

Communication protocols and network security

Introduction and repetition of the basics

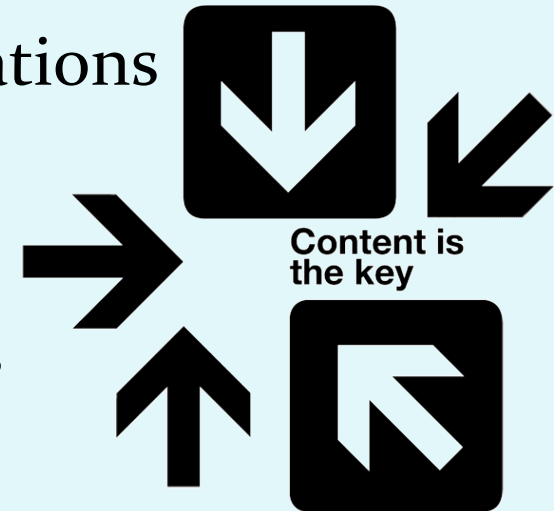
Communication protocols and network security

- **Professor:**
dr. Andrej Brodnik (Ljubljana)
- **Teaching Assistant:**
as. dr. Gašper Fele Žorž
- **Implementation of course :**
 - 3 hours of lectures –consisting of two parts, 2 hours of lab work per week
 - contact : e-mail, consultation hours, forum on the course web page



Content of the course

- Repetition of the basics of communications (ISO/OSI, TCP/IP, protocols, services, security),
- control and management of networks,
- distribution (multicasting),
- real-time applications,
- security: authentication, authorization, records, safe transfers, VPN, certification, firewalls, IDS systems,
- information for network operation, LDAP,
- IEEE 802.



Content of the course – an indicative plan

week	lecture	HW	SEM
8.10.	Introduction	1	
15.10.	Starting your computer, network configuration	1	
22.10.	Managment and control of the networks	1	
29.10.	Circulation and real-time applications	2	
5.11.	Distribution	2	
12.11.	Distribution/Preparation for the test	2	
19.11.	MIDTERM TEST 1		SEM1
26.11.	Elements of network security	3	
3.12.	Authentication , authorization, records (AAA)	3	
10.12.	Authentication, authorization and records (AAA) / avtorizacija in beleženje (AAA) / Data for network operation (LDAP)	3, 4	
17.12.	Visiting lecturer		
24.12.	<<< Christmas holidays >>>		
31. 12.	<<< Christmas holidays >>>		
7.1.	Družina IEEE 802	4	
14.1.	MIDTERM TEST 2		SEM2

Obligations

Final grade(≥ 50):

- | | |
|-------------------------------------------|------------|
| • 4 pieces of homework: | 20% |
| • 2 seminar papers: | 40% |
| • <u>written exam or 2 midterm tests:</u> | <u>40%</u> |
| | 100% |

Obligations :

- notes : 2 x per lecture, 1x laboratory work
- homework ≥ 40 , each homework ≥ 20
- seminar paper ≥ 40 , each seminar paper ≥ 20
- written exam ≥ 50 , each of the midterm tests ≥ 40

Obligations

The grade also takes into account:

- participation in the forums
- Complementing the notes
- assistance to the colleagues
- ...

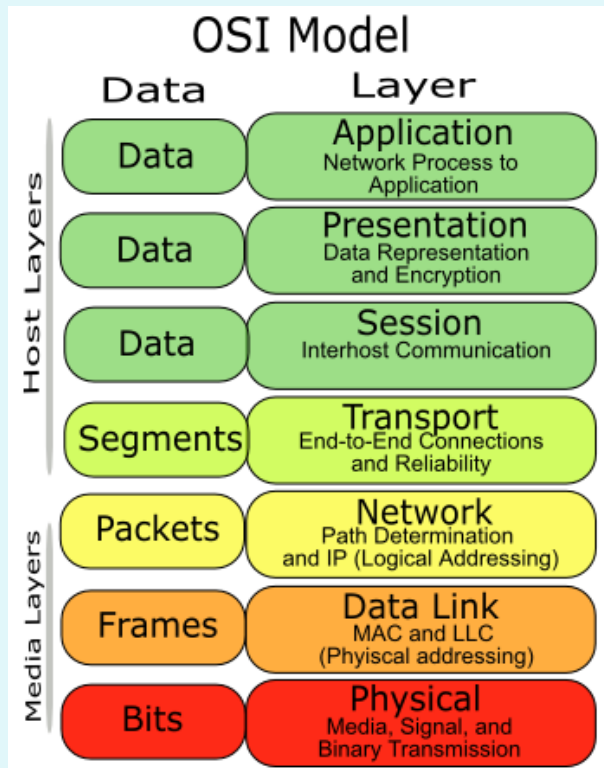
Literature

- J. F. Kurose, K. W. Ross: Computer Networking, 5th edition, Addison-Wesley, 2010.
- A. Farrel: The Internet and Its Protocols: A Comparative Approach, Morgan Kaufmann, 2004.
- E. Cole: Network Security Bible, Wiley, 2nd edition, 2009.
- Mani Subramanian: Network Management: An introduction to principles and practice, Addison Wesley Longman, 2000
- RFC
- ...

