Програмиране в UNIX среда

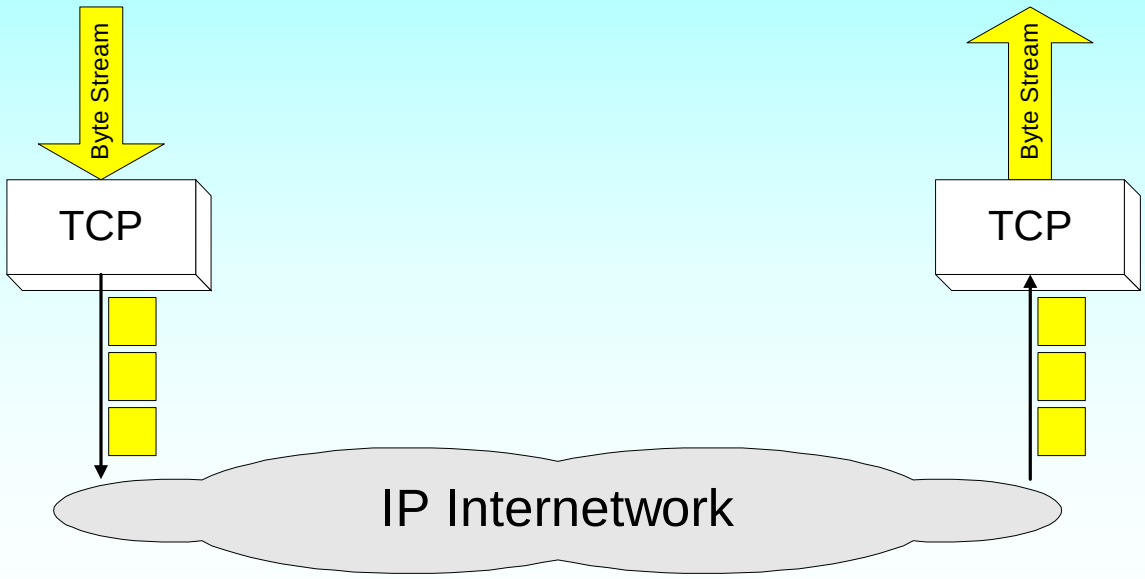
Основи на системната администрация

Мрежи

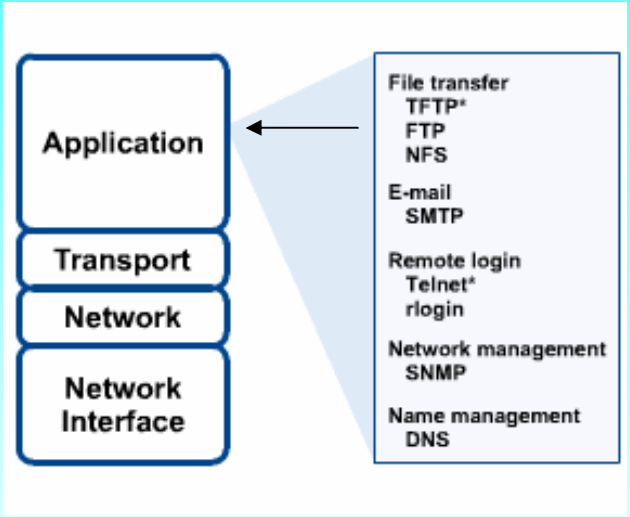
TCP

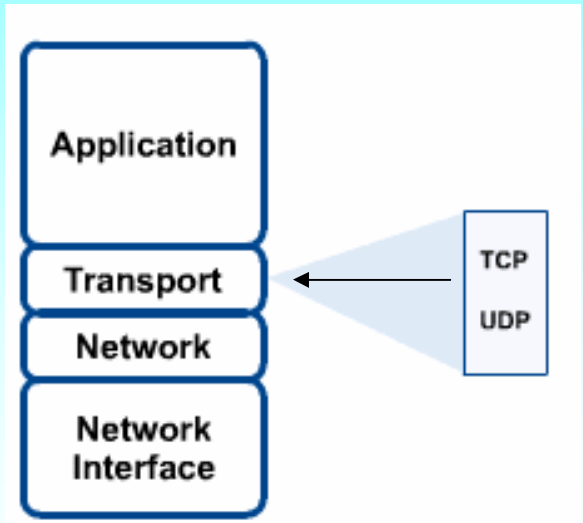
Overview

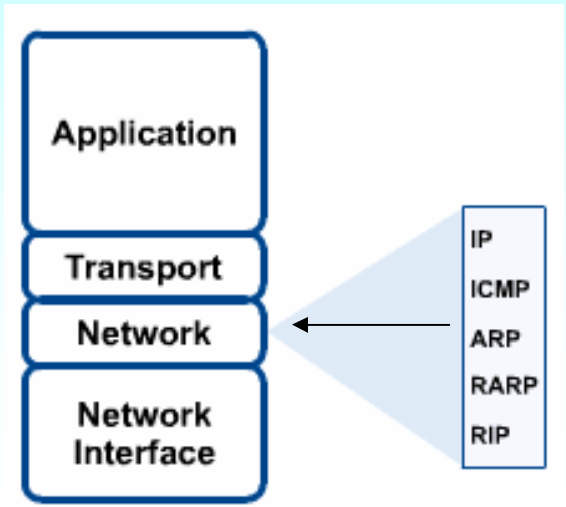
- **TCP = Transmission Control Protocol**
- TCP is a connection-oriented protocol that provides a reliable unicast end-to-end byte stream over an unreliable internetwork.



Networking Protocol: TCP/IP







TCP/IP Model

Application Layer
Application programs using the network

Transport Layer (TCP/UDP)
Management of end-to-end message transmission, error detection and error correction

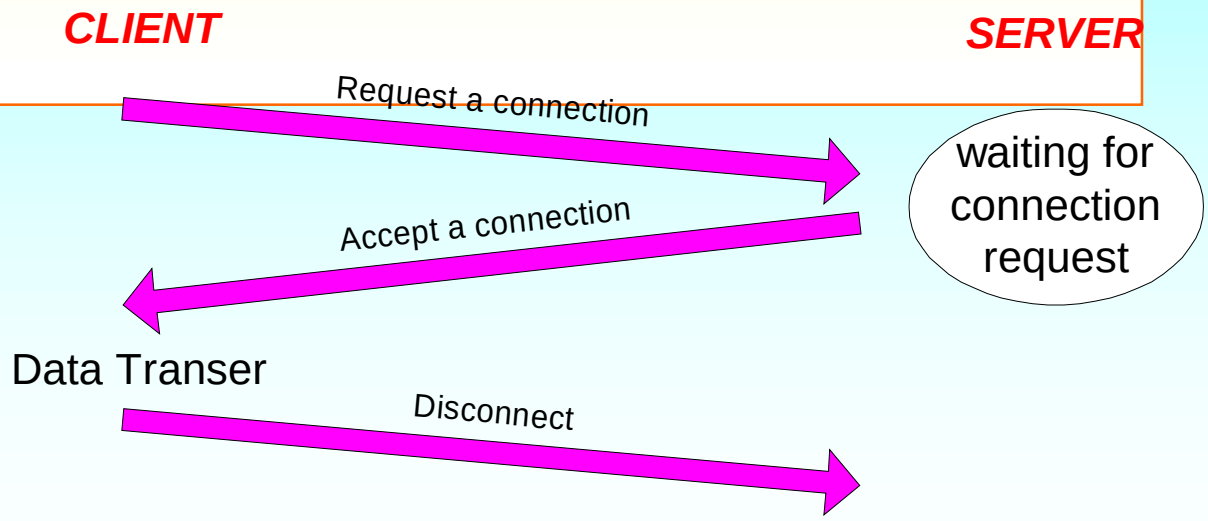
Network Layer (IP)
Handling of datagrams : routing and congestion

Data Link Layer
Management of cost effective and reliable data delivery, access to physical networks

Physical Layer
Physical Media

Connection-Oriented

- Before any data transfer, TCP establishes a **connection**:
  - One TCP entity is waiting for a connection ("**server**")
  - The other TCP entity ("**client**") contacts the server
- The actual procedure for setting up connections is more complex.
- Each connection is full duplex

