TCP/IP – part 1
Introduction

Last modification date: 25.03.2020

Introduction

Selected characteristics of communication networks:

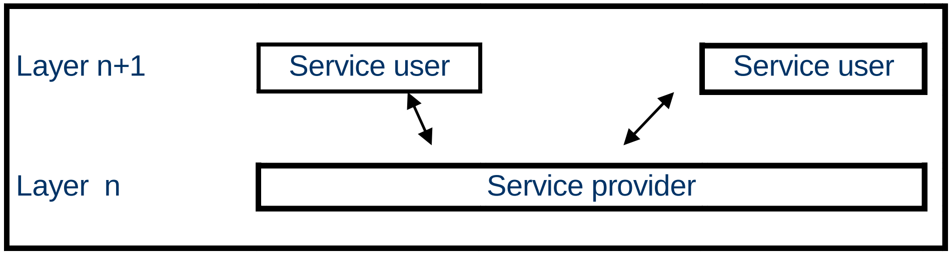
- Communication medium: cable, coaxial cable, twisted pair, fiber, electromagnetic (radio) waves
- Connection topology: bus, ring, star, tree, lattice
- Transmission method: simplex, half-duplex, full-duplex;
- Channel access: central polling, token passing, CSMA/CD, CSMA/CA
- scope: WAN, MAN, LAN.

- All the above is of secondary importance for this course. We will concentrate on fundamentals of high-level organization and network-based communication built around TCP/IP protocol suite. For detailed information on network operation and administration – enroll a network communication and/or data transmission course.

OSI communication architecture

- The **Open Systems Interconnection ((OSI) model** is a conceptual model of a communication system without regard to their underlying internal structure and technology..
- The model partitions communication system in 7 abstraction layers. A layer serves the layer above it and is served by the layer below it.

7	Application layer
6	Presentation l.
5	Session layer
4	Transport layer
3	Network layer
2	Data link layer
1	Physical layer


Layer n+1

Service user

Service user

Layer n

Service provider

- Virtual communication between corresponding layers

System A		System B
application l.	<--- layer 7 prot. ---->	application l
presentation l.	<--- layer 6 prot. ---->	presentation l.
session l.	<--- layer 5 prot. ---->	session l.
transport l.	<--- layer 4 prot. ---->	transport l.
network l.	<--- layer 3 prot. ---->	network l.
data-link l.	<--- layer 2 prot. ---->	data-link l.
physical l.	<== layer 1 prot. ==>	physical l.
	Physical connection	

Layer N at the target computer system receives exactly the same message which was sent by Layer N of the sender computer system.

OSI reference model

The communication network is partitioned into the following layers:

- **Layer 1: Physical layer** – handles the mechanical and electrical details of the physical transmission of a bit stream
- **Layer 2: Data-link layer** – handles the *frames*, or fixed-length parts of packets, including any error detection and recovery that occurred in the physical layer
- **Layer 3: Network layer** – provides connections and routes packets in the communication network, including handling the address of outgoing packets, decoding the address of incoming packets, and maintaining routing information for proper response to changing load levels

Layered OSI model (Cont.)

- **Layer 4: Transport layer** – responsible for low-level network access and for message transfer between clients, including partitioning messages into packets, maintaining packet order, controlling flow, and generating physical addresses
- **Layer 5: Session layer** – implements sessions, or process-to-process communications protocols
- **Layer 6: Presentation layer** – resolves the differences in formats among the various sites in the network, including character conversions, and half duplex/full duplex (echoing)
- **Layer 7: Application layer** – interacts directly with the users, deals with file transfer, remote-login protocols and electronic mail, as well as schemas for distributed databases

Internet

- **Internetwork** consists of a set of local networks connected via routers. Internetworks are built to provide uniform services over a set of heterogeneous computer systems..
- **Internet** – the world-wide internetwork based on a family of TCP/IP protocols.
- **Architecture** of TCP/IP protocol family communication is not compatible with the OSI reference model OSI (RFC1122).
- The Internet Engineering Task Force (IETF) produces high quality, relevant technical documents to make the Internet work better.
- **IETF Standards** (Proposed, Draft or Internet Standards (RFC1610) are made public as so called **Requests For Comments** (**RFC**), see: https://www.rfc-editor.org/retrieve/,

