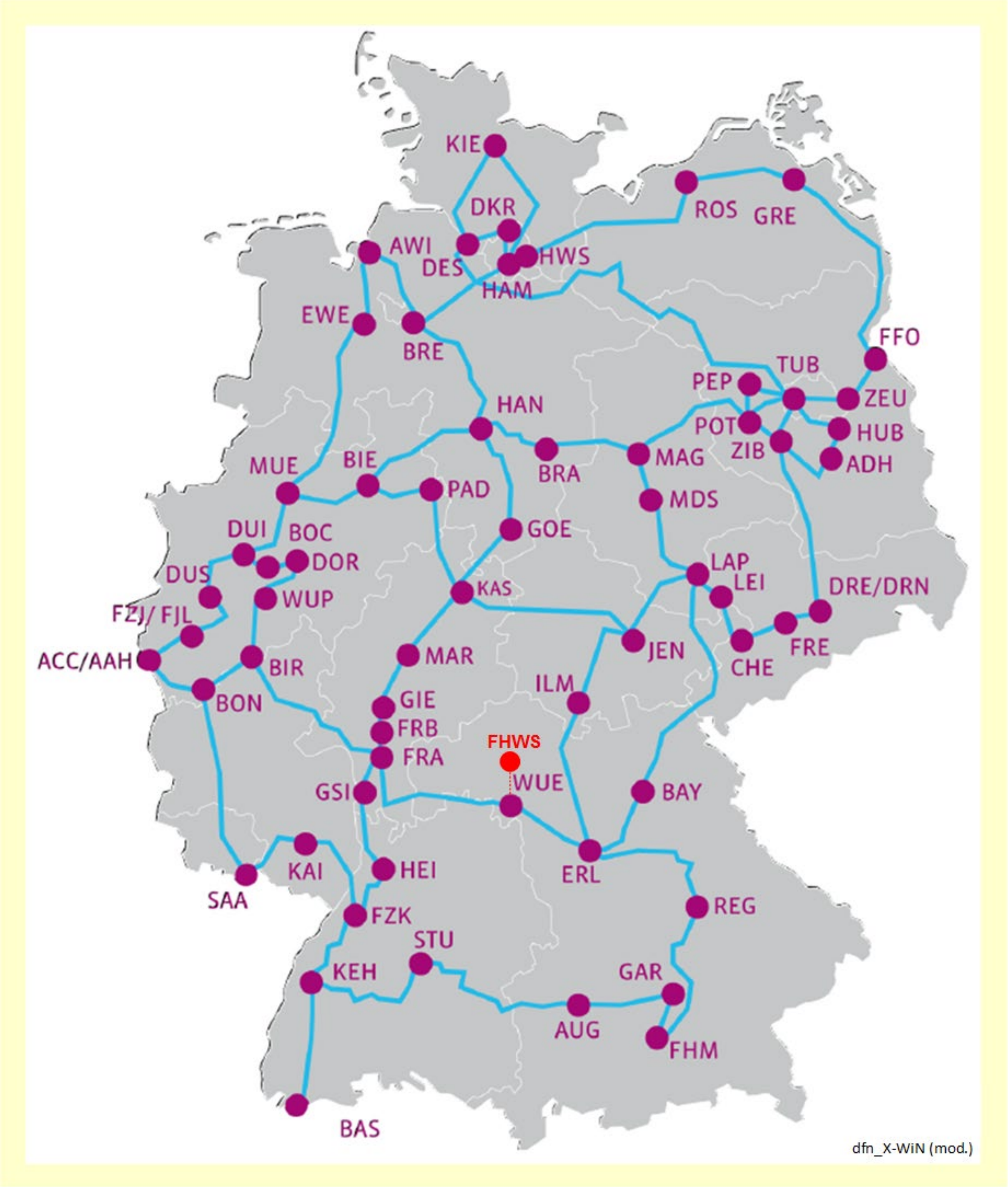


Fig. 6.1-1: Topology of the fiber-based scientific network X-WiN of the DFN-Verein (DFN: Deutsches Forschungsnetz) /DFN2018 mod./

During data communication in the network, two processes or programs exchange data. The main task of the network layer is to transmit the information to the receiver and transfer it to the transport layer there. If necessary, the transmission path is broken down into different sections between the network nodes: **Point-to-point connections**.

The important tasks involved are

- Selection of packet routes in the communication network,
- Routing from the source to the destination,
- Conversion of addresses at the transition between different networks,
- Data flow control: prevention of congestion when several data packets are moving at the same time,
- Provision of an accounting function for the network operator (counting of data traffic, charging and calculation, accounting).

The packets are routed from the sender to the receiver in spatially extended networks via so-called routers. Routers are hardware devices with software of layers 1 to 3 according to the ISO/OSI model. They connect different subnetworks. Only those packets are interpreted that are addressed to the corresponding router. By default, no packet transport takes place.

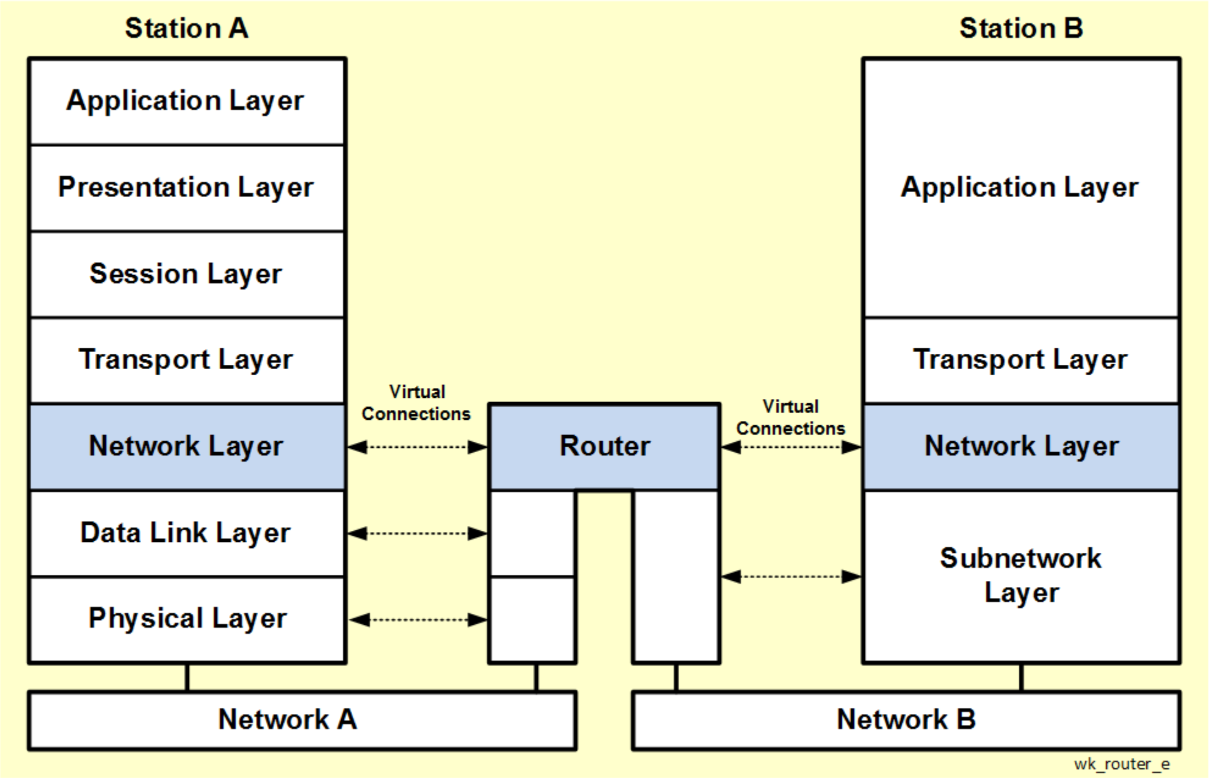


Fig. 6.1.2: ***Router*** **between two networks**

Multiprotocol routers can route packets between networks with different protocols.

In a complex communications network, the network layer represents the interface between the network operator (provider) and the customer. The layer must therefore be clearly defined.

The following design criteria are of primary importance:

- Independence of the services from the technology of the subnetworks,
- Shielding of the transport layer (layer 4) from the subnetworks present in the communication system,
- Provision of a uniform format for the network addresses in all subnetworks.

The network layer provides the user with two different classes of service:

- **Connection-oriented service class:**
  - Establishing a virtual connection: each packet chooses the same route,
  - All routers involved must know the connection route
  - Local labeling of each packet with an additional number field in the header to indicate the virtual connection (independent numbering in each router),
  - Short failure of a router terminates current virtual connection,
  - Example: ATM networks (ATM: Asynchronous Transfer Mode).