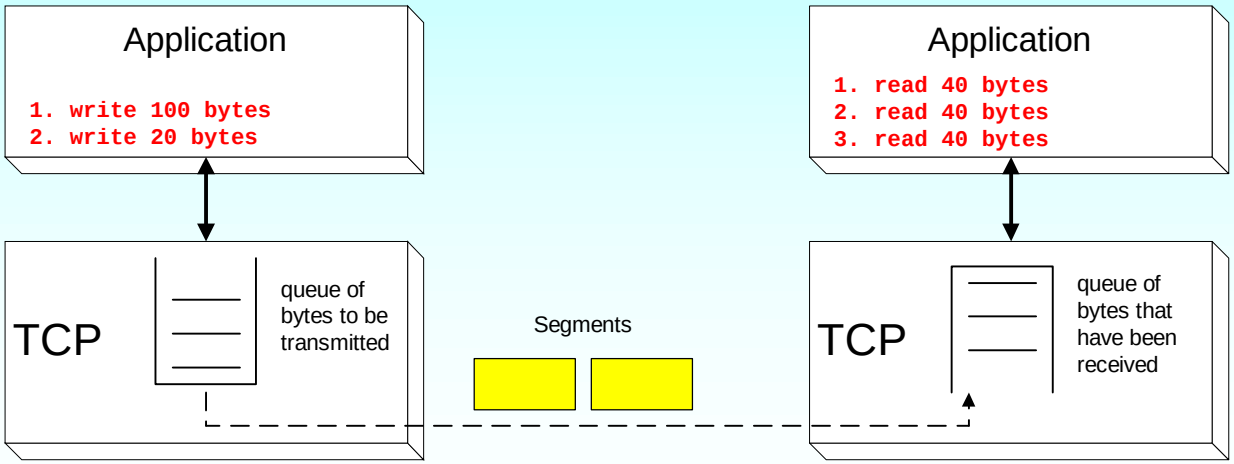


Reliable

- Byte stream is broken up into chunks which are called **segments**
  - Receiver sends acknowledgements (ACKs) for segments
  - TCP maintains a timer. If an ACK is not received in time, the segment is retransmitted
- **Detecting errors:**
  - TCP has checksums for header and data. Segments with invalid checksums are discarded
  - Each byte that is transmitted has a sequence number

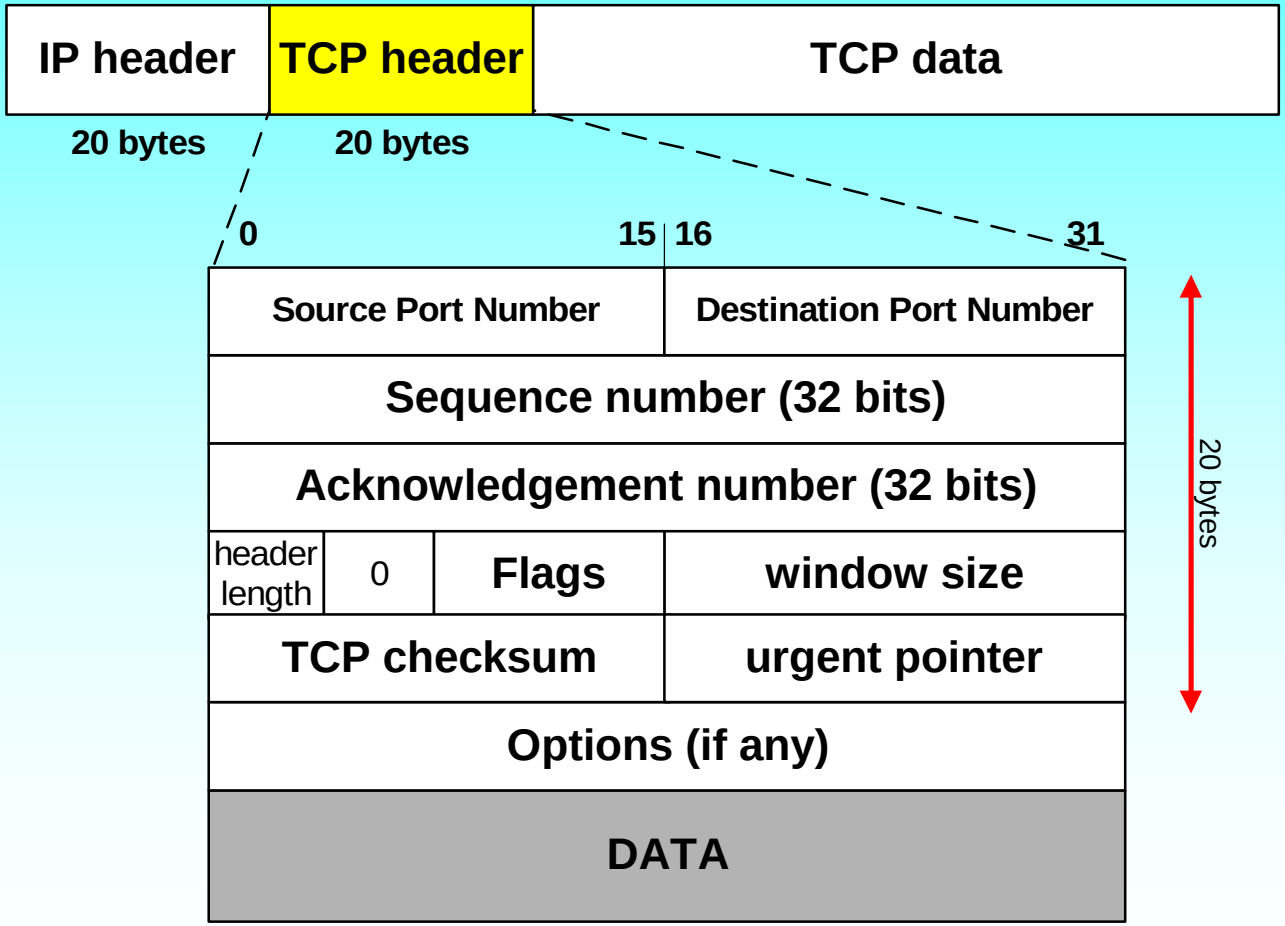
Byte Stream Service

- To the lower layers, TCP handles data in blocks, the segments.
- To the higher layers TCP handles data as a sequence of bytes and does not identify boundaries between bytes
- So: Higher layers do not know about the beginning and end of segments !



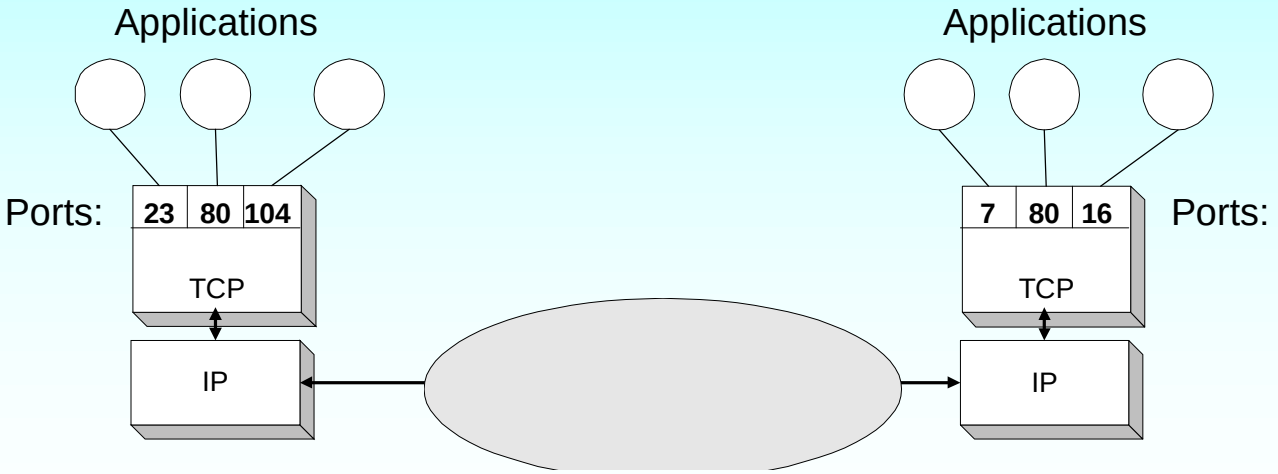
Format of TCP segments

- TCP segments have a min. 20 byte header with ≥ 0 bytes of data.