Repetition of the basics of computer communications

ISO/OSI model

- The model consists of seven layers, which define the layers of related functions of the communication system.



Application layer

Presentation layer

Session layer

Transport layer

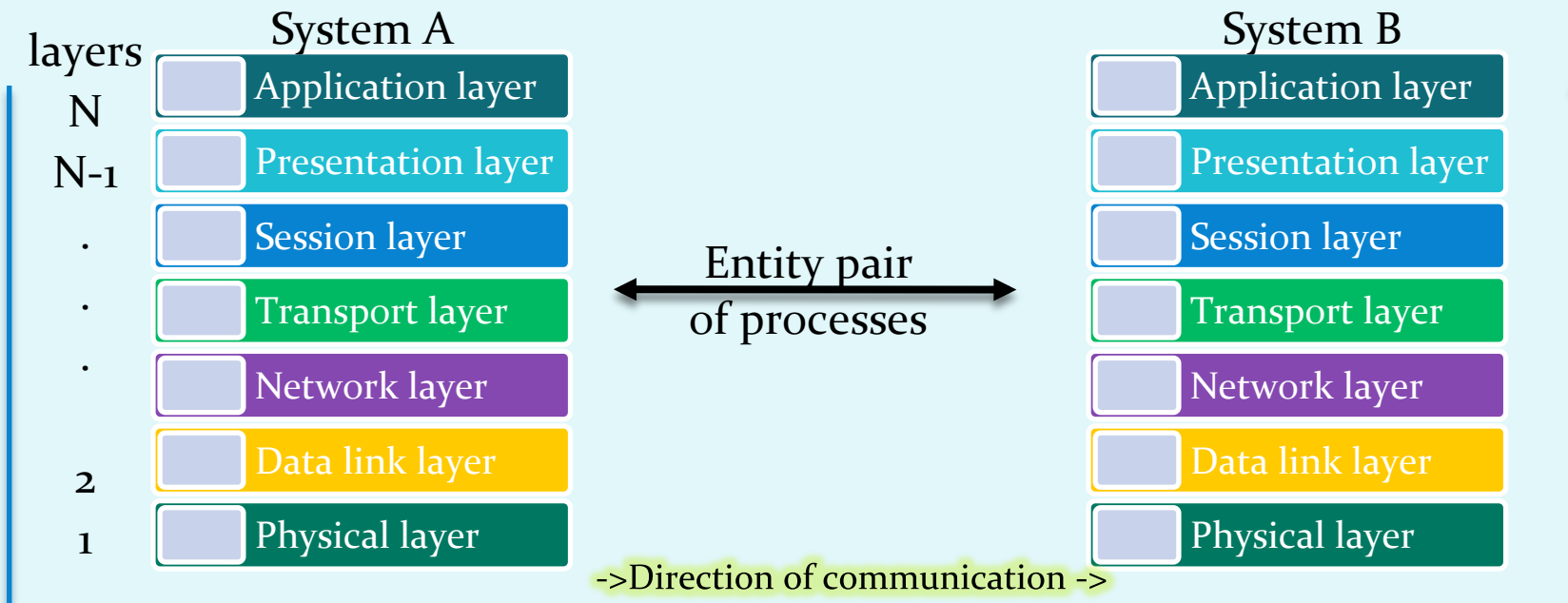
Network layer

Data link layer

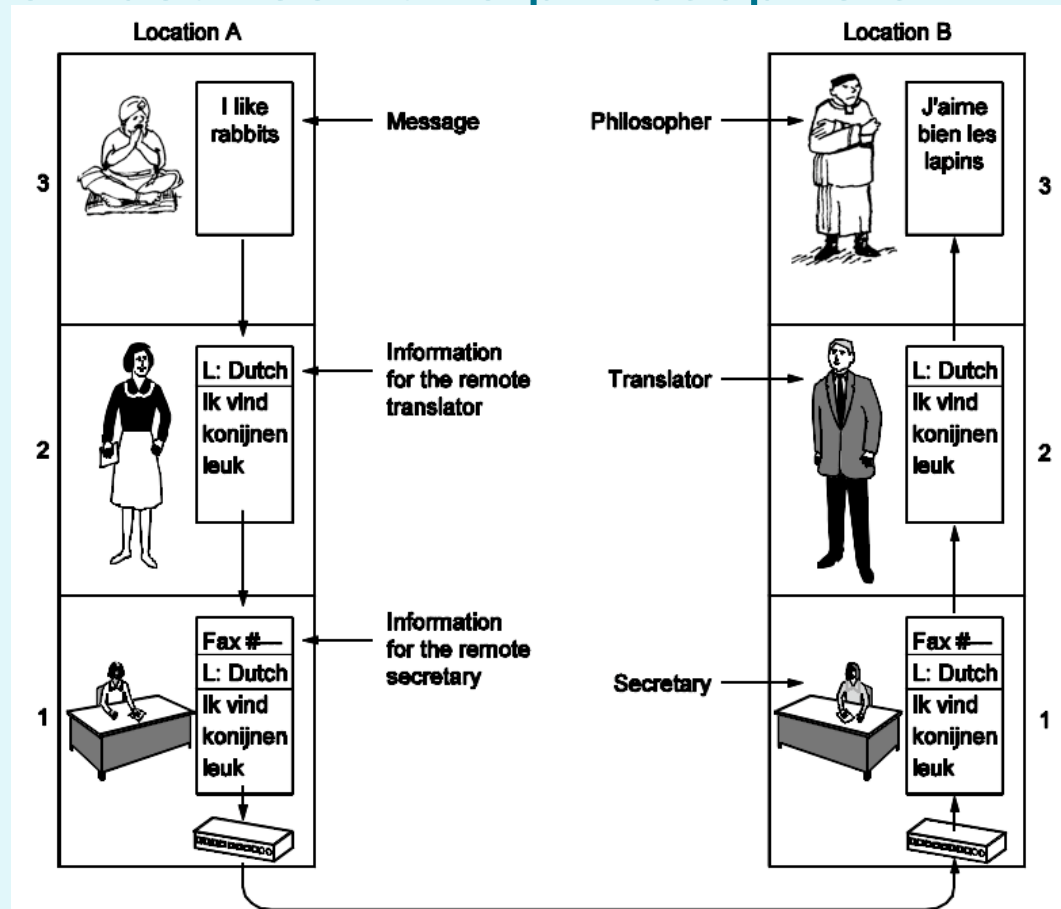
Physical layer

ISO/OSI model

- layer N provides services (serving) for layer N +1