- **Connectionless class of service:**
  - Sending independent packets (datagrams),
  - Each packet contains full source and destination address,
  - No route selection: each packet is transported independently of its predecessor,
  - Difficult congestion monitoring,
  - Routers do not keep routing tables as in virtual connections,
  - Routers possess tables about reachable other routers,
  - Example: Internet (Internet Protocol (IP)).

In practice, combinations between the two classes of service are also possible and in use, e.g. the application of the Internet protocol in ATM networks.

6.2 Internet Protocol IPv4

6.2.1 Properties

The **Internet Protocol (IP Protocol)** is integrated into the network layer (Internet layer) of the TCP/IP protocol suite. Its task is to transport data packets, datagrams (IP datagrams), from the sender to the receiver. The characteristic features of the Internet protocol are

- **Connection-independent (connection-free) high-performance data transfer service:**
  - no flow control and no debugging in the data part,
  - no feedback to the sender in case of error,
  - independent routing decisions for each datagram,

- **Unreliable switching service:**
  When communicating over the connection-independent IP protocol, the following error sources can occur:
  - Loosing IP datagrams due to overloading of a router,
  - errors in a datagram, error in routing,
  - Incorrect sequence of datagrams due to the selection of different transport routes for different datagrams,
  - duplication of datagrams received due to errors in the timer of the sender's transport layer of the sender.

Each Internet datagram is an independent data unit (in contrast to the data stream) and consists of a header (IP header) and a data part.

6.2.2 Header of the Internet Protocol

The Internet Protocol header (IP header) is generated by the network software and contains all information for delivering the encapsulated datagrams to the receiver (cf. figure). Its length is always an integer multiple of a 32-bit word (4 bytes).
The components of the IP header (IPv4) are briefly explained in the following overview:

- **Version number (VERS):**
  - Development number of the current IP version (currently still and mainly used version: IPv4),
  - Software can reject incompatible (outdated) datagram versions,

- **Length of header (header length HLEN):**
  - Length of the header in units of 32-bit words (4 bytes),
  - Normal length without options: five 32-bit words = 20 bytes,
  - position of the first data byte:
    `First Databyte = First byte of the IP datagram +(HLEN·4).`

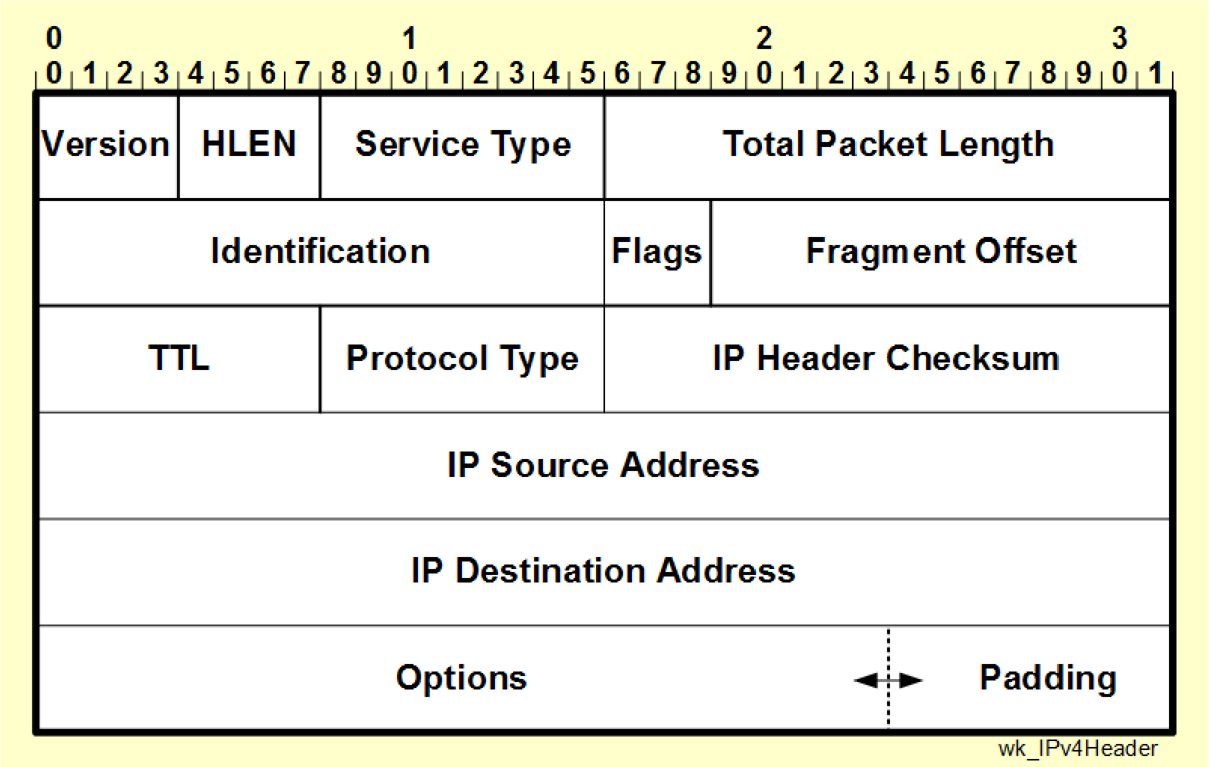


Fig. 6.6.2-1: Schematic structure of the IPv4 header

- **Service type (type of service TOS):**
  - Indication of the priorities of the IP datagrams,
    e.g. Telnet session with few data: Minimizing the delay time,
    File transfer with FTP: maximizing the throughput,

  - Bit sequence (from left to right):

Bit	Type of data delivery priority
0,1,2	Priority over other datagrams (000, 001, ..., 111)
3	Delay: 0 = Normal waiting time,
	1 = Low waiting time,
4	Throughput: 0 = Normal throughput
5	Reliability: 0 = Normal reliability
6	Cost 0 = Normal cost
7	Reserved: 0

  - Service type information can be ignored by routers,

- **Total packet length:**
  - Length of the entire IP datagram including the IP header in units of bytes (as opposed to header length HLEN),
  - Theoretical maximum length: 216 bytes = 65535 bytes,

  - Start of data:

```
First data byte = First byte of header + (HLEN · 4).
```

- End of data:

```
Last data byte = First byte of header +
                 + Total packet length
```

- Length of data:

```
Data length = Packet length - First data byte
```

- **Identification:**
  - Identification of the fragments with respect to their affiliation to the datagrams,

- **Flags:**
  - Auxiliary variables for the assembly of fragmented datagrams,

- **Fragment offset:**
  - Position of the fragment stored in a datagram (partial message) relative to the beginning of the original datagram,

- **Datagram survival time (time to live TTL):**
  - Typical start value when sending: 32,
  - Decrementing the TTL value when passing a router,
  - If TTL = 0, destruction of the datagram by a router and sending a corresponding message via ICMP (Internet Control Message Protocol) to the host,

- **Protocol type:**
  - The protocol used to encapsulate the data contained in the datagram,
  - Values for the protocol field:

Protocol	Value (decimal)	Value (binary)
ICMP	1	0000 0001
IGMP	2	0000 0010
TCP	6	0000 0110
UDP	17	0001 0001

- **IP header checksum:**
  - Checksum to verify the header information of a datagram (Assumption: value zero in the checksum field!),
  - Consideration of the header information as a sequence of 16-bit numbers,
  - Summation according to one's complement arithmetic,
  - storage of the obtained sum in the checksum field (correct transmission: Checksum field contains only ones when received),
  - Error case: datagram is discarded without feedback,

- **IP source address:**
  - Internet protocol address of the sender,
  - no changes by the router on the communication path,

- **IP destination address:**
  - Internet protocol address of the receiver,
  - no changes by intermediate routers,

- **Options:**
  - Aids for testing and debugging the network software,
  - Control over the routing of datagram fragments,
  - Options are not supported by all routers and hosts,
  - Structure of three subfields (bit counting from left to right):

Bit(s)	Subfield
0	Copy
1,2	Option class:

Option class	Bit pattern	Task
0	00	Datagram or network control
1	01	Reserved
2	10	Debugging and Measuring
3	11	Reserved

Bit(s)	Subfield
3,4,5,6,7	Option number: - Selection of a specific option:

Class	Number	Length	Description
0	2	11	Security (Military)
0	3	variable	Source routing, loose
0	7	variable	Record-Route
0	9	variable	Source routing, rigid
2	4	variable,	Internet time signature

- **Padding region:**
  - Padding with zero bits so that the total length of the header (HLEN) is an integer multiple of 32-bit words.

6.2.3 Internet Checksum

The checksum calculation used in the Internet Protocol in the network layer is comparatively simple:

- The entire transmitted data packet is broken down into words (unsigned 16-bit integer values).
- All transmitted words are added according to 16-bit one's complement arithmetic.
- The complement of the result of the addition corresponds to the checksum.