TCP header fields

- **Port Number:**
  - A port number identifies the endpoint of a connection.
  - A pair **(IP address, port number)** identifies one endpoint of a connection.
  - Two pairs **(client IP address, client port number)** and **(server IP address, server port number)** identify a TCP connection.



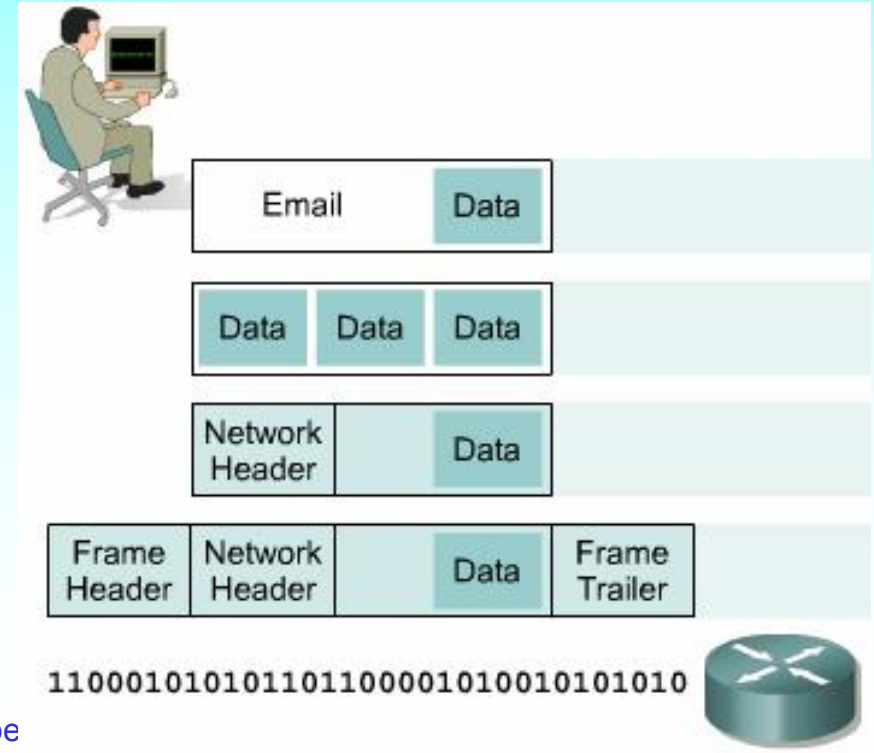
TCP States

State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for Ack
SYN SENT	The client has started to open a connection
ESTABLISHED	Normal data transfer state
FIN WAIT 1	Client has said it is finished
FIN WAIT 2	Server has agreed to release
TIMED WAIT	Wait for pending packets (“2MSL wait state”)
CLOSING	Both Sides have tried to close simultanesously
CLOSE WAIT	Server has initiated a release
LAST ACK	Wait for pending packets

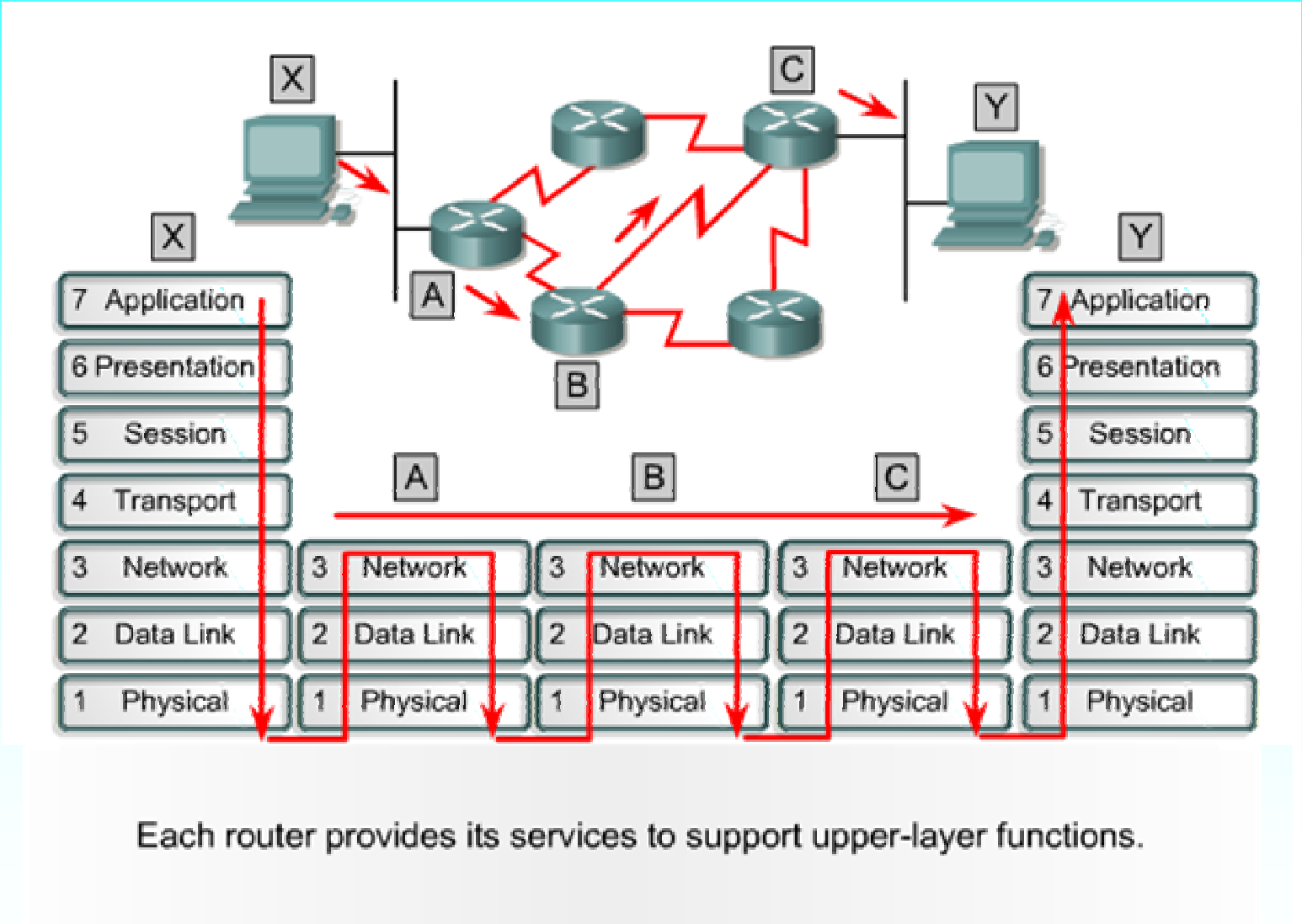
- **INTERNET PROTOCOL**
- **(IP)**

IP като маршрутен протокол (Routed Protocol)

- **IP is a connectionless, unreliable, best-effort delivery protocol.**
- **IP accepts whatever data is passed down to it from the upper layers and forwards the data in the form of IP Packets.**
- **All the nodes are identified using an IP address.**
- **Packets are delivered from the source to the destination using IP address**