- layer N requires services (deliverability) from layer N-1,
- Protocol: rules of communication between processes on the same location,
- Entity pair: pair of processes that communicate on the same layer



Analogy : conversation between two philosophers



- Why layers?

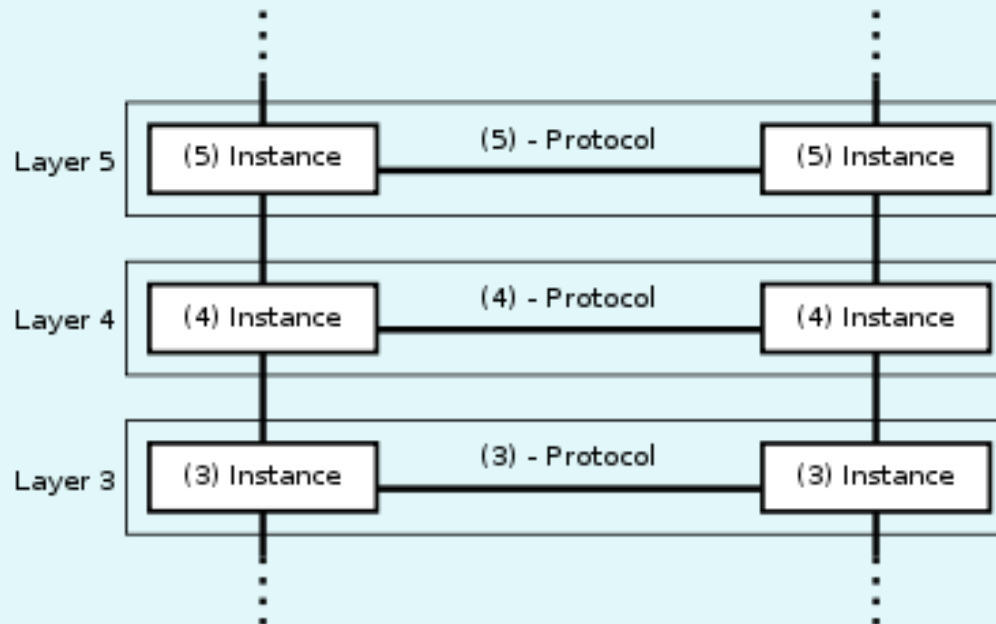
- systematic concept of system architecture,
- The change of implementation of one part of the system is independent from the rest of the system.