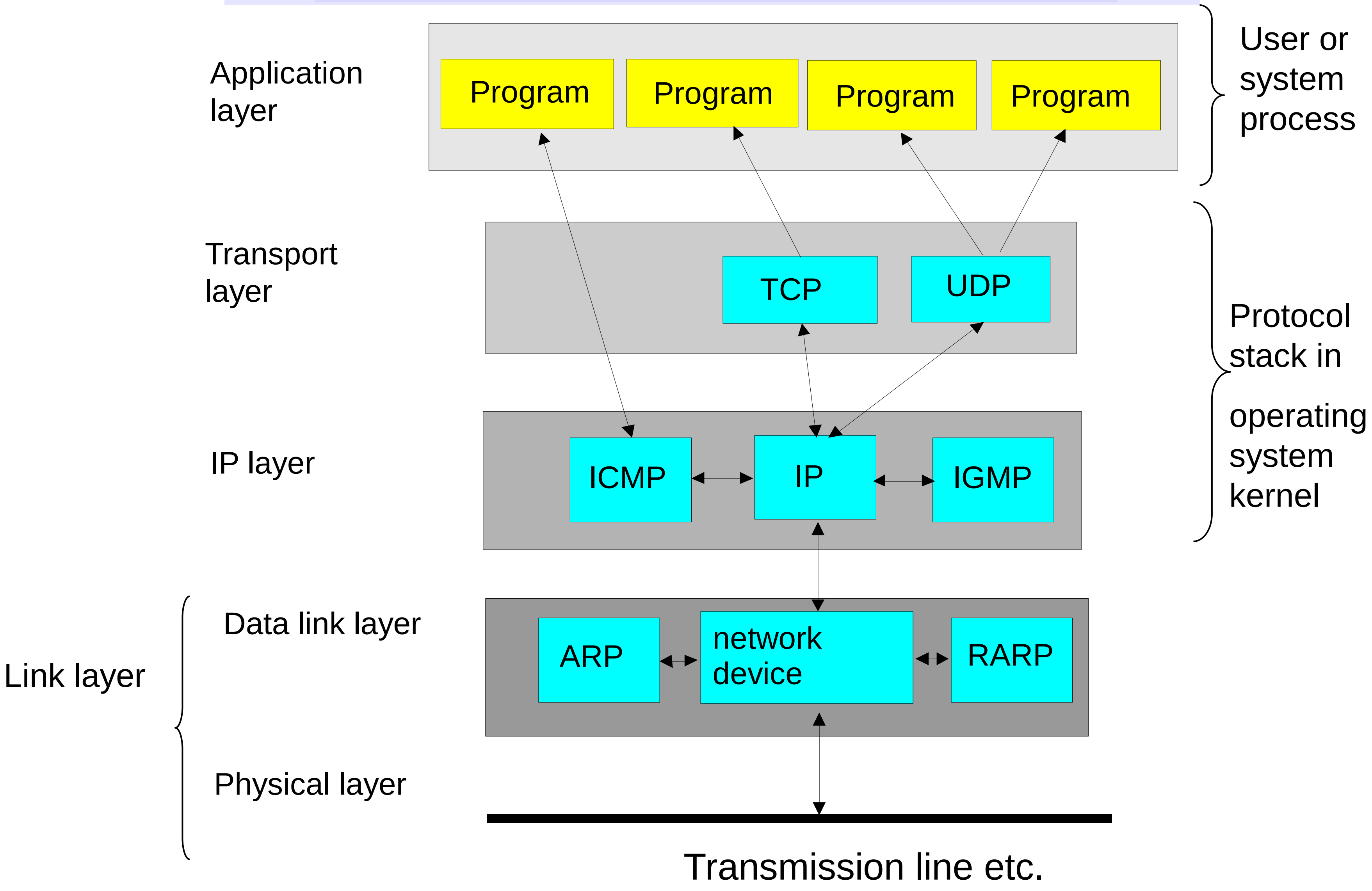
  http://tools.ietf.org/html/

Layered TCP/IP communication model (DoD)