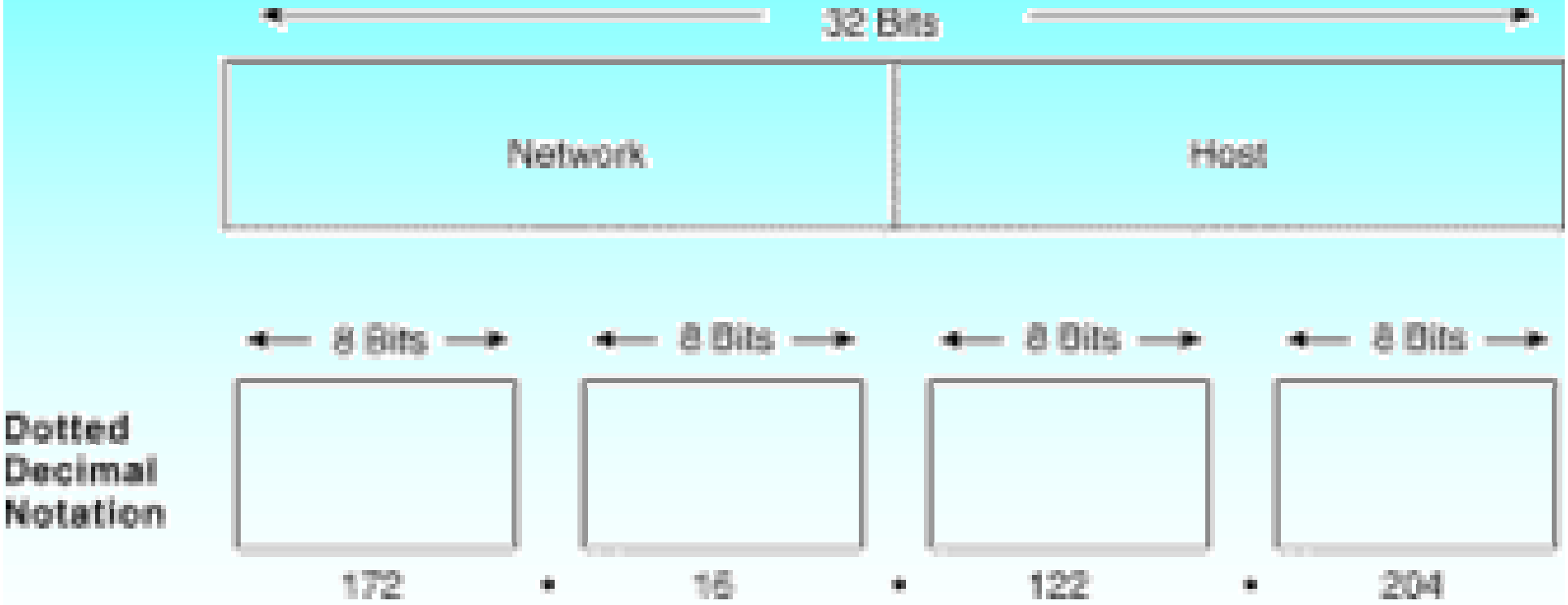
Предвижване на пакети



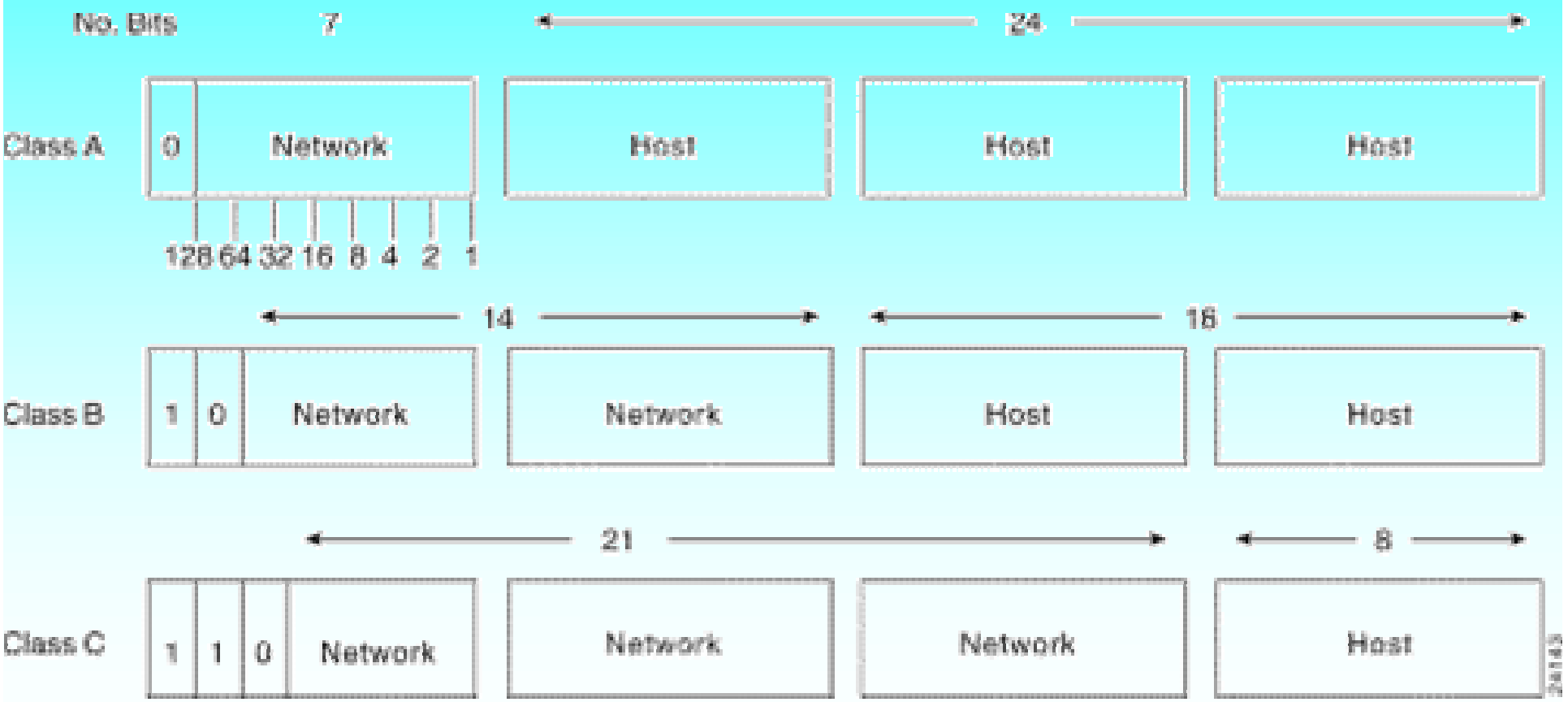
IP адрес

- **IP address is for the INTERFACE of a host. Multiple interfaces mean multiple IP addresses, i.e., routers.**
- **32 bit IP address in dotted-decimal notation for ease of reading, i.e., 193.140.195.66**
- **Address 0.0.0.0, 127.0.0.1 and 255.255.255.255 carries special meaning.**
- **IP address is divided into a network number and a host number.**
- **Also bits in Network or Host Address cannot be all 0 or 1.**

IP адрес


32 Bits
Network
Host
8 Bits
8 Bits
8 Bits
8 Bits
Dotted
Decimal
Notation
172
16
122
204

IP адрес


No. Bits
7
24
Class A
0
Network
Host
Host
Host
128 64 32 16 8 4 2 1
14
16
Class B
1
0
Network
Network
Host
Host
21
8
Class C
1
1
0
Network
Network
Network
Host

IP адрес

- **Class A : Address begins with bit 0. It has 8 bit network number (range 0.0.0.0-to-127.255.255.255), 24 bit host number.**
- **Class B : Address begins with bits 10. It has 16 bit network number (range 128.0.0.0-to-191.255.255.255), 16 bit host number.**
- **Class C : Address begins with bits 110. It has 24 bit network number (range 192.0.0.0-to-223.255.255.255), 8 bit host number.**
- **Class D : Begins with 1110, multicast addresses (224.0.0.0-to-239.255.255.255)**
- **Class E : Begins with 11110, unused**