The checksum calculation is fast, but relatively insensitive to error combinations that occur compared to the CRC check. For example, swapped words in the data packet cannot be detected.

Exercise

E.6.2.3-1: Internet Checksum

Develop a function to calculate the Internet checksum

```
unsigned short cksum(unsigned short *buf, int n)
```

with **`buf`**: data buffer,
`n`: length of the buffer in 16-bit units.

It is to be assumed that, if necessary, the data packet has already been extended to the 16-bit size with padding bits.

6.2.4 Fragmentation

Long data packets are broken down into shorter fragments during communication transmission: **Fragmentation**.
TCP/IP allows a maximum datagram length of 216 bytes = 65536 bytes. In contrast, various network technologies further restrict the **maximum packet sizes (maximum transfer block, maximum transfer unit MTU)**:
If the maximum packet size is exceeded, fragmentation is automatically performed by the IP protocol.

Examples of maximum packet sizes (maximum transfer unit MTU) for different network technologies:

Network technology	maximum packet size (MTU)
Ethernet	1518 Byte
IEEE 802.3	1518 Byte
FDDI	4495 Byte
Transport-Protokoll TCP	576 Byte
Token Ring (16 MBit/s)	17914 Byte
X.25 (Maximum)	1024 Byte
X.25 (Standard)	128 Byte

The IP protocol always tries to send the largest possible data packets. Their total length is always an integer multiple of eight bytes (64 bits), the unit of fragmentation. The beginning of a fragment, calculated from the beginning of the original datagram in units of bytes, is stored in the 13-bit header field "Fragment Offest".

To reconstruct the datagrams from the individual fragments, the IP protocol uses the last two bits of the 3-bit-long flag field in the IP header:

Bit	Description	Comment
0	reserved	Bit always contains the value zero,
1	***Don't-Fragment-Bit DF***	Value 1: no fragmentation
2	***More-Fragment-Bit MF***	Value 1: more fragments follow, Value 0: last fragment.

When the first datagram is received with the "More fragments" bit set, the receiver starts a timer, the **"Reassembly Timer"**, and stores the received datagrams in a data buffer. Based on the source addresses and the identification fields, the fragments are correctly assembled by the IP protocol.

If the reassembly timer expires before all fragments are received, the fragments received so far are discarded.

6.2.5 Addressing

6.2.5.1 Internet Protocol addresses

In a TCP/IP communications network, such as the Internet, globally unique, logical **Internet Protocol (IP) addresses** identify individual IP interface cards. Each network adapter on the Internet must be assigned a unique IP address. When a network adapter is physically replaced, the IP address can be retained. Since a computer can have multiple network interface cards installed, it can also have multiple IP addresses. IP addresses are the basis for routing in **TCP/P communication networks.**

The American central registration organization **Internet Assigned Numbers Authority IANA** has taken the lead in assigning the worldwide unique IP addresses in TCP/IP networks. The national German organization DENIC (DE-NIC, NIC: National Information Center), where organizations and companies can apply for IP addresses, is currently located in Frankfurt/Main (DENIC eG, www.denic.de).

IP addresses are 32 bits long according to the IP Protocol Version 4 (IPv4). They are usually specified in **dotted decimal notation**.

Example: IP addresses in different formats:

Dotted decimal notation:	`194.129.25.100`
Binary format:	`1100 0010 1000 0001 0001 1001 0110 0100`
Hexadecimal format:	`0xC2811964`

Each group separated by a dot in the dot-decimal notation corresponds to a byte (octet) within the value range [0, ..., 255].

In version 6 (IPv6, Internet Protocol Next Generation IPng), the address width is extended from 32 bits to 128 bits (theoretically $3.40 \cdot 10^{38}$ addresses).

The Internet addresses consist of an identifier (class bits, class identifier) to identify the network class, a network address (Net ID) and a computer address (Host ID). The identifier is part of the network address.
Historically the IP addresses are divided into five classes (categories) depending on the size of the network and the "reputation of the address holder" (according to American criteria):

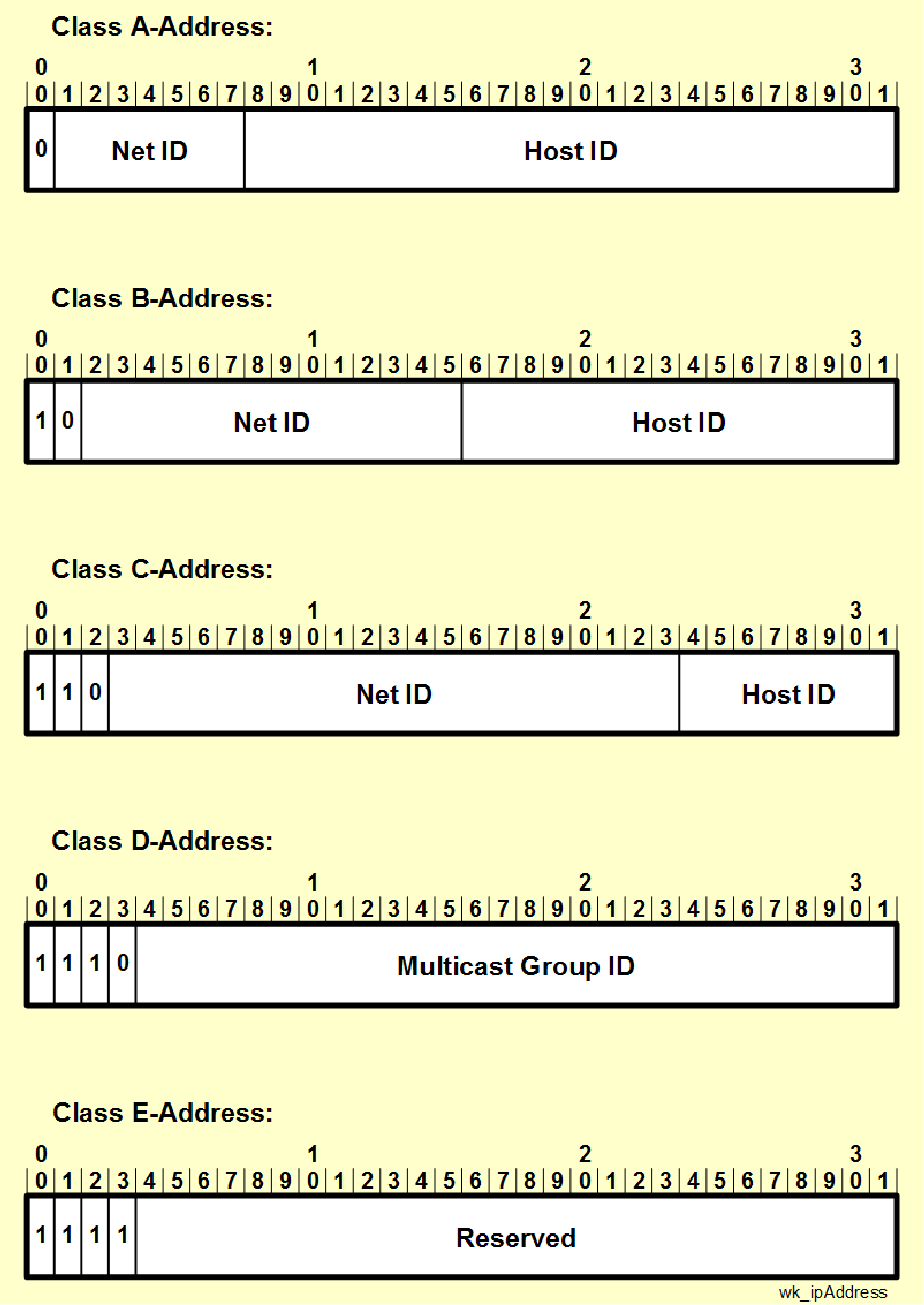


Fig. 6.2.5.1-1: Internet Protocol Address formats (IPv4)

- **Unicast Class A:**

Unicast Classes specify addresses for individual computer (hosts).

Identification:	0
Net address :	Length: 8 (7) Bit
	Number of et addresses: 2^7 = 127
Host address:	Length: 24 Bit
	Max. number of addresses: 2^{24} = 16 777 216
Address range:	0.0.0.0 ... 127.255.255.255
„private“ Addresses:	10.0.0.0 ... 10.255.255.255 (10/8)
Reservations:	0.rrr.rrr.rrr
	127.rrr.rrr.rrr

- **Unicast Class B:**

Identification:	10
Net address:	Length: 16 (14) Bit
	Number of net addresses: 2^{14} = 16384
Host address:	Length: 16 Bit
	Max. number of addresses: 2^{16} = 65536
Address range;	128.0.0.0 ... 191.255.255.255
„private“ addresses:	172.16.0.0 ... 172.31.255.255 (172.16/12)
Reservations:	128.0.rrr.rrr
	191.255.rrr.rrr

- **Unicast Class C:**

Identification:	110
Net address:	Length: 24 (21) Bit
	Max. number of addresses: 2^{21} = 2 097 152
Host address:	Length: 8 Bit
	Max. number of addresses: 2^8 = 256
Address range:	192.0.0.0 ... 223.255.255.255
„private“ addresses:	192.168.0.0 ... 192.168.255.255 (192.168/16)
Reservations:	192.0.1.rrr
	223.255.555.rrr

- **Multicast Class D:**

Multiple computers are addressed simultaneously via a multicast address.