wg RFC 1122

4	Application layer
3	Transport layer
2	IP layer
1	Link layer

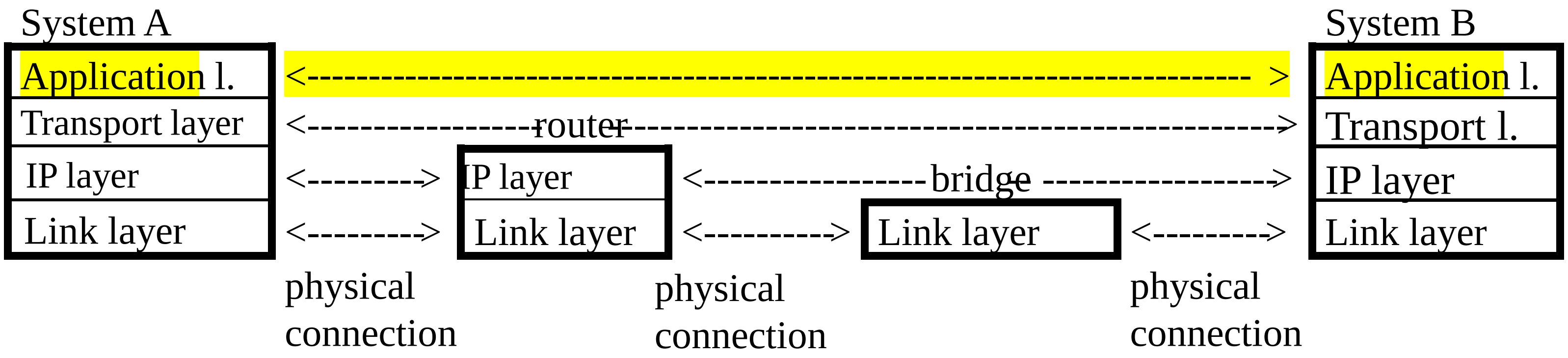
TCP/IP network model



Example: TCP/IP

- The transmission of a network packet between hosts on an Ethernet network
- Every host has a unique IP address and a corresponding Ethernet **Media Access Control** (**MAC**) address
- Communication requires both addresses
- **Domain Name Service** (**DNS**) can be used to acquire IP addresses
- **Address Resolution Protocol** (**ARP**) is used to map MAC addresses to IP addresses
    - **Broadcast** to all other systems on the Ethernet network
- If the hosts are on the same network, ARP can be used
    - If the hosts are on different networks, the sending host will send the packet to a router which routes the packet to the destination network

Communication in TCP/IP network



System A

Application l.

Transport layer

IP layer

Link layer

<--- >

<-----------------router-->

<---------> IP layer <-----------------bridge ------------------>

<---------> Link layer <-----------> Link layer <----------->

physical connection

physical connection

physical connection

System B

Application l.

Transport l.

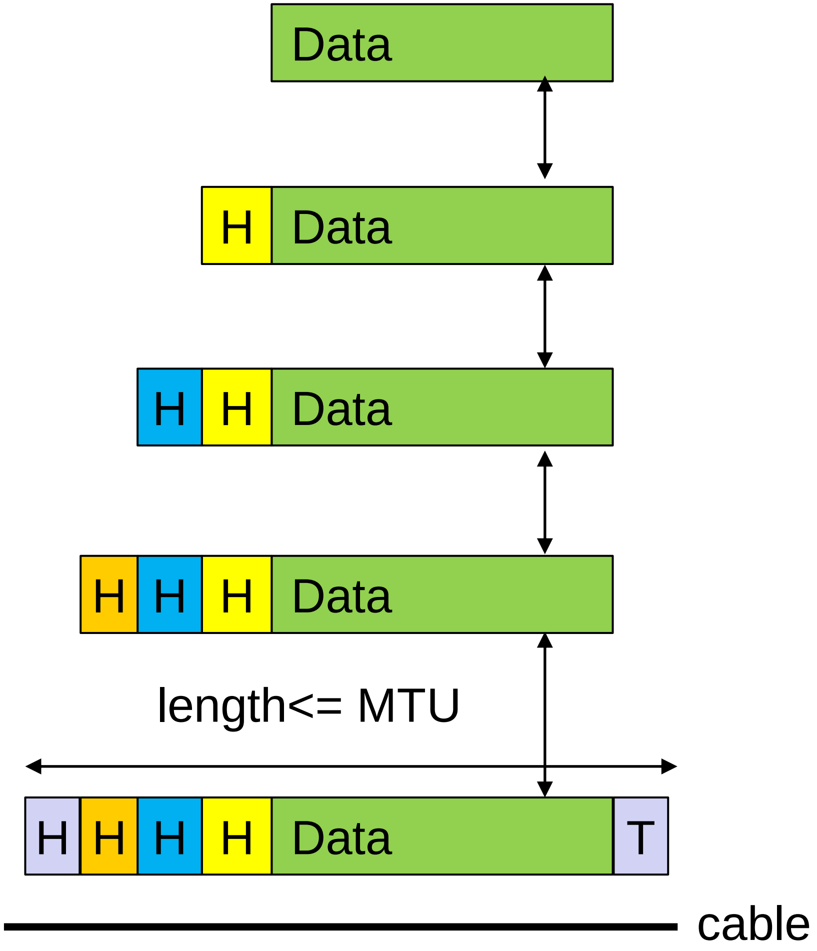
IP layer

Link layer

- Example protocol stack

Layer nr	Layer	Protocol
4	application	HTTP
3	transport	TCP
2	Internet (IP)	IP
1	link	Ethernet 802.3u

Sending data in TCP/IP net (+Ethernet)



Application layer

Transport layer

IP layer

Ethernet

Encapsulation:

MTU for Ethernet: 1500, for Ethernetu 802.3: 1492 octets (wg RFC 1122)

Example–TFTP

- Example of encapsulation - Trivial File Transfer Protocol (RFC1350)