Subnet Mask

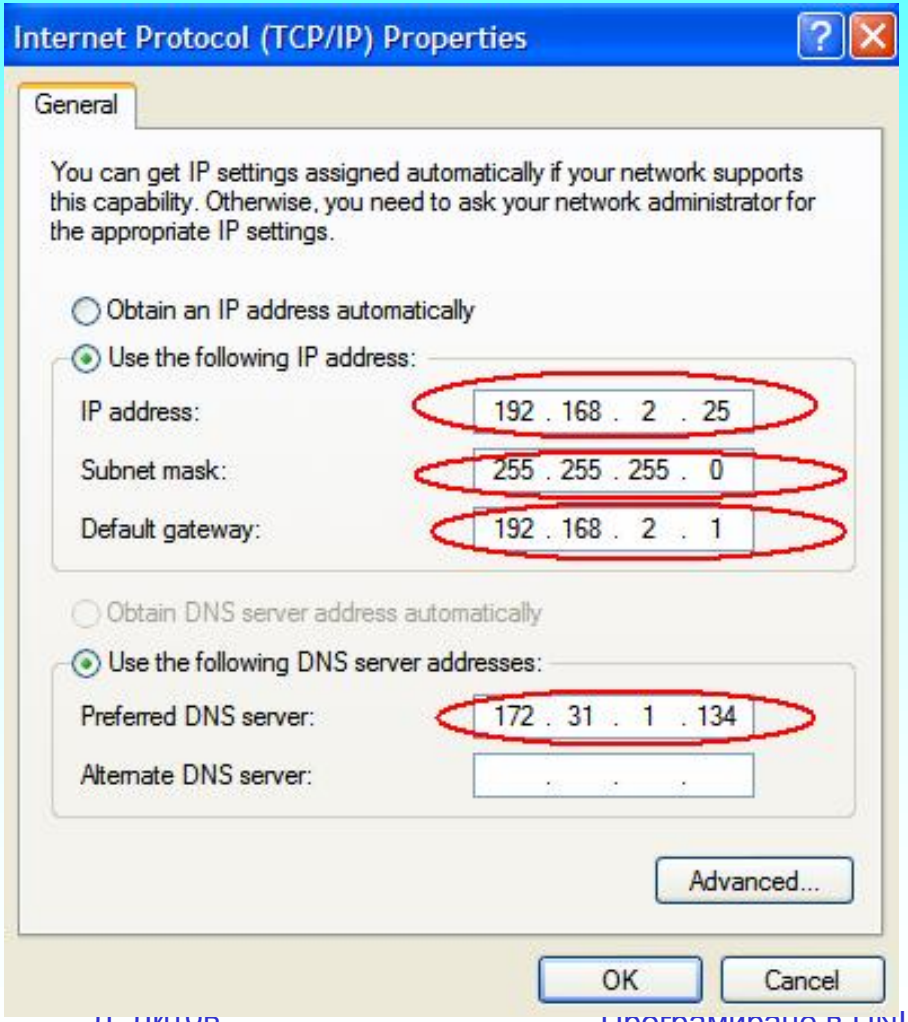
- Consider IP address = 192.168.2.25
  - First few bits (left to right) identify network/subnet
  - Remaining bits identify host/interface
- Number of subnet bits is called subnet mask, e.g.
  - Subnet IP Address range is 192.168.2.0 – 192.168.2.255 or Mask = 255.255.255.0
  - Subnet IP Address range is 192.168.2.0 – 192.168.2.15 or Mask = 255.255.255.240

IP адрес, подмрежова маска и шлюз
(IP Address, Subnet Mask and Gateway)

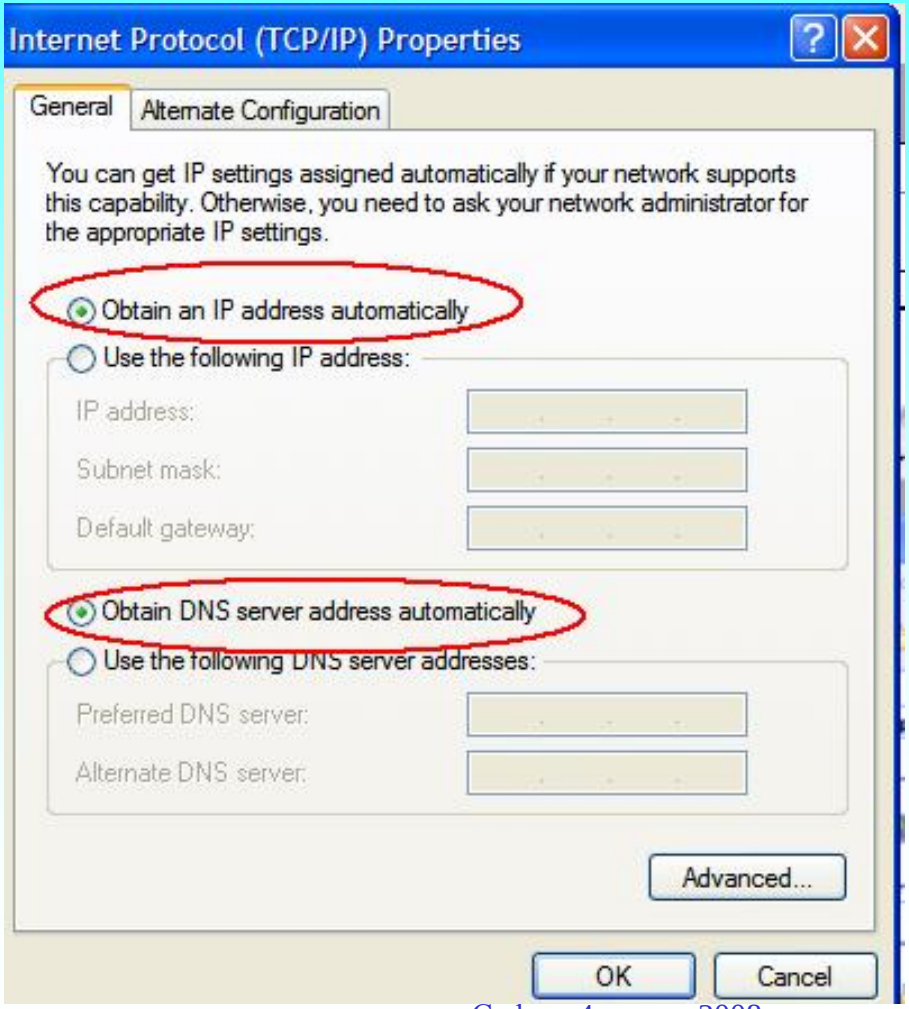
- **IP Address and Subnet Mask define the Subnet**
- **For Example IP address 172.31.1.0 and Subnet Mask of 255.255.240.0 means that the subnet address ranges from 172.31.0.0 to 172.31.15.255**
- **Another notation is 172.31.1.0/28**
- **The first Address is the Network Address and the last Address is the Broadcast Address. They are reserved and cannot be assigned to any node.**
- **The Gateway Address is the Address of the router where the packet should be sent in case the destination host does not belong to the same subnet**

IP configuration of an interface

Static



DHCP



ARP
(Address Resolution Protocol)

- **ARP (Address Resolution Protocol) is used in Ethernet Networks to find the MAC address of a node given its IP address.**
- **Source node (say 192.168.2.32) sends broadcast message (ARP Request) on its subnet asking ``Who is 192.168.2.33''.**
- **All computers on subnet receive this request**
- **Destination responds (ARP Reply) since it has 192.168.2.33**
  - **Provides its MAC address in response**

IPv6

- **Internet Protocol Version 4 is the most popular protocol in use today, although there are some questions about its capability to serve the Internet community much longer.**
- **IPv4 was finished in the 1970s and has started to show its age.**
- **The main issue surrounding IPv4 is addressing—or, the lack of addressing—because many experts believe that we are nearly out of the four billion addresses available in IPv4.**
- **Although this seems like a very large number of addresses, multiple large blocks are given to government agencies and large organizations.**
- **IPv6 could be the solution to many problems posed by IPv4**