Identification	1110
Net address:	only Multicast identification (4 Bit)
Address range:	224.0.0.0 ... 239.255.255.255
Reservations:	various reservations
Local M-addresses:	239.0.0.0 ... 239.255.255.255

- **Class E:**

The class is currently undefined; it is reserved for research purposes and later applications.

Identification:	1111
Net address:	only E-Class-Identifier (4 Bit)
Address range:	240.0.0.0 ... 255.255.255.255
special addresses :	reserved range with 28 bits width.

Some dedicated addresses are reserved for special services:

Dotted decimal address	**Significance**
0.0.0.0	Broadcast address for old SUN networks
Nr.0.0.0	Identification of a specific Class A network
Nr.Nr.0.0	Identification of a specific Class B network
Nr.Nr.Nr.0	Identification of a specific Class C network
Nr.Nr.Nr.255	Broadcast address of a specific entire network
255.255.255.255	Broadcast address of all data stations in a network
127.0.0.1	Address of the local computer *(localhost)*

Nr = any value in the range [1, 255].

The gateway address specifies the address of the computer that forwards datagrams from a network area to the outside, whose addresses are not known in the local network. According to a non-standardized convention, the gateway address is usually set equal to the decremented broadcast address.

6.2.5.2 Subnets

To further divide the address range of a network, **subnets** can be set up by the system administrator. Subnet addresses are "real" IP addresses, and the network can be divided via routers. However, the subnet addresses are only known in one local area net.

Subnets are set up by introducing **subnet masks**, which define the division into a network address part, subnetwork address part and host address part:

Net address : Subnet address : Host address.

The bits of the subnet mask representing the network portion are all set to the value one, and the bits for the host portion are all assigned the value zero.
The first and last subdivisions of each subnet have remained reserved in the past (Internet Request for Comment RFC 950); current policies also allow the use of the first possible address in a subnet.

Routing according to the Internet Protocol distinguishes between addresses in the local subnet and between addresses in other subnets. A bitwise AND operation between a node to be addressed and the local subnet address provides the local network address within the same subnet, extended by the subnet address.

Example:

Station A wants to establish a connection with communication partner B.

IP address of node A: **89.236.4.85**
Local subnet mask: **255.224.0.0**

IP address of destination node B:**89.234.85.50**

```
     0101 1001 1110 1010 0101 0101 0011 0010 (B)
AND  1111 1111 1110 0000 0000 0000 0000 0000 (S)
------------------------------------------------
     0101 1001 1110 0000 0000 0000 0000 0000
```

Nodes A and B are located in the A network 89.00.00.00 in the same subnet.

Exercises

E.6.2.5.2-1: A TCP/IP network with the network address 195.137.18.0 is to be "subnetted" into eight subnets.

a) Which network class does the network belongs to?
b) What is the subnet mask?
c) What are the broadcast addresses in the subnets called?
d) What are the gateway addresses called?
e) What is the maximum number of data stations that can be addressed in each subnet?

6.2.5.3 Classless IP addresses

Because of the ineffectiveness of assigning IP addresses if not all possible hosts in a network class are used, classless IP addressing was introduced as part of routing: **Classless Interdomain Routing CIDR**.

In the CIDR notation, the division of IP addresses into different classes is omitted. An arbitrary number of prefix bits defines the network portion of the 32-bit IP address. The remaining bits identify the hosts.

The notation in the form of the netmask is omitted. The number of prefix bits is appended to the IP address, separated by a slash. Components that are not included in the prefix, i.e. zero bytes, can be omitted from the specification.

Example:

Network of class B: `150.127.0.0` with subnet mask `255.255.0.0`

CIDR notation: `150.127.0.0/16`

alternative CIDR notation: `150.127/16`

Any substructuring is possible.

6.3 Routing

6.3.1 Definition

Routing refers to the pathfinding for the transmission of a datagram from the sender to the receiver. The routing algorithm in the TCP/IP Internet protocol only determines the path to a **target network**, but not to the actual receiver, which a datagram travels from the sender to the receiver. Special protocols are responsible for routing datagrams within a local area network: **Address Resolution Protocol ARP and Reverse Address Resolution Protocol RARP**.

6.3.2 Datagram transfer between different physical networks

Large communication networks, e.g. wide area networks (WANs), consist of a collection of meshed data stations (nodes) that are interconnected by a network of individual edges (data lines, telephone lines, satellite links, ...).
Several additional hosts can be connected to the existing nodes.

Neighboring nodes are connected by exactly one edge. Two nodes are connected if there is an edge between them or if a path via several edges and several other nodes connects the two nodes under consideration.

To characterize the data transmission between neighboring nodes, the average transmission time is often used as a quantitative quality measure. It is a function of the physical properties of the transmission medium and the current load of data traffic. If there is a high volume of data, queues in front of the nodes must be taken into account.
The task of routing is to find the best possible path for a datagram between sender and receiver with respect to an optimization condition, e.g., minimization of the transmission time.

Routing techniques can be roughly divided into two classes:

- **Non-adaptive techniques:**
  Static procedures without consideration of the current network topology (e.g., outage of edges) and the current traffic load,

- **Adaptive techniques:**
  Dynamic procedures that adapt to the current conditions of the network.

6.3.2.1 Flooding

Flooding is the simplest non-adaptive routing method. Each node resends an incoming datagram on all edges except the incoming edge. Thus, all reachable nodes in the network receive the datagram. Faulty edges and nodes are automatically bypassed. The datagrams reach the sender with the absolute shortest delay time, because the optimal path is also always chosen as a possibility.

The receiver must detect, filter out and destroy all duplicates. The main disadvantage of the method is the time exponential growth of the number of datagrams wandering in the network. To avoid permanent circling (oscillation) of datagrams in ring-shaped meshes, the lifetime is limited.

Selective flooding by considering a-priori information (e.g., restricting sending to one direction when the geographical location of the receiver is known) can help improve the method. It is sometimes used when using non-intelligent bridges (hardware coupling elements on ISO/OSI layer 2) between different local networks.

6.3.2.2 Concept of the shortest path

6.3.2.2.1 Path tables and source-sink trees

The non-adaptive **concept of "shortest path"** is a global optimization method and tries to find a "shortest" path through the network according to a given metric. Optimization criteria include the average delay time between two nodes, geographic distance, bandwidth, or transmission cost.

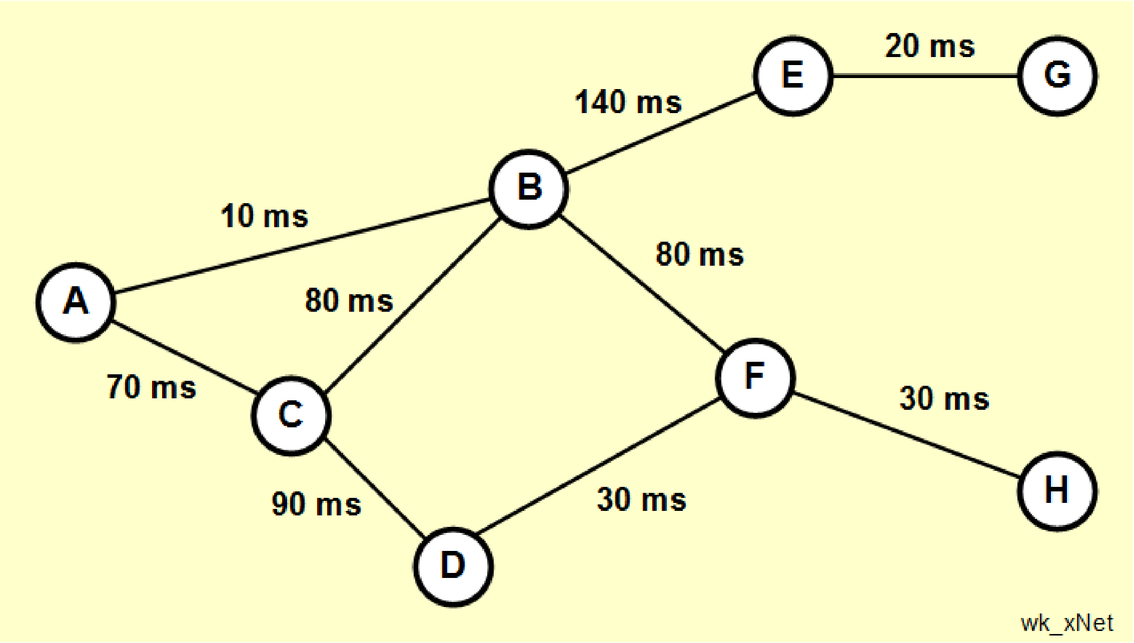
Based on a theoretical estimation, for example, of the mean transmission times based on the physical characteristics of the transmission media and estimates of the presumed mean utilization, a **"source-sink tree" (sink tree)** is created in tree structure without loops and fixed route tables for the optimized path selection for each node. The path tables are stored in the routers.