Ethernet	Ethernet	IP	UDP	TFTP	data	Ethernet
8	14	20	8	4		4

<-----------------------Ethernet frame (72 - 1526 octets) --------------------------->

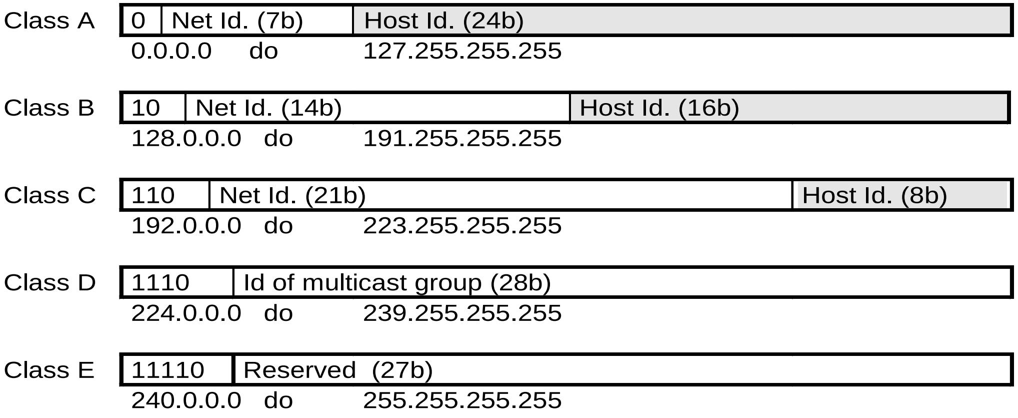
Types of data transmissions/routing schemes

- **Unicast** – separate copy of data is sent from a source to each recipient which requests it (one-to-one association between destination address and network endpoint)
- **Multicast** – a single data copy is sent to a group of recipients (one-to-unique many association)
- **Broadcast** – a single data copy is sent to all recipients of the same network segment to which the sender belongs (one-to-many association)
- **Anycast** – a single data copy is sent to a single member of a group of potential receivers that are all identified by the same destination address (topologically nearest)
- **Geocast** refers to the delivery of data to a group of destinations in a network identified by their geographical locations. It is a specialized form of Multicast addressing used by some routing protocols for mobile ad hoc networks.

Classful TCP/IP addressing

IPv4 address classes →
Class D: multicast
Class E: restricted use

Since 1970s IP addresses (domains, network protocols) Internetu) were managed by Internet Assigned Numbers Authority (IANA), currently witihin the Internet Corporation for Assigned Names and Numbers (ICANN)