IPv6

- **IPv6 uses 128 bit address instead of 32 bit address.**
- **The IPv6 addresses are being distributed and are supposed to be used based on geographical location.**

qLAN технологии

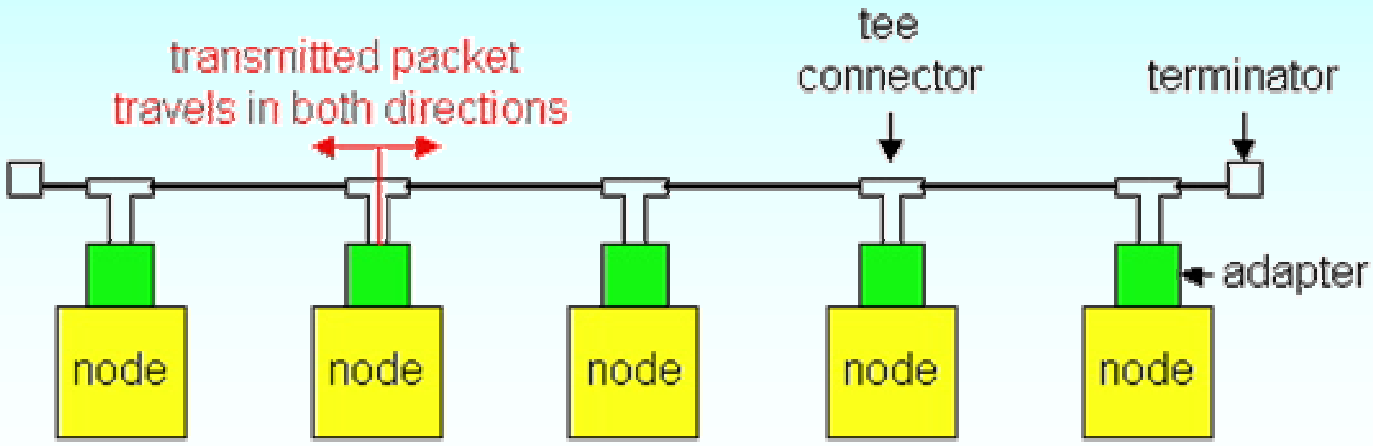
Технологични възможности

- **Ethernet**
- **Fast Ethernet**
- **Gigabit Ethernet**
- **10 Gig Ethernet**

- **WLAN**

Media access

- **Ethernet and Wi-Fi are both “multi-access” technologies**
  - Broadcast medium, shared by many hosts
  - Simultaneous transmissions will result in collisions
- **Media Access Control (MAC) protocol required**
  - Rules on how to share medium
- **The Data Link Layer is divided into two Part MAC Media Access Control) Sublayer and LLC (Logic Link Control) Sublayer**



802.3 Ethernet

- **Carrier-sense multiple access with collision detection (CSMA/CD).**
  - CS = carrier sense
  - MA = multiple access
  - CD = collision detection
- **Base Ethernet standard is 10 Mbps.**
  - 100Mbps, 1Gbps, 10Gbps standards came later

Ethernet CSMA/CD

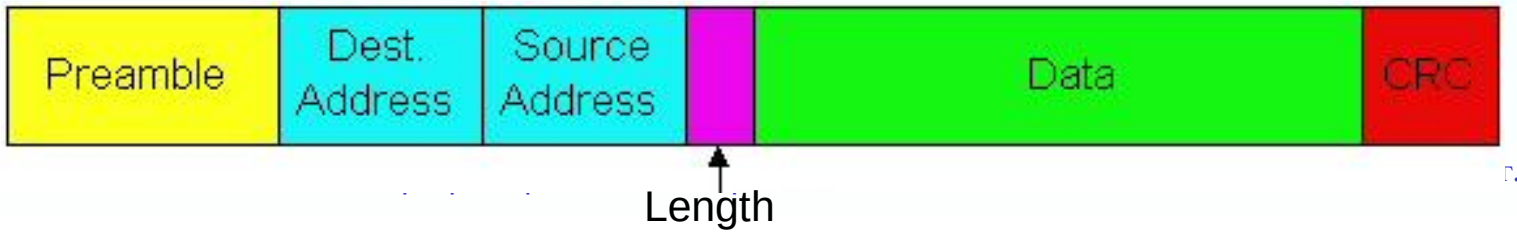
- **CSMA/CD (carrier sense multiple access with collision detection) media access protocol is used.**
  - Data is transmitted in the form of packets.
  - Sense channel prior to actual packet transmission.
  - Transmit packet only if channel is sensed idle; else, defer the transmission until channel becomes idle.
  - After packet transmission is started, the node monitors its own transmission to see if the packet has experienced a collision.
  - If the packet is observed to be undergoing a collision, the transmission is aborted and the packet is retransmitted after a random interval of time using Binary Exponential Backoff algorithm.

Ethernet Address

- **End nodes are identified by their Ethernet Addresses (MAC Address or Hardware Address) which is a unique 6 Byte address.**
- **MAC Address is represented in Hexa Decimal format e.g 00:05:5D:FE:10:0A**
- **The first 3 bytes identify a vendor (also called prefix) and the last 3 bytes are unique for every host or device**

Ethernet Frame Structure

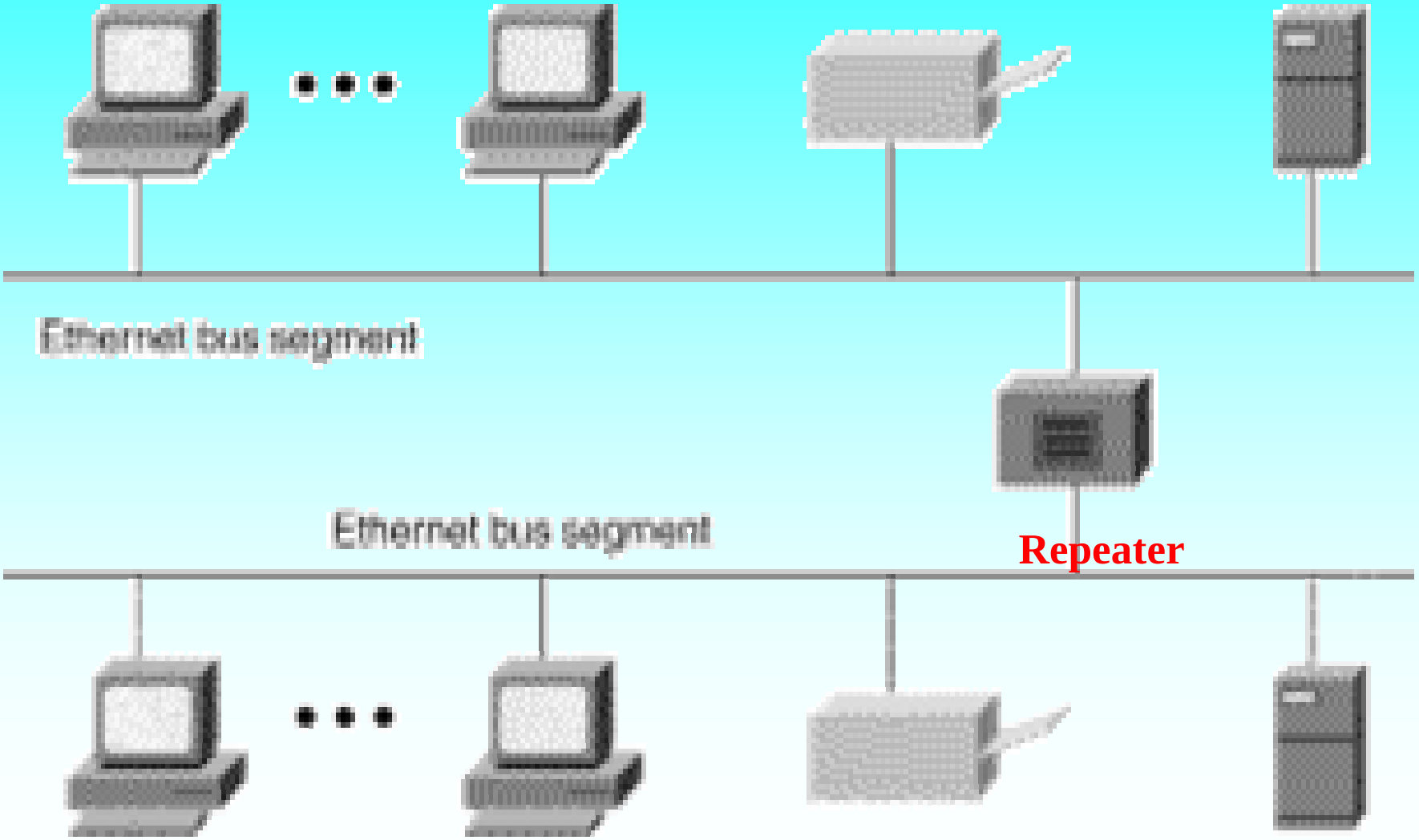
- **Preamble:**
  - 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
  - Used to synchronize receiver, sender clock rates
- **Addresses:** 6 bytes, frame is received by all adapters on a LAN and dropped if address does not match
- **Length:** 2 bytes, length of Data field
- **CRC:** 4 bytes generated using CR-32, checked at receiver, if error is detected, the frame is simply dropped
- **Data Payload:** Maximum 1500 bytes, minimum 46 bytes
  - If data is less than 46 bytes, pad with zeros to 46 bytes