ISO/OSI model

In other words:

Each layer has its own protocols (the language used for communication by the processes on the same layer)

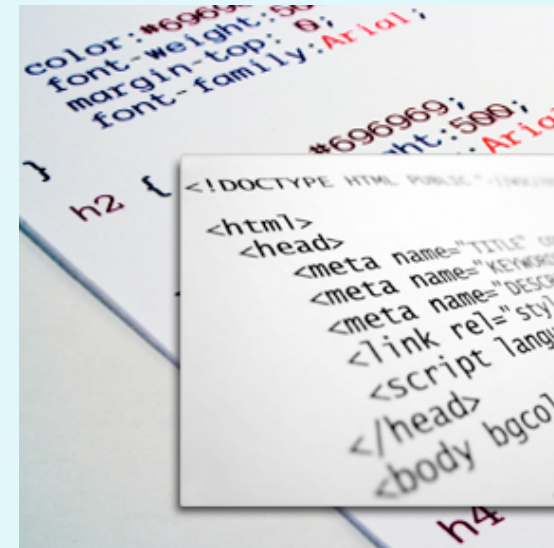
The protocols are specific for the services provided by the layer



OSI layers: detailed

- **Application layer**

- closest to the user,
- Allows application interaction with network services.
- standard services : telnet, FTP, SMTP, SNMP, HTTP



OSI layers

- **Presentation layer**

- Determines the meaning of the data between the entity pair of the application layer,
- syntax and semantics,
- provides coding, data compression, security mechanisms

- **Session layer**

- controls conversations between applications,
- logical connection between applications,
- usually it's built into the applications.

OSI layers

- **Transport layer** (unit: SEGMENT)
 - effective, reliable and transparent data transfer between users; Provide these services to higher layers,
 - Mechanisms: control of the flux, segmentation, control of the errors.,
 - Connection and connectionless oriented transfers,
 - TCP, UDP, IPSec, GRE, L2TP, PPP

The TCP Segment Format

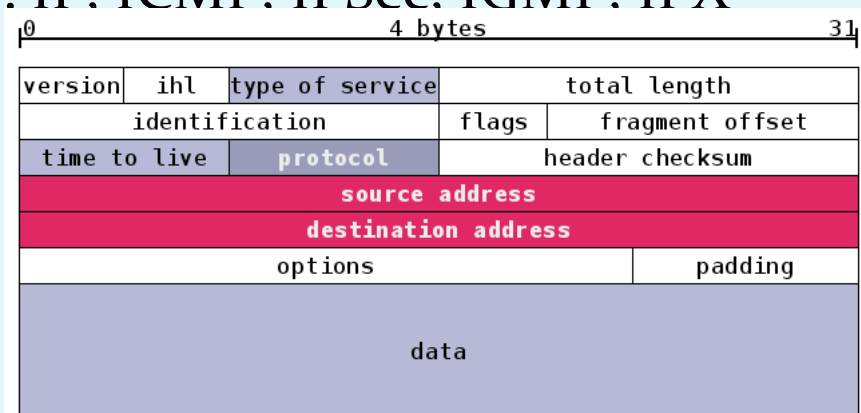


The UDP Segment Format



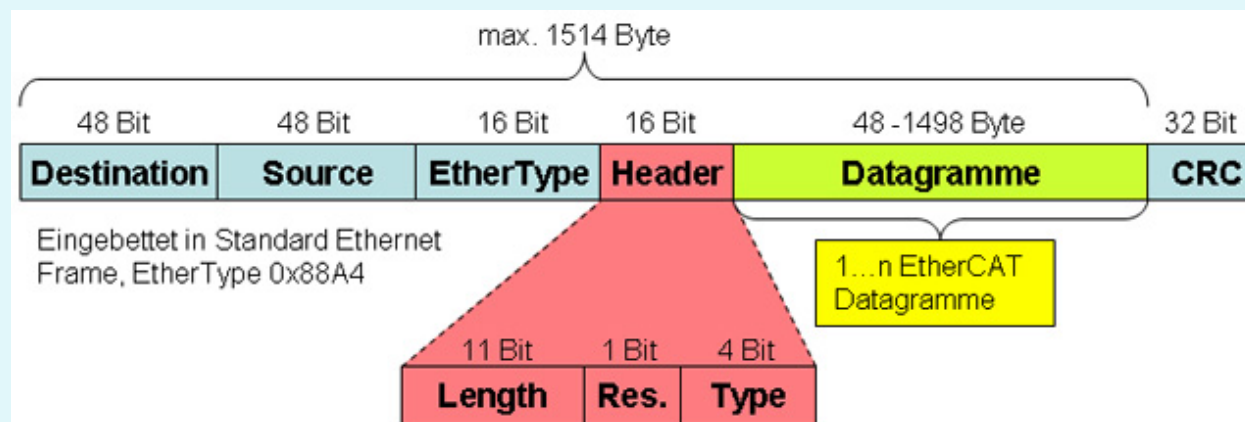
OSI layers

- **Network layer** (unit: PACKAGE)
 - routing(Connection and connectionless oriented services)
 - transmission of packages from the source to the target computer,
 - can provide : guaranteed delivery, correct sequence, fragmentation, avoiding of clogging,
 - routing, routers, routing algorithms,
 - protocols : IP. ICMP. IPSec. IGMP. IPX