Class A: 0 | Net Id. (7b) | Host Id. (24b)
0.0.0.0 do 127.255.255.255

Class B: 10 | Net Id. (14b) | Host Id. (16b)
128.0.0.0 do 191.255.255.255

Class C: 110 | Net Id. (21b) | Host Id. (8b)
192.0.0.0 do 223.255.255.255

Class D: 1110 | Id of multicast group (28b)
224.0.0.0 do 239.255.255.255

Class E: 11110 | Reserved (27b)
240.0.0.0 do 255.255.255.255

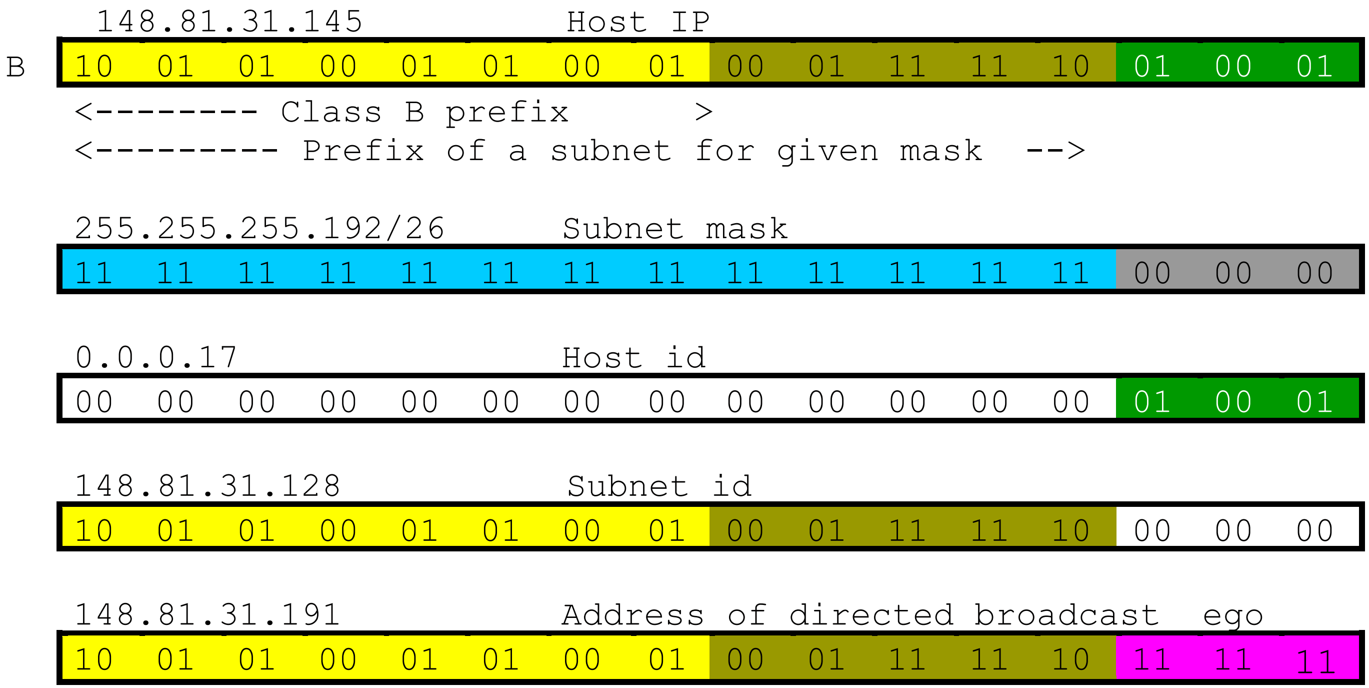
Special IP addresses

- Prefix>0 & host idi=0 => (sub)network address
- Prefix>0 & host id=all ones (bits) => address of **directed broadcast** (all host of the specified network)
- Adres 255.255.255.255 => adres **restricted broadcast** (all hosts in the local network)
- Adres 0.0.0.0 => the sender does not know its IP
- Prefix 127/8 : **loopback** addresses (packets do not leave the computer); 127.0.0.1 ←→ localhost
- 224.0.0.0-224.0.0.255 – used for network topology detection and management. Packets are not routed. .
- Address blocks of local (non-routed) networks (RFC1918):
  - 10.0.0.0 - 10.255.255.255 (1 klasa A; prefix 10/8)
  - 172.16.0.0 - 172.31.255.255 (16 klas A; prefix 172.16/12)
  - 192.168.0.0 - 192.168.255.255 (256 klas C; prefix 192.168/16)

Classless Inter-Domain Routing

Split of the address into net and host id is made with address mask.

Example.



```
   148.81.31.145            Host IP
B  10 01 01 00 01 01 00 01 00 01 11 11 10 01 00 01
   <-------- Class B prefix      >
   <--------- Prefix of a subnet for given mask  -->

   255.255.255.192/26       Subnet mask
   11 11 11 11 11 11 11 11 11 11 11 11 11 00 00 00

   0.0.0.17                 Host id
   00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 01

   148.81.31.128            Subnet id
   10 01 01 00 01 01 00 01 00 01 11 11 10 00 00 00

   148.81.31.191            Address of directed broadcast  ego
   10 01 01 00 01 01 00 01 00 01 11 11 10 11 11 11
```

CIDR: Classless InterDomain Routing (RFC1519)

Example mask= CIDR / 26 can be written as dot addressi: 255.255.255.192, tzn. address prefix (subnet id) is 26 bit long, and suffix: 6-bit long.

Port numbers

- Communication endpoints are addressed as **IP_nr:port_nr** , where **IP_nr** is an IP number assigned to internet interface of the host (IPv4: 32b) and **port_nr** is an integer port number (IPv4: 16b), which selects the endpoint from a set of all endpoints of that host
- When a single octet is sent serially, and the octet represents an integer number– the first bit has the largest weight e.g. number 170 is represented as follows ----------------------------->