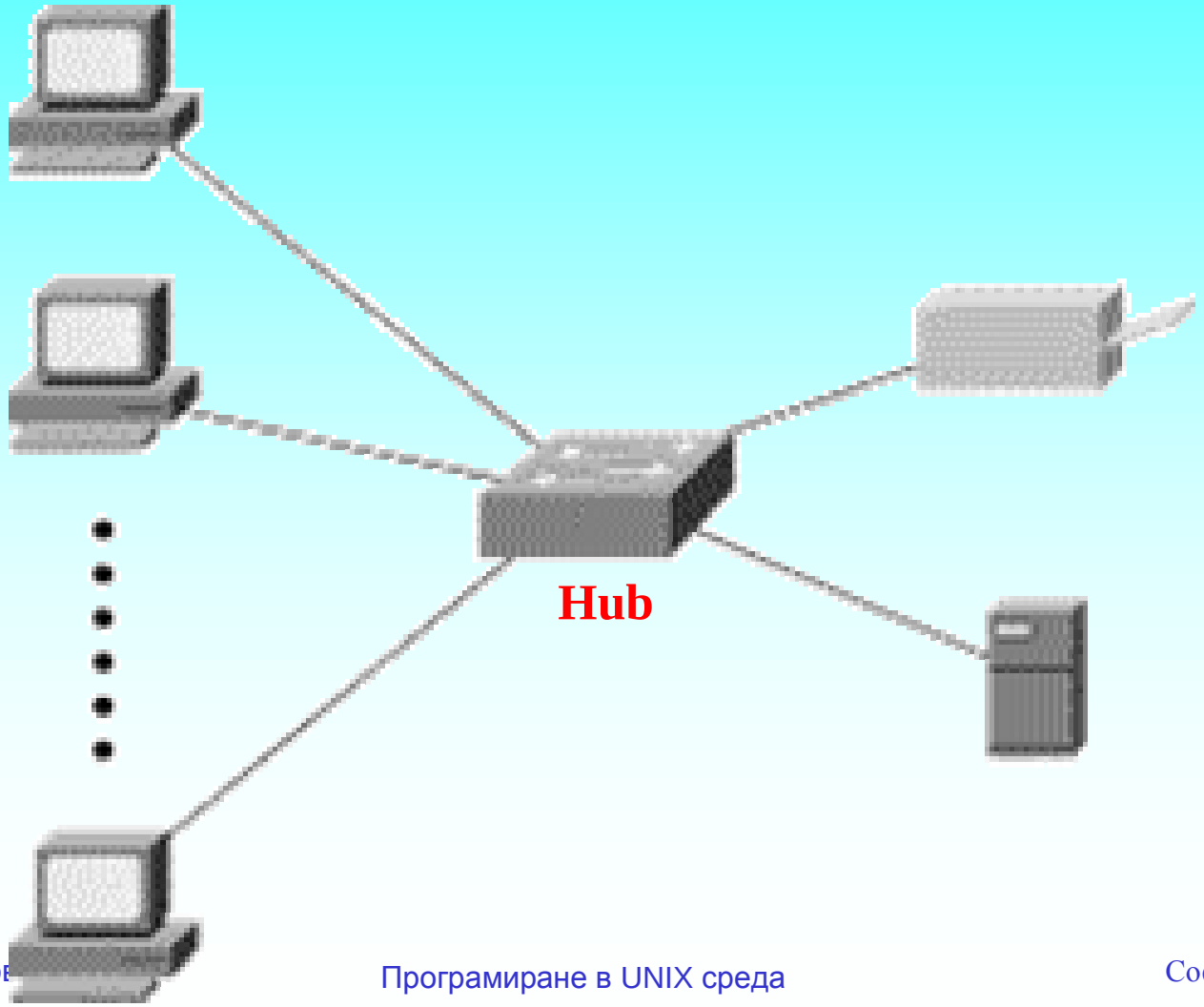
Ethernet

- 10 Base 5 (Thicknet) (Bus Topology)
- 10 Base 2 (Thinnet) (Bus Topology)
- 10 Base T (UTP) (Star/Tree Topology)
- 10 Base FL (Fiber) (Star/Tree Topology)

Ethernet BUS Topology



Ethernet STAR Topology



Ethernet

- **Physical Media :-**
  - 10 Base5 - Thick Co-axial Cable with Bus Topology
  - 10 Base2 - Thin Co-axial Cable with Bus Topology
  - 10 BaseT - UTP Cat 3/5 with Tree Topology
  - 10 BaseFL - Multimode/Singlemode Fiber with Tree
  - Topology

- **Maximum Segment Length**
  - 10 Base5 - 500 m with at most 4 repeaters (Use Bridge to extend the network)
  - 10 Base2 - 185 m with at most 4 repeaters (Use Bridge to extend the network)
  - 10 BaseT - 100 m with at most 4 hubs (Use Switch to extend the network)

Fast Ethernet

- **100 Mbps bandwidth**
- **Uses same CSMA/CD media access protocol and packet format as in Ethernet.**
- **100BaseTX (UTP) and 100BaseFX (Fiber) standards**
- **Physical media :-**
  - 100 BaseTX - UTP Cat 5e
  - 100 BaseFX - Multimode / Singlemode Fiber
- **Full Duplex/Half Duplex operations.**

Fast Ethernet

- **Provision for Auto-Negotiation of media speed: 10 Mbps or 100Mbps (popularly available for copper media only).**

- **Maximum Segment Length**
  - **100 Base TX - 100 m**
  - **100 Base FX - 2 Km (Multimode Fiber)**
  - **100 Base FX - 20 km (Singlemode Fiber)**

Gigabit Ethernet

- **1 Gbps bandwidth.**
- **Uses same CSMA/CD media access protocol as in Ethernet and is backward compatible (10/100/100 modules are available).**
- **1000BaseT (UTP), 1000BaseSX (Multimode Fiber) and 1000BaseLX (Multimode/Singlemode Fiber) standards.**
- **Maximum Segment Length**
  - 1000 Base T - 100m (Cat 5e/6)
  - 1000 Base SX - 275 m (Multimode Fiber)
  - 1000 Base LX - 512 m (Multimode Fiber)
  - 1000 Base LX - 20 Km (Singlemode Fiber)
  - 1000 Base LH - 80 Km (Singlemode Fiber)

10 Gig Ethernet

- **10 Gbps bandwidth.**
- **Uses same CSMA/CD media access protocol as in Ethernet.**
- **Propositioned for Metro-Ethernet**
- **Maximum Segment Length**
  - 1000 Base-T - Not available
  - 10GBase-LR - 10 Km (Singlemode Fiber)
  - 10GBase-ER - 40 Km (Singlemode Fiber)

SSH - Lab

Ø We will now practice the following concepts:

- The use of known_hosts files
- SSH connection with password authentication
- RSA version 2 protocol key generation
- Public key copying
- Connecting with private key passphrase using key-based authentication
- Using scp with RSA key authentication
- Some ssh “hacks” without passwords.

*Technically you are still challenged (even if that is a bad pun in English).

SSH - Lab Cont.

Ø The use of known_hosts files

- Ø Connect to the machine next to your machine using ssh:
  `ssh admin@pcN.cctld.pacnog2.dnsdojo.net`
- Ø If this is your first connection to this machine you should see (example uses host1 connecting to host2):

```
pc1# ssh admin@pc2.cctld.pacnog2.dnsdojo.net
The authenticity of host 'pc2.cctld.pacnog2.dnsdojo.net (192.216.0.2)'
can't be established.
RSA1 key fingerprint is 60:f7:04:8b:f7:61:c4:41:6e:9a:6f:53:7d:95:cb:29.
Are you sure you want to continue connecting (yes/no)?
```

- Ø Go ahead and answer “yes” here, but we'll discuss the implications of this in class. Are there ways around this? Could this be a “man in the middle” attack? What file is created or updated? Why?