Networks with global optimization according to the shortest path principle as a routing method operate with relatively high utilization when the traffic volume is correctly estimated. Higher traffic can lead to chronic congestion. Edge failures lead to an unnecessarily large number of connection interruptions, and in borderline cases to the disintegration of the total network into several subnets.

Exercises

E.6.3.2.2.1-1: Path tables and source-sink trees

a) Create a path table for node F in the experimental network below.
b) Construct source-sink trees for nodes A and F.



6.3.2.2.2 Dijkstra’s algorithm

A well-known algorithm for determining the shortest path is the method according to the Dutch computer scientist Edsger W. Dijkstra:

Assumption: The network is considered as a weighted, undirected graph.

- Each edge is labeled with its metric-dependent "path length" relative to the initial or target node.

- At the beginning, since there is no path yet, each edge is given the path length "infinity" (programming practice: very large number).

- Starting from any source or destination node, the actual distances of a node to all neighboring nodes are determined
  and the "shortest" connection to the next node is selected as a partial path.

- The process is repeated until the complete connection from the source node to the destination node is established.

Exercises

E.6.3.2.2.2-1: Make clear the flow of Dijkstra's algorithm based on the graph on the next page. Node A serves as the starting point.

Top: Network of a weighted graph with eight nodes A to H,

Bottom: Five algorithmic steps (one step per line) to reach the first connections relative to node A.

E.6.3.2.2.2-2: Work out a software function for finding the shortest path through a given network according to Dijkstra's algorithm and test the function on a self-selected example.

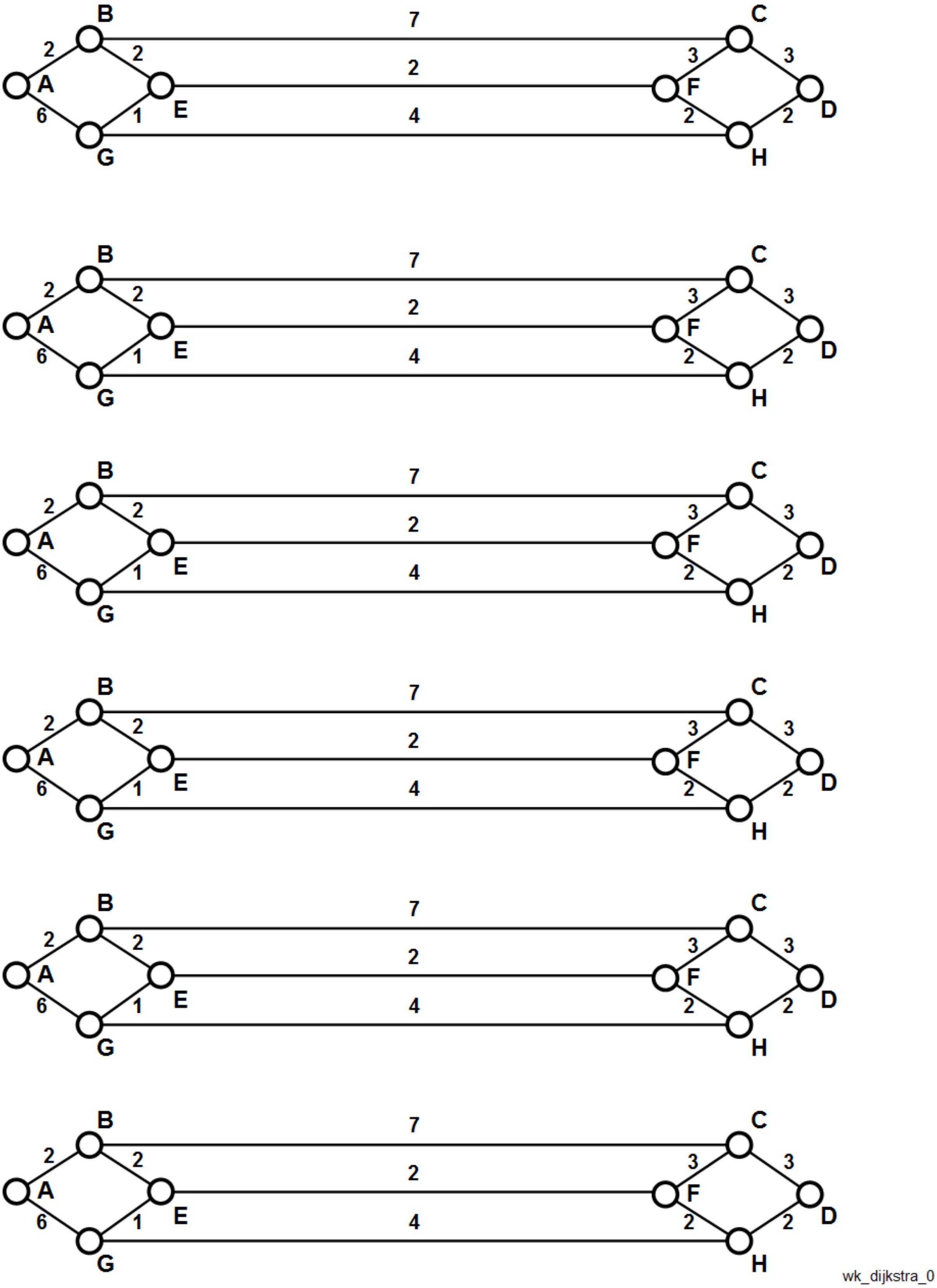


Fig. 6.3.2.2-1: Routing algorithm according to Dijkstra

6.3.2.3 Backward Learning

Backward Learning (Distance Vector Routing, distributed Bellman-Ford Routing, Ford-Fulkerson Routing) is an adaptive routing method and works with path tables. For each node there is a separate table (vector) with the current transmission times and the preferred connections to all other nodes in the network. When sending, a node always chooses the outgoing line with the currently shortest delay time.

Initially, the tables are initialized with arbitrary initial values. Updating is done by observing the incoming datagrams, which receive additional timing information from the sender. Assuming that the transfer is directionally invariant in all network connections, the transfer time can be determined directly and the path table can be updated continuously.

In case of erratic time variations, the correction is performed according to a given continuous algorithm.

The method is also used in intelligent bridges (ISO/OSI layer 2: MAC Layer Brigde) in plug-and-play solutions.

Disadvantages of the method are mainly of two kinds:

- enormous space requirement for the route tables in large networks with many nodes (~ number of connections · number of nodes),
- disconnection to more remote nodes in case of a line failure.

An improvement of the method is achieved by omitting the condition for directional invariance with respect to the transfer time. Neighboring nodes transmit the delay times they have determined to each other at periodic intervals or only in the event of unusual changes. By successively forwarding the current data, the entire network is constantly updated.

This method was originally used in the Internet, but had to be replaced due to the extreme transmission overhead involved in forwarding the time vectors in the growing network.

The current adaptive routing method on the Internet considers a set of elementary connections whose transfer times are provided to each node with a defined minimum computing power in the form of a list. The currently measured transfer times are periodically propagated throughout the network using a flooding algorithm and used to recalculate all connections. To avoid oscillations, a maximum lifetime (time to live) is defined in accordance with the Internet Protocol.

6.3.2.4 Hierarchical Routing

In large communication networks, equal treatment of all routers results in very long routing tables with large memory requirements and enormous CPU time during processing.

A remedy is the introduction of hierarchical routing by dividing the locations of the routers into different regions, e.g. in the case of geographically widely distributed networks. Several hierarchy levels can be useful (regions -> clusters -> zones -> groups).

Under certain circumstances, hierarchical routing can lead to longer path lengths.

6.3.3 Datagram transfer within a physical network

6.3.3.1 Logical names, logical addresses and hardware addresses

In data communication networks, each host has one or more network interface cards (NIC). Communication networks according to the IEEE/ISO standards use hardware addresses of different widths (MAC address, Medium Access Control Address) depending on the network technology, e.g. 48-bit wide Ethernet addresses or 16- or 48-bit wide IBM Token Ring addresses.