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|1 0 1 0 1 0 1 0|
+-+-+-+-+-+-+-+-+
```

- When multibit number is transmitted, the largest weight digit is sent in the first bit of the first byte. This ordering of bit importance is known as **network order** (or *Big Endian*).
- **Internet Assigned Numbers Authority** (IANA) coordinates assignment of standard port numbers to TCP/IP (RFC1700).
- IANA defined well-known ports (system ports), they have numbers: 1-1023. ---->
- **Ports registered** but not managed by IANA (user ports) are in range: 1024-49151, while **dynamic (ephemeral) ports** in range: 49152-65535.

.

protokół	port(y)
ECHO	7
FTP	20,21
SSH	22
TELNET	23
DNS	53
HTTP	80
POP3	110
IMAP	143
HTTPS	443
X11	6000-6007

Basic TCP/IP protocols

- **Internet Protocol** (IP) implements two basic functions (RFC791)
  - addressing
  - fragmentation of data packets (**datagrams**)
- IP defines format of data packets (datagrams), formulates rules of forwarding datagrams between networks and rules which define how routers and hosts should treat packets when error message should be generated and when packets can be discarded.
- It does not solve problems related to:i
  - Packet duplication
  - Delayed delivery or not-in-order delivery of packets
  - Data corruption
  - Data loss
- TCP/IP stack has to treat correctly IP packets of length: **576** up too **65535 octets**
- For a physical network with small Maximum Transfer Unit (MTU) it is necessary to **fragment** packets on entry and **merge** fragments on the receiver side
- Datagram field ``Time To Live'' (TTL) enables **removal** of undelivered package after TTL hops (passes through a router); the sender is sent an error message via ICMP
- One can request routing method (rigid, free) and also memorizing of the route within the package (if routers support that).

Basic TCP/IP protocols: IP

IP datagram structure (RFC791)

- IHL*4=header length, max.15*4=60B
- Type of service:
  - 3b priority
  - 4b type
  - 1b set to 0
- Identification is used with fragmentation de-fragmentation
- Flags
  - DF : don't fragment
  - MF : more fragments, =0 when no (more) fragments
- TTL – time to live count (0-255); each router hop decrements the counter
- Protocols (RFC1700):
  - 1: ICMPv4
  - 2: IGMPv4
  - 6: TCP
  - 17 UDP

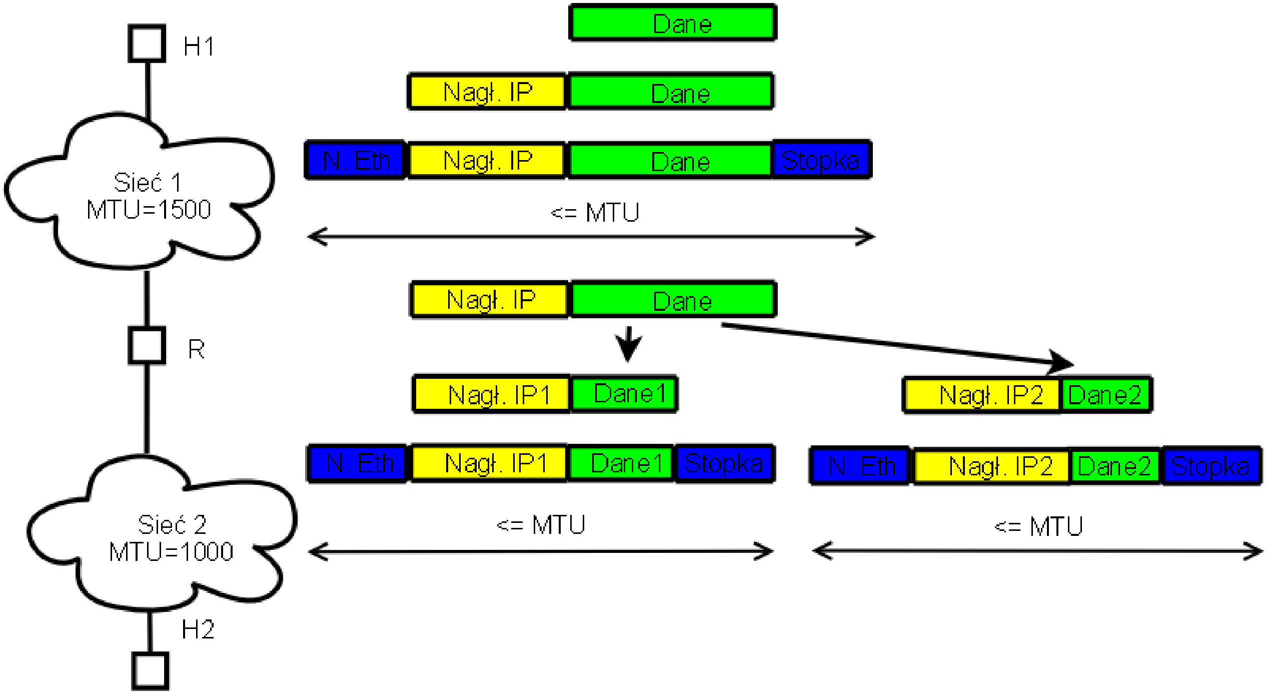
```
 0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |Version|  IHL  |Type of Service|          Total Length         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |         Identification        |Flags|      Fragment Offset    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Time to Live |    Protocol   |         Header Checksum       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                       Source Address                          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Destination Address                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Options                    |    Padding    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
      Options
CLASS NUMBER LENGTH DESCRIPTION
----- ------ ------ -----------
  0     0      -    End of Option list.  This option occupies only
                    1 octet; it has no length octet.
  0     1      -    No Operation.  This option occupies only 1
                    octet; it has no length octet.
  0     2     11    Security.  Used to carry Security,
                    Compartmentation, User Group (TCC), and
                    Handling Restriction Codes compatible with DOD
                    requirements.
  0     3     var.  Loose Source Routing.  Used to route the
                    internet datagram based on information
                    supplied by the source.
  0     9     var.  Strict Source Routing.  Used to route the
                    internet datagram based on information
                    supplied by the source.
  0     7     var.  Record Route.  Used to trace the route an
                    internet datagram takes.
  0     8      4    Stream ID.  Used to carry the stream
                    identifier.
  2     4     var.  Internet Timestamp.
```

Note: options can be get and set using socket interface functions: **getsockopt**, **setsockopt**.

IP: fragmentation

- Each network technology determines maximum packet length (MTU).
- Router, which received a packet which is longer from the MTU of the next route segment has to split the packet into several smaller ones (RFC791).
- Each fragment contains a piece of the original data area and modified header (MF, offset).



- To assemble the fragments of an internet datagram, an internet protocol module (for example at a destination host) combines internet datagrams that all have the same value for the four fields: identification, source, destination, and protocol, taking into account offset field.
- Fragments of IP packets can be fragmented again (and again …) Miimum MTU=68 octets.
- Every internet destination must be able to receive a datagram of 576 octets either in one piece or in fragments to be reassembled.

Basic TCP/IP protocols: ICMP

ICMP protocol (RFC792) is typically used to report (to sender) error messages in processing of IP packets.

Most used error messages:

- datagram cannot reach its destination,
- the gateway does not have the buffering capacity to forward a datagram,
- the gateway can direct the host to send traffic on a shorter route
- time-to-live reached 0 and the packet was dropped

Other messages

- Slowing down the sender (flow control)
- Echo request and echo response
- Time request, time response
- Incorrect parameters of an IP packet

Basic TCP/IP protocols: UDP

- **UDP** protocol (RFC768) provides connectionless transport, because it does not require any long-term relationship between sender and receiver hosts.
- Structure of the UDP datagram:

```
 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Source port(or 0)           |        Destination port       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Length              |       Checksum                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   up to 65,515 data octets ...                                |
|   ......                                                      |
```

- Checksum is the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header, and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets (see RFC768).

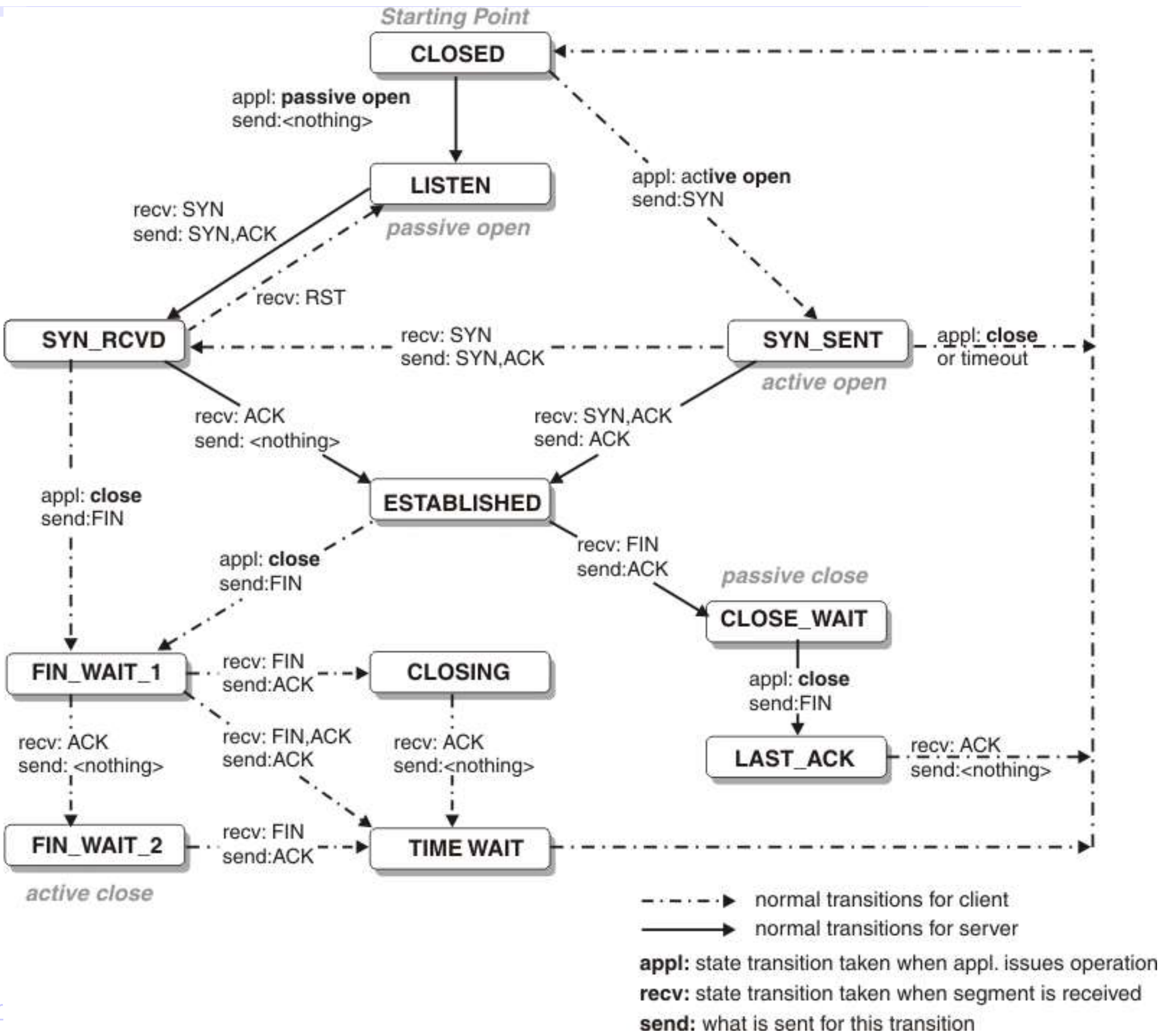
Basic TCP/IP protocols: TCP

- **TCP** protocol (RFC793, RFC1122) provides:
  - Duplex, connection-oriented, end-to-end reliable transport protocol
  - Flow control
- Data stream is created by sending TCP segments :