SSH - Lab Cont.

Ø ssh connection with password authentication

- **Ø At the prompt below when you answered yes, you were asked to enter in the admin password for pc2.cctld.pacnog2.dnsdojo.net:**

```
host1# ssh admin@pc2.cctld.pacnog2.dnsdojo.net
The authenticity of host 'pc2.cctld.pacnog2.dnsdojo.net (192.216.0.2)' can't
be established.
RSA2 key fingerprint is 60:f7:04:8b:f7:61:c4:41:6e:9a:6f:53:7d:95:cb:29.
Are you sure you want to continue connecting (yes/no)? yes
```

- **Ø And, this is what you should have seen:**

```
Warning: Permanently added 'pc2.cctld.pacnog2.dnsdojo.net' (RSA2) to the list
of known hosts.
    [/etc/ssh/ssh_host_key.pub]

admin@pc2.cctld.pacnog2.dnsdojo.net's password:
```

- **Ø Now you are “securely” connected as admin to pc2.cctld.pacnog2.dnsdojo.net - We will discuss what happened during this connection.**

SSH - Lab Cont.

Ø rsa1/rsa2/dsa Key Generation

- **Ø We will now generate a single RSA SSH protocol 2 key of 2048 bits. To do this, issue the following command. If you are logged in on the other machine, logout first!**

```
ssh-keygen -t rsa -b 2048
```

- **Ø You will be prompted for a file location for the key as well as for a passphrase to encrypt the key file. Be sure to enter a passphrase. Private key files without passphrases are a security hole, or maybe not... We'll discuss this as we complete this excercise. You can use a passphrase other than what was given in class for the *admin account* if you wish.**

SSH - Lab Cont.

Ø RSA 2 Key Generation

- **Here is the output from the command “ssh-keygen -t rsa -b 2048”:**

```
pc1# ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key
(/admin/.ssh/id_rsa): [enter]
Enter passphrase (empty for no passphrase): [pw]
Enter same passphrase again: [pw]
Your identification has been saved in
/admin/.ssh/id_rsa.
Your public key has been saved in
/admin/.ssh/id_rsa.pub.
The key fingerprint is:
0f:f5:b3:bc:f7:5b:c8:ce:79:d0:b1:ab:2c:67:21:62
admin@pc1.ws.cctld.ke
pc1#
```

SSH - Lab Cont.

Ø Public Key Copying

- Ø Now that you have a public and private RSA(2) set of keys you can take advantage of them. We will copy the public key to the same host you connected to previously, save this to the files *known_hosts*, and then reconnect to the host and see the difference:
- Ø First you must copy the public key files to the host you used previously (pcn.cctld.pacnog2.dnsdojo.net):

```
cd ~/.ssh
scp id_rsa.pub admin@pcn.cctld.pacnog2.dnsdojo.net:/tmp/.
```

- Ø You will be prompted for the password for the host *and username* you are connecting to. We continue with our example using pc1 connecting to pc2 as admin.

SSH - Lab Cont.

Ø Public Key Copying

Ø The output from the command on the previous page looks like:

```
pc1# scp *.pub admin@pc2.ws.cctld.ke:/tmp/.
admin@pc2.ws.cctld.ke's password:
id_rsa.pub                    100% |*****************************|   408       00:00
pc1#
```

You now have the public key file sitting on the host that will need them to use RSA/DSA public/private key authentication with you. You next step is to place these keys in the appropriate files.

You need the RSA keys in *~/.ssh/authorized_keys*

You can try to figure this out, or go to the next slide for steps to do this:

SSH - Lab Cont.

Ø Public Key Copying

- Ø To copy the public keys to the correct places do the following:

```
ssh admin@pcn.cctld.pacnog2.dnsdojo.net
cat /tmp/id_rsa.pub >> ~/.ssh/authorized_keys
rm /tmp/id_rsa.pub
exit
```

- Ø If you are unsure of what these commands do they will they are explained in class. In addition, you can do this many different ways, and you could issue the commands differently as well. If you understand what these commands do and have a preferred method, then feel free to use it.
- Ø Go to the next slide to connect with your public/private keys!

SSH - Lab Cont.

Ø Public/Private Key Connection

- Ø To connect using your RSA protocol 2 key simply type:

  `ssh admin@pcn.cctld.pacnog2.dnsdojo.net`

- Ø And, here is the output you should see (pc1 to pc2 example):

```
host1# ssh admin@pc2.cctld.pacnog2.dnsdojo.net
Enter passphrase for RSA key 'admin@pc1.cctld.pacnog2.dnsdojo.net':
```

- Ø This is actually pretty neat! You *did not enter in the admin password for the admin account on pcn.cctld.pacnog2.dnsdojo.net, but rather you used the passphrase that you chose for your private RSA protocol 2 key when you issued the command "**ssh-keygen -t rsa -b 2048**" - This was used to decode the encoded random number exchanged between the hosts (remember "Magic Phrase?").*

SSH - Lab Cont.

Ø SCP Public/Private Key Connection

- Ø First disconnect from the ssh session you previously made:

  `exit`

- Ø Now, try copying a file from your machine to the other machine (pick a small file) using SCP (SeCure coPy):

  ```
  scp filename admin@pcn.cctld.pacnog2.dnsdojo.net:/tmp/.
  ```

- Ø What did you notice? You should have noticed that you no longer get a password challenge to this account on this node, but rather you need to provide your RSA protocol 2 private key passphrase.
- Ø *This is expected. SCP and SSH are from the same package - OpenSSH and both use RSA and DSA keys in the same way.*

SSH - Lab Cont.

Ø Another SSH tool - SFTP

- Ø In addition to scp, ssh has a secure ftp tool called sftp. Give it a try:
- Let's use sftp to get your neighbor's /etc/motd file and place it in your /tmp directory.
  - **sftp admin@pcN.cctld.pacnog2.dnsdojo.net**
  - Once you are connected:
  - **sftp> lcd /tmp [change local directory to /tmp]**
  - **sftp> cd /etc [change remote directory to /etc]**

SSH - Lab Cont.

- ***Now let's use the power of scp***
- ***Multiple file and directory copy:***
  - ***Let's copy all the files and directories in /usr/ports/palm from your machine to your neighbor's machine using one command (1.4Mb):***
    - ***`scp -r /usr/ports/palm/* admin@pcN.cctld.pacnog2.dnsdojo.net/tmp/.`***
  - ***"-r" for recursively copy***
  - ***"/tmp/." to place files in your neighbor's /tmp directory.***

SSH - Lab Cont.

- ***Now let's use the power of scp some more!***

  (Note: we may skip this exercise...)

- ***Copy a file from one remote machine to another.***
- ***Let's move /etc/fstab on your left neighbor's machine to /tmp/fstab.copy on your right neighbor's machine using a single command.***

  -
```
scp
admin@pcLEFT.cctld.pacnog2.dnsdojo.net:/etc/fstab \
admin@pcRIGHT.cctld.pacnog2.dnsdojo.net/tmp/fstab.copy
```

- ***"\" for newline, not part of the command.***
- ***If admin password is the same on both you only enter it once.***
- ***Did you notice we renamed the file as well?***

Литература:

- http://www.wylug.org.uk/talks/2003/04/unix.pdf
- http://ce.sharif.edu/courses/ssc/unix/resources/root/Slides/unixhistory.pdf
- http://www.cs.uga.edu/~eileen/1730/Notes/intro-UNIX.ppt
- http://remus.rutgers.edu/cs416/F01
- http://www.cs.virginia.edu/~cs458/
- http://www.bobbooth.staff.shef.ac.uk/hpcs/materials/material.html
- http://www.comm.utoronto.ca/~jorg/teaching/ece461
- http://home.iitk.ac.in/~navi/sidbilinuxcourse/
- http://www.cs.washington.edu/homes/bershad/Mac/ssh/practicalmagic.pdf
- http://www.cs.cf.ac.uk/Dave/C/CE.html
- http://www.le.ac.uk/cc/tutorials/c/ccccintr.html
- http://www.shef.ac.uk/uni/academic/N-Q/phys/teaching/phy225/index.html