OSI layers

- **Data link layer** (unit: FRAME)
 - asynchronous / synchronous communication,
 - physical addressing : MAC address,
 - detection and debugging of errors (parity, CRC, checksum)
 - Control of the flux, framing
 - protocols : Ethernet, PPP, Frame Relay



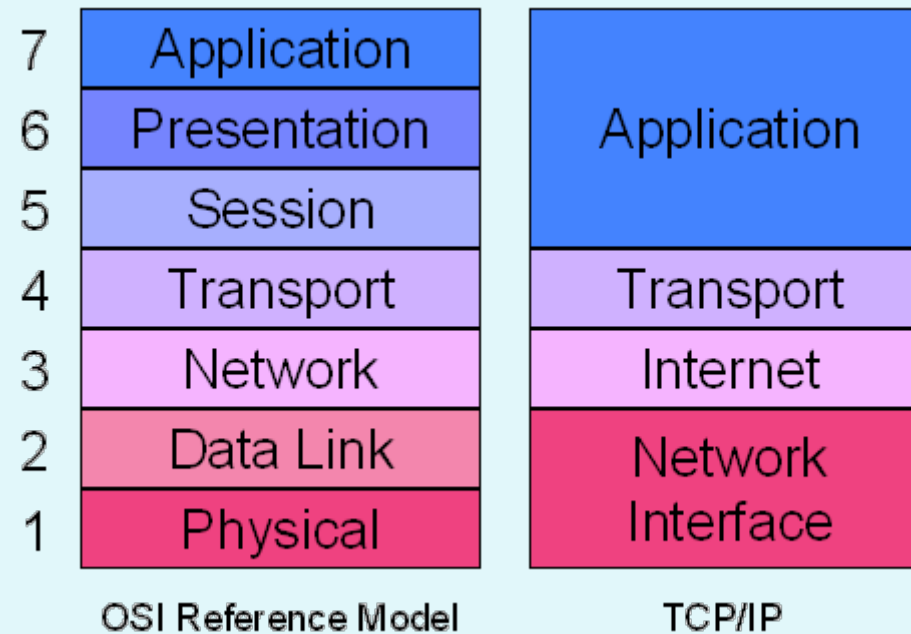
OSI layers

- **Physical layer**

- transmission of bits through the channels(copper/optics/wireless),
- digital, analog media,
- UTP, optics, coaxial cables, wireless networks,
- RS-232, T1, E1, 802.11b/g, USB, Bluetooth