Unix/Linux hosts in TCP/IP networks maintain, if no Domain Name Service DNS is used, a compilation of the Internet addresses, the logical computer names and the possibly agreed alternative names (alias name) in the file **`/etc/hosts`**:

Example of the **`/etc/hosts`** file:

```
#IP address    Logical name        Alias name

127.0.0.1      localhost
192.8.3.2      Bob                 Faraday
192.8.3.12     Maud
192.8.3.24     Andy                Maxwell
192.8.3.65     Shirley
```

(Lines prefixed with a double cross (number sign) contain comments).

The logical names and the alias names are only for ease of use and simplification of operation on the network.

The `/etc/ethers` file lists the relationship between MAC hardware addresses and logical names:

`/etc/ethers` file example:

```
08:00:02:00:04:16        Bob
00:00:0C:12:18:24        Maud
08:00:20:14:72:11        Andy
08:00:5A:24:07:28        Shirley
```

Within a local area network, two protocols are involved in the conversion of Internet addresses and MAC addresses:

- **Address Resolution Protocol ARP:**
  Conversion of an Internet address to a physical hardware address,

- **Reverse Address Resolution Protocol RARP:**
  Determination of the Internet address belonging to a hardware address.

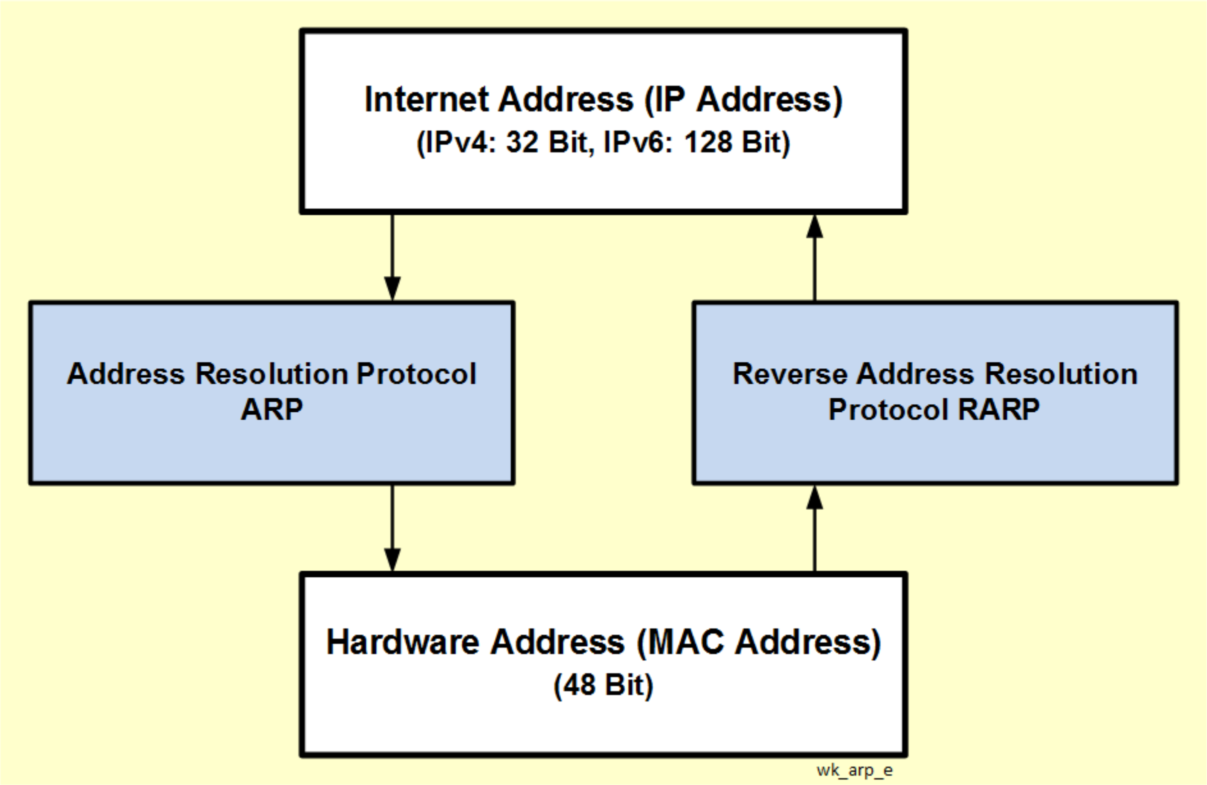

Internet Address (IP Address)
(IPv4: 32 Bit, IPv6: 128 Bit)
Address Resolution Protocol ARP
Reverse Address Resolution Protocol RARP
Hardware Address (MAC Address)
(48 Bit)
wk_arp_e

Fig. 6.3.3.1-1: Address Resolution Protocol and Reverse Address Resolution Protocol

6.3.3.2 Functionality of the Address Resolution Protocol

For direct communication between hosts on a local area network, each host maintains an **Address Resolution Protocol (ARP) cache** that contains the mappings of physical hardware addresses (MAC addresses) to logical Internet addresses for all known hosts. ARP works dynamically, detecting current changes in the network and adjusting the addresses accordingly.

During data transmission in the local network, the sender takes the MAC address of the receiver from the ARP cache and transmits the IP datagram directly to the receiver.

After a host has booted up, its ARP cache is initially empty. In the case of a data communication from host A to a host B that is not yet entered in the ARP cache, the following steps are processed:

- The ARP process in host A sends an ARP request packet with the Internet address of host B via broadcast message over the network: **ARP request**.

- The ARP processes of all other hosts in the local network receive the ARP request datagram.

- The requested recipient, Host B, recognizes its own Internet address and responds to the ARP request datagram with an ARP reply datagram: A**RP Reply**.

- Host A stores the assignment of Host B's Internet address in its ARP cache after receiving the ARP reply datagram.

- The actual data transmission from Host A to Host B can now take place.

The ARP request datagram also contains the mapping from the Internet address to the MAC address of the sender. Since there is a high probability of a later reply from the receiving station to the original sender, the receiver adds the sender information to its ARP cache after receiving the ARP datagram.

ARP datagrams are not forwarded by routers. Their propagation is limited to the local network. Bridges and switches between two different logical networks, on the other hand, forward the ARP messages.

The figure below shows the structure of an ARP datagram:

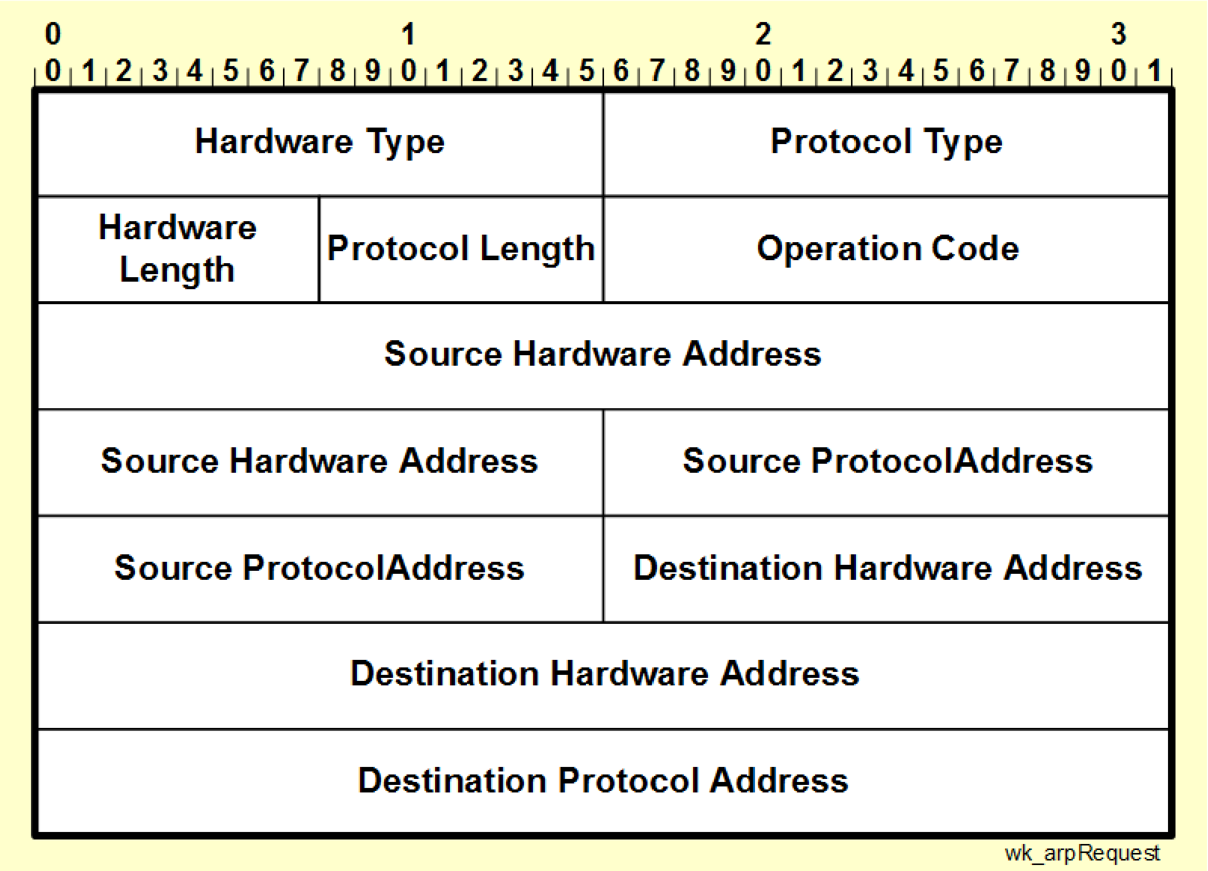


Fig. 6.3.3.2-1: Format of the ARP request frame package

ARP datagrams are 28 bytes long and contain information about network technologies and addresses:

- **Hardware type:**
  Type of network technology (1: Ethernet, 6: IEEE 802),

- **Protocol type:**
  Hardware protocol number specification (0x0800: IP),

- **Length of the physical address (Hardware length):**
  Length of the physical address in bytes (6: Ethernet and IEEE 802),

- **Length of the protocol address (Protocol length):**
  Length of the logical protocol address in bytes (4: IP),

- **Operation code:**
  Type of ARP datagram (1: ARP request, 2: ARP reply, 3: RARP request, 4: RARP reply),

- **Physical source address (Source Hardware Address):**
  MAC address of the sender,

- **Source Protocol Address:**
  Logical protocol address of the sender (IP address),

- **Physical destination address (Destination Hardware Address):**
  MAC address of the receiver,

- **Receiver Protocol Address (Destination Protocol Address):**
  Logical protocol address of the receiver (IP address).

6.3.3.3 Functionality of the Reverse Address Resolution Protocol

The Reverse Address Resolution Protocol RARP enables the logical Internet address to be determined from the hardware address (MAC address) of the link layer.

Usually it is used in hosts without their own RAM space, e.g. in drive-less workstations (discless stations) or terminals with X Window protocols (X terminals).

For the RARP protocol to work, a host on the local network must act as a RARP server that maintains a list of MAC address and Internet address mappings. Upon receiving a request from a drive-less workstation, the RARP server sends a RARP response datagram back to the sender.

6.3.4 Internetwork-Routing

The network layer of the TCP/IP protocol suite includes a number of other protocols for performing routing tasks in connected subnets within autonomous systems and on the Internet. The most important protocols are discussed below as examples:

- **Routing Information Protocol RIP:**
  - Interior Gateway Routing protocol for communication of routers within an autonomous system,
  - original routing protocol in the Internet, partly still in someuse today,
  - Basis: Distance-Vector-Routing based on the Bellman-Ford algorithm,
  - Structure of routing tables (Distance Vector Routing),
  - Exchange of information for updating routing tables every 30 - 60 s,

- **Open Shortest Path First Protocol OSPF:**
  - Routing protocol within a single autonomous system with implemented link-state algorithm,
  - Open, non-proprietary quasi-standard (RFC 1247) on the Internet,
  - developed in 1990 by IETF (Internet Engineering Task Force),
  - Transmission of so-called "link state" packets (LSPs), special datagrams to

determine the neighboring routers and the "costs", to reach them or the associated networks,
- The learned information is passed on to all known routers, not only to the nearest neighbors,
- Formation of an extended knowledge base about the general network state,
- Formation of hierarchical network structures possible: formation of domains,
- Authentication of routing information by an 8-bit password to disable misconfigured routers,

- **Exterior Gateway Protocol EGP:**
  - Possibility of common communication of routers in different autonomous systems,
  - Distance Vector Routing,
  - Implementation of a large number of routing rules for communication with different different autonomous networks,
  - possible assumption of transit services against payment,
  - Consideration of the overall network in the form of a tree topology,

- **Border Gateway Protocol BGP:**
  - Routing protocol for external gateways for communication of routers in different autonomous systems,
  - Extension of the EGP protocol (but less complex compared to the OSPF protocol),
  - Structure of the overall network by means of arbitrarily interconnected subdomains (autonomously attached networks AANs),
  - Searching for any possible path, not necessarily for the shortest connection (main goal: reachability),
  - Implementation of a large number of routing rules for communication with different autonomous networks,
  - Possible adoption of transit services in exchange for payment,

- **Hello Protocol:**
  Routing protocol with the "measurement" of distances via estimated delay times in the transmission of datagrams (instead of via the determination of hops between communicating systems),

- **Gated:**
  UNIX/Linux daemon that simultaneously implements the RIP, Hello, and EGP routing protocols to optimize small autonomous systems.

6.4 Error messages on the network layer

6.4.1 Internet Control Message Protocol

The Internet Control Message Protocol ICMP (RFC 792) is used at the network layer level for reporting error conditions and the occurrence of unforeseen conditions. ICMP messages are packaged into IP datagrams for data transmission and are transmitted by the Internet Protocol service.

The main tasks of the Internet Control Message protocol are summarized below:

- Checking for reachability of hosts and routers,
- Information about better transmission paths to hosts through a router,
- Notification in case of data flooding by hosts and routers,
- Message about receiving corrupted datagrams,
- Information about receiving datagrams in case of exceeded time,
- Notification about incorrect parameter specifications in the datagram header,
- Exchange of messages about synchronization of timers,
- Information about the formats of the used Internet addresses.

6.4.2 ICMP format and message types

6.4.2.1 IP header and ICMP header

When sending ICMP messages, the "Type of Services" field in the Internet Protocol (IP) header is set to "zero" (0) and the "Protocol" field is set to "one" (1).

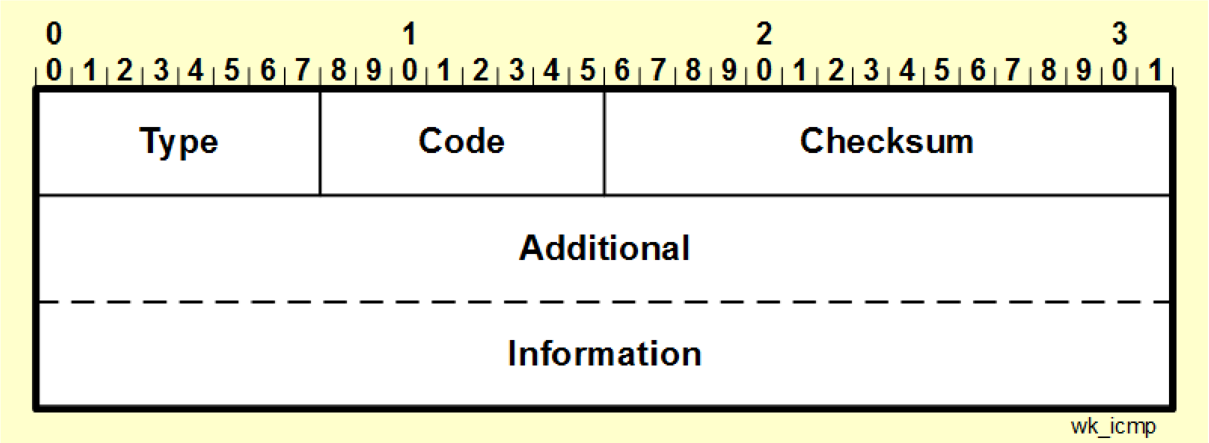


Fig. 6.4.2.1-1: ICMP packet format

Internet Control Message messages (ICMP messages) consist of different numbers of data fields depending on the type of message. The first three data fields are the same for all ICMP messages:

- **ICMP message type (Type):**
  Code identifying the type of ICMP messages possible:

Type	message	Type	message
0	Echo reply	12	Parameter problem
3	Destination unreachable	13	Timestamp request
4	Source quench	14	Timestamp reply
5	Redirect	15	Information request
8	Echo request	16	Information reply
11	Time exceeded	17	Address mask request
		18	Address mask reply

- **ICMP code (code):**
  Code number depending on the current problem, e.g. network or host unreachable, fragmentation without fragmentation bit set, etc.

- **ICMP checksum:**
  Checksum for error detection based on the entire contents of the ICMP message (checksum field itself is not considered in the calculation).

The different ICMP messages can be divided into two categories:

- ICMP error messages,
- ICMP request messages.

6.4.2.2 Error messages

ICMP error messages are sent under the following common special conditions:

- Receiver not available,
- Transfer time exceeded (TTL = 0),
- Invalid parameters in an IP header,
- Interrupt request to the sender in case of data flooding,
- Route redirection.

As network traffic grows, the need for ICMP messages increases considerably. To protect the communication network from data flooding with ICMP messages, the following basic rules apply:

- ICMP messages to datagrams regarding routing and passing ICMP messages are not generated.