```
0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |          Source Port          |       Destination Port        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        Sequence Number                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Acknowledgment Number                      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Data |           |U|A|P|R|S|F|                               |
 | Offset| Reserved  |R|C|S|S|Y|I|            Window             |
 |       |           |G|K|H|T|N|N|                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |           Checksum            |         Urgent Pointer        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Options                    |    Padding    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                             data                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- For end-to-end reliability TCP uses :
  - Checksum for each TCP segment
  - Retransmission of segments lost (not confirmed)
  - Adaptation of sending speed to actual transport delay and receiver reception speed

TCP state transition diagram (*)



From IBM Knowledge Center

TCP: buffers, flow of control, windows

- TCP software allocates a receiving buffer, then it sends its size to the other side of communication.
- While data arrive the recipient sends back acknowledgment and information about free space in the receiving buffer (so called window size)..
- Sender which received zero window size offer has to stop sending data – until the recipient offers positive window size.
- If TCP detects segment loss – it sends one segment and waits for acknowledgment. When the acknowledgment arrives 2 segments are sent and the sender waits for acknowledgment, and so. Altogether an exponential increase of the number of segments sent ahead of acknowledgment can be seen – until half of the offered window size is filled.
- By default TCP software collects in the sending buffer data fragments to be sent (Nagle algorithm RFC896) . When PSH flag of TCP segment is set – buffer content is sent out immediately.
- When an application sets socket flag URG it is possible to store in TCP segment 1 “urgent byte” to the output buffer; In the TCP segment the flag URG=1, and urgent pointer points at the data position just after the inserted byte.

Auxiliary TCP/IP protocols (*)

- Address resolution protocol **ARP** (RFC826) defines 2 basic requests:
  - Address request (sent to the broadcast address)
  - Answer to the address request which is sent to the requester
- Reverse address resolution protocol **RARP** (RFC903) is used by a computer to get its IP address, given its MAC (physical) address
- Boot protocol **BOOTP** (RFC951), allows a diskless client machine to discover its own IP address, the address of a server host, and the name of a file to be loaded into memory and executed. BOOTP packet is sent by UDP datagram with sender address: all zero bits, receiver address: all 1 bits.
- **DHCP** protocol (RFC2131) is based on the Bootstrap Protocol (BOOTP), adding the capability of automatic allocation of reusable network addresses and additional configuration options.

IPv6

- In 1998 IETF defined IPv6 standard (RFC2460); it is to replace commonly used IPv4 standard.
- Points of novelty
  - 128-bit addresses (provide 7.9e28 times larger address space than IPv4)

  Notation: 8 groups (separated with ‘**:**’) of 4 hexadecimal numbers, eg.:

  1234:5678:0000:0000:0000:0000:9ABC:DEF0. Short form:

  1234:5678::9ABC:DEF0

  - Maximum data packet size: (2^{16}-1) instead of (2^{32}-1)
  - New packet format → lower cost of header processing by routers
  - Flexibility (extra headers)
  - Resource reservation possible (guarantee of throuput and latency)
  - Authentication, IPSEC security
  - Simplification of multicast transmission, address ranges for multicast.