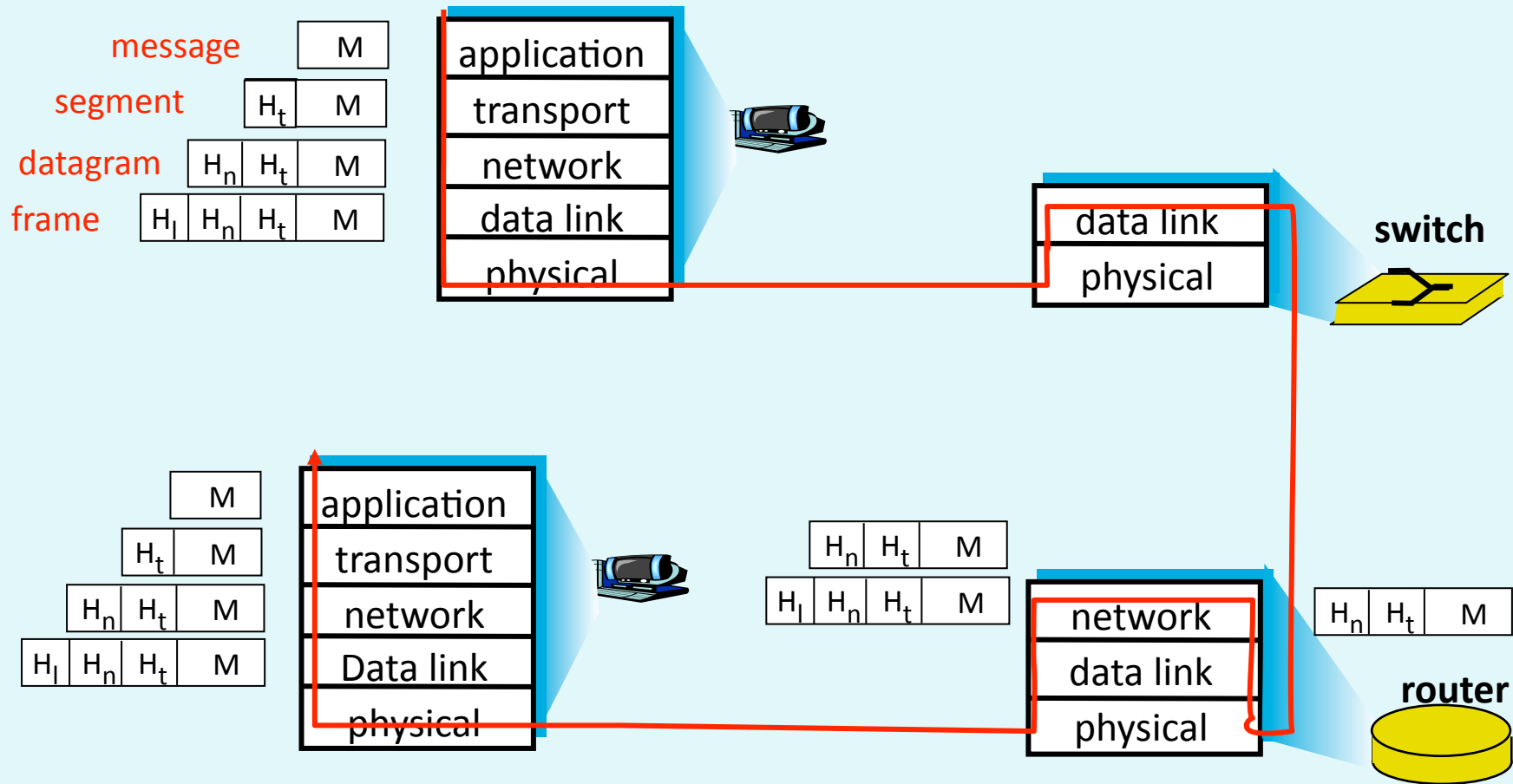
OSI model and model TCP/IP



Comparison of models :

- ISO OSI: **de iure**, theoretical, systematic, lack of implementations(products),
- TCP/IP: **de facto**, adjustable, unsystematic, many products