- The generation of ICMP messages during the transmission of IP datagrams in multicast frames is completely omitted.

- ICMP messages about IP datagram segments may be generated at most for the first segment.

6.4.2.3 Request messages

ICMP request messages enable requests regarding various network information and responses to other ICMP request messages.

Frequently used ICMP request messages are

- **Echo request and response:**
  - Connection verification with the ping command
    (Ping: Packet InterNet Groper),
  - Replying to the echo request by sending back an echo reply to the originator,
  - Calculation of the transfer time since the emission of the original echo
    request at the arrival of the echo response,
  - ICMP message types: 8 = echo request, 0 = echo response,

  Example:

```
$ ping 195.18.144.26
PING skydiver (195.18.144.26): 56 data bytes
64 bytes from skydiver (195.18.144.26): icmp_seq=0 ttl=16 time=10 ms
64 bytes from skydiver (195.18.144.26): icmp_seq=1 ttl=16 time=10 ms
64 bytes from skydiver (195.18.144.26): icmp_seq=2 ttl=16 time=10 ms
64 bytes from skydiver (195.18.144.26): icmp_seq=3 ttl=16 time=10 ms
^C
```

- **Time stamp request and response:**
  - Request for a time stamp with information about the current date and time,
  - Time stamp: 32-bit information indicating the elapsed seconds since
    January 1, 1900,
  - ICMP message types: 13 = request, 14 = response,

- **Subnet mask request and response:**
  - Existence of a subnet mask server in the local network required,
  - Subnet mask request from a host to the subnet mask server to determine
    its own subnet mask,
  - ICMP message types: 17 = request, 18 = response.

6.5 Internet Protocol Version 6

Because of the enormous growth of the Internet in the last decade of the twentieth century and the expanded requirements in terms of speed, service offerings, and data security (authentication, encryption) the Internet Protocol version 4 (IPv4) is currently being replaced by version 6 (IPv6, RFC 1883, RFC 1887), which has been developed by the IETF (Internet Engineering Task Force) since 1990 in an evolutionary manner.
IPv5 was used as an experimental protocol in 1993 at the time IPv6 was adopted.

6.5.1 Addresses

The new IPv6 protocol expands the address space from four bytes (IPv4) to 16 bytes (IPv6) and thus makes it possible to address $2^{128} = 3.4028 \cdot 10^{38}$ subscribers in principle. The notation is in eight groups of 16-bit long integer values, each separated by a colon, in hexadecimal notation.

Example:

```
7000:0000:0000:0000:0246:9876:6ACD:FF5E
```

Various simplifications can be used:

- Leading zeros in a group can be omitted.

- Groups with consecutive zeros can be replaced by a double colon "::".
  A double colon may appear only once in the address specification.

- IPv4 addresses can still be written as two colons followed by the old dot-decimal notation.

Example:

IPv4: `193.18.5.24` *IPv6:* `::C1.12.05.18`

The protocol distinguishes between three basic address types:

- **Unicast addresses:**
  Identification of a single interface of an end station or router,

- **Multicast addresses:**
- Identifying a group of terminals for group communication,

- **Anycast addresses:**
  Identification of a set of interfaces typically belonging to different intermediate systems.

6.5.2 Header

The header has been simplified to improve network throughput compared to the IPv4 version and contains only seven fields (compared to 13 fields previously), each with a fixed length.

The IPv6 base header is used to transmit data without optional information. It has the following structure:

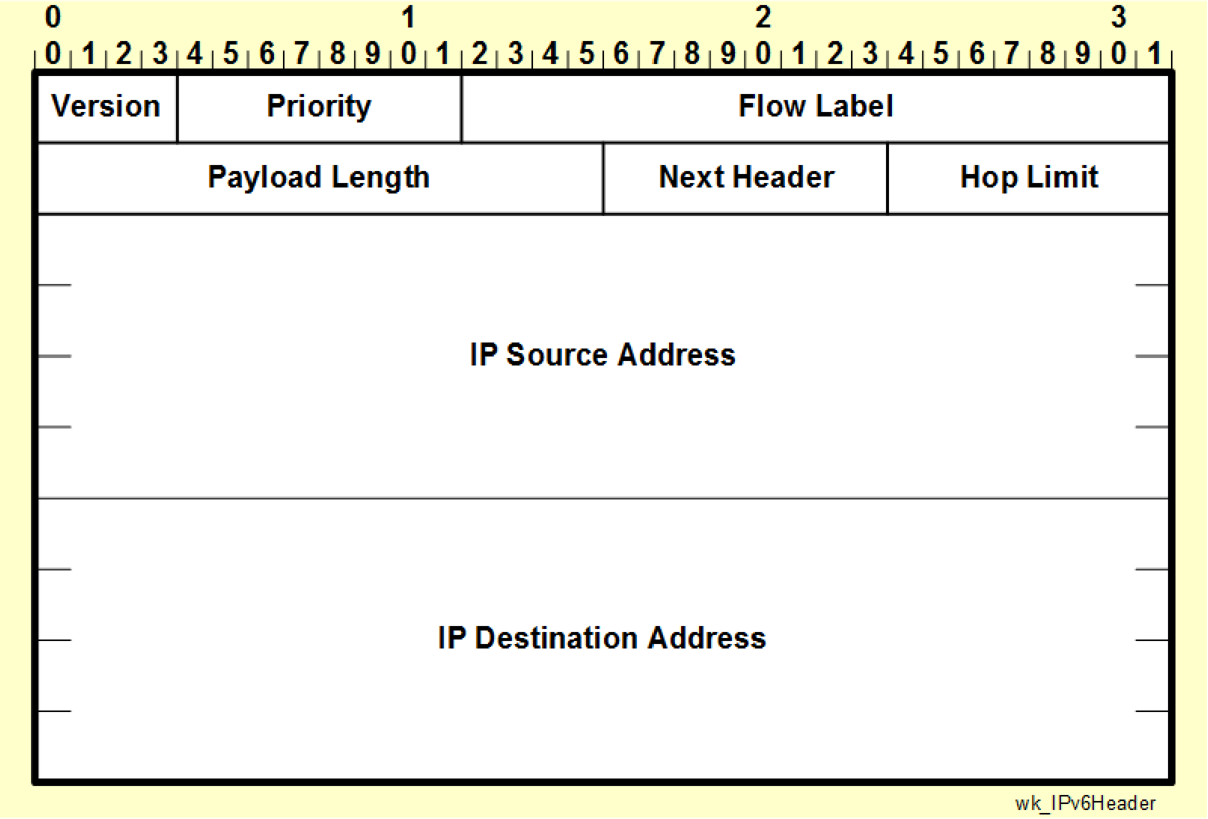


Fig. 6.5.2-1: Internet Protocol Header of the Internet Protocol Version 6 (IPv6)

The following list explains the contents of the various fields:

- **Version (version number):**
  Identification of the current Internet protocol: 4 for IPv4, 6 for IPv6,

- **Priority:**
  Datagram priority:
  0: no priority,
  1: background traffic,
  2: unattended traffic,
  4: attended bulk transfer,
  6: interactive traffic,
  7: control traffic,

- **Flow Label:**
  - Identification of different data flow characteristics,
  - Currently still of experimental character,

- **Payload Length (packet size):**
  Indication of the total size of the IP packet (including the IP header data),

- **Next Header:**
  Reference to subsequent header:
  0: IP information, 6: TCP information, 43: Routing information,
  58: ICMP information,

- **Hop Limit (maximum number of router passes):**
  - Specification of the maximum number of hops (hops between two routers) during transmission,
  - Each router decrements Hop Limit,
  - Deleting the data packet when the value reaches zero,

- **IP Source Address:**
  128-bit wide sender address,

- **IP Destination Address:**
  128-bit wide receiver address.

The header of version 6 has some serious differences compared to the one in version 4:

- Because of the fixed length of eight fields, the length indicator (IHL) becomes obsolete.

- The protocol field (Protocol in IPv4) is replaced by the Next Header field.

- There is no fragmentation information because all hosts and routers must support packets with the length of 576 bytes. If packets that are too large are used, the router sends back an error message requesting the sender to fragment all future packets to the receiver.
  Fragmentation by the router is completely eliminated for efficiency.

- Checksums are no longer calculated at the network layer, since the link layer and the transport layer provide their own backup mechanisms.

All optional information is stored in extension headers:

- **Hop-by-hop information:**
  Partial route description,

- **Destination Option Header 1:**
  optional receiver information, tunneling options for intermediate stations,

- **Routing header:**
  further specifications for routing,

- **Fragment header:**
  definition of fragmentation rules,

- **Authentication header:**
  authentication of the communication participants,

- **Destination Option Header 2 (Encapsulation-Security-Payload Header):**
  Information about data encapsulation, security and encryption,

- …