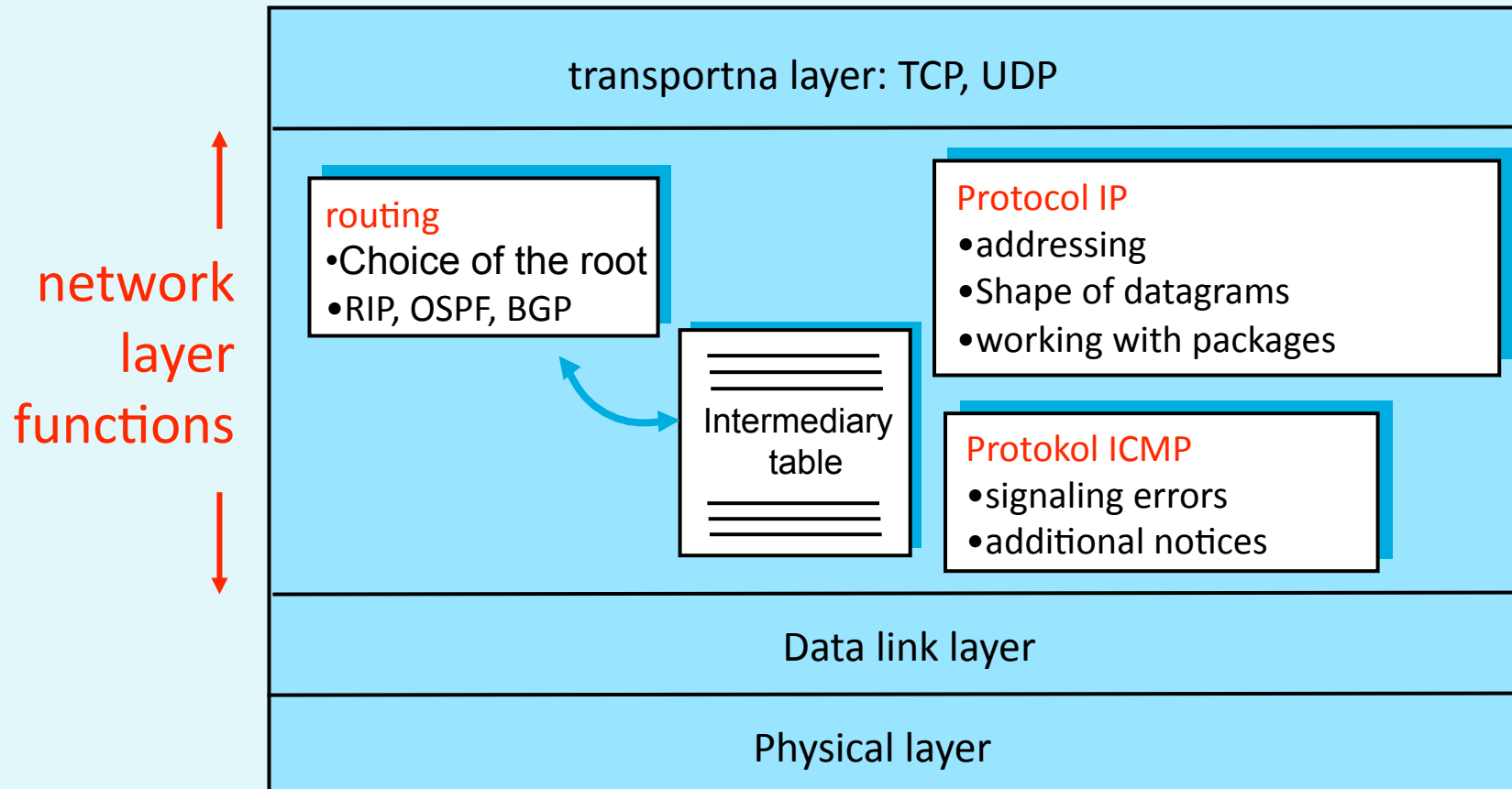
Encapsulation



Network and transport layer: detail

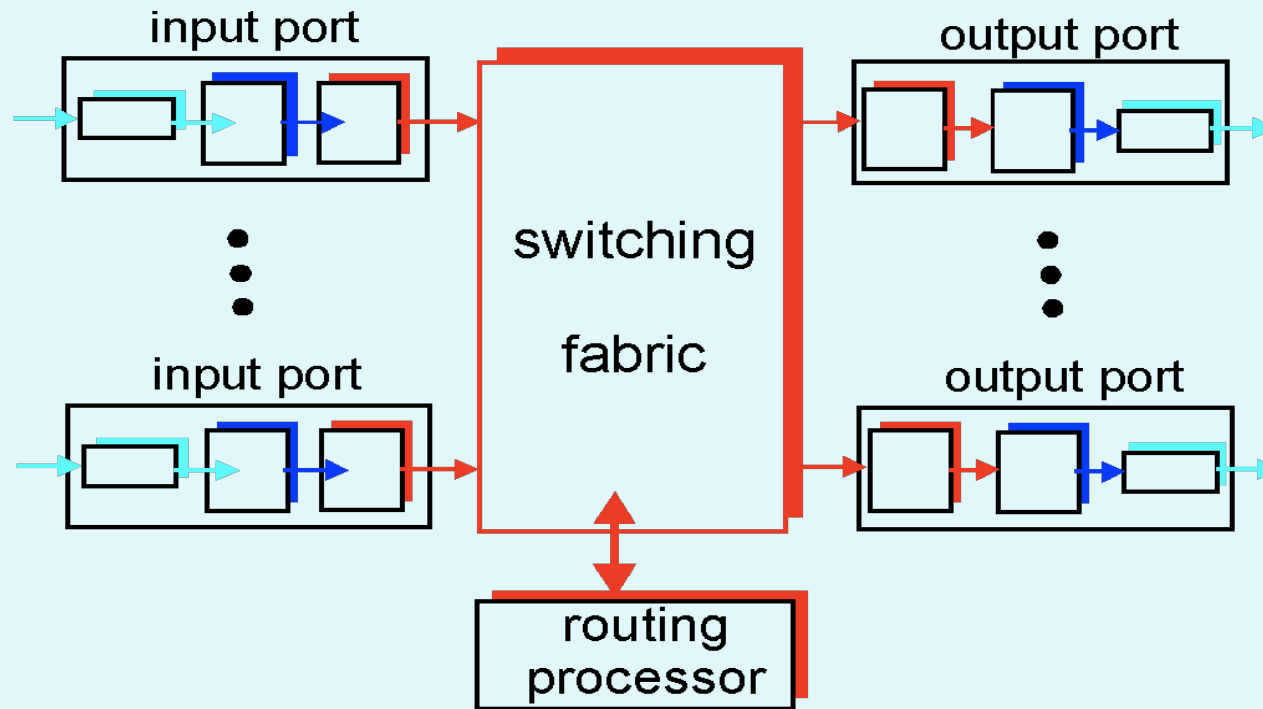
Network layer:

Network layer functions



Network layer: Routers

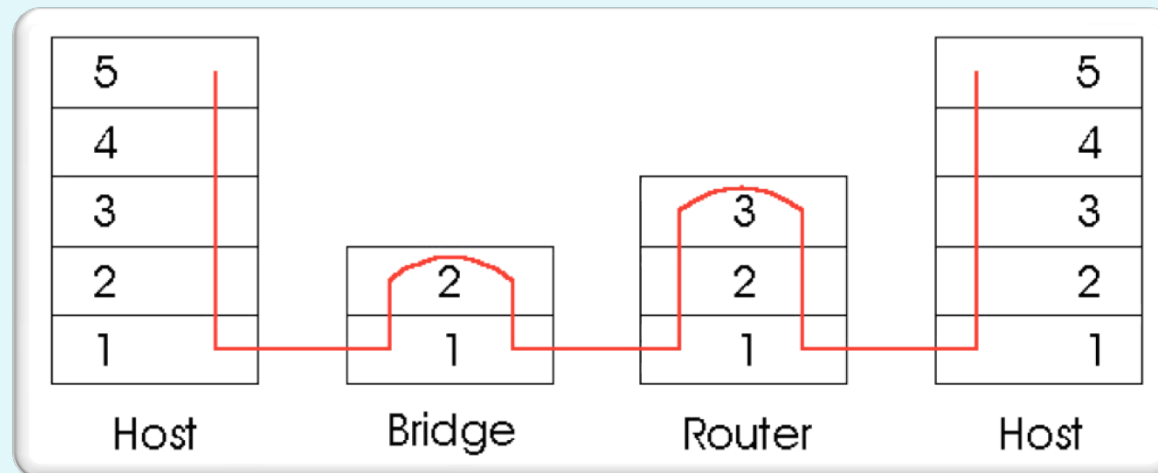
- Use of routing protocols (RIP, OSPF, BGP)
- *forwarding* datagrams between the input and output ports



Network layer:

Comparison of active equipment

- **router:**
 - device that works on the NETWORK layer
 - maintains arp tables, perform directional algorithms
- **switch:**
 - Device that works on DATA LINK layer,
 - maintains the switching table, perform filtration and network detection
- **hub:**
 - device that operates at the PHYSICAL layer, it is no longer in use



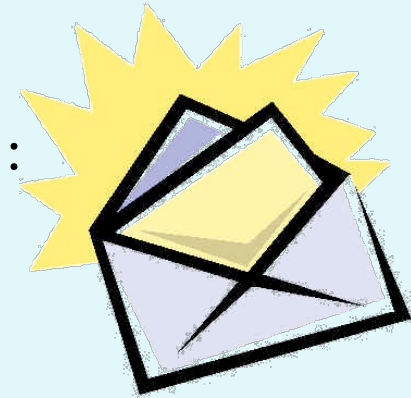
Network layer: IPv4

- Protocol on network (3.) layer OSI model
- **IPv4 address** is 32-bit address interface. Example :

11000001 00000010 00000001 01000010

OR

193.2.1.66



- **Subnetwork** is a crowd of IP addresses that are accessible among each other without the intercession of the router. Mask (32 bits) provides part of the IP address that represents the subnet address. example:

11111111 11111111 11110000 00000000 (255.255.255.240)

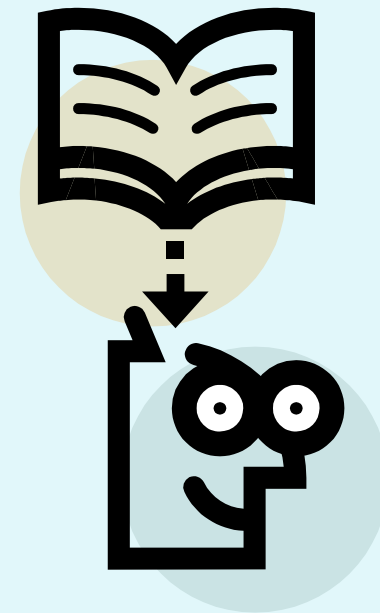
means that the first 20 bits of the IP address represents the network address and the remaining 12 bits are address of the interface.

Network layer: Exercise!

- The IP address of some interface and mask of the subnetwork are given
193.90.230.25 /20

What is the address of the subnetwork?

What is address of interface?



Network layer: IPv6

- **Advantages :**
 - larger address space : 128 bites
 - Quick direction and intercession and QoS is enabled by the format of the head, there is no fragmentation,
 - The implementation of IPSec within IPv6 is obligatory
- **Address :** consisting of 64 bits for the subnet ID + 64 bites for interface ID

```
0010000111011010 0000000011010011 0000000000000000 0010111100111011  
0000001010101010 0000000011111111 1111111000101000 1001110001011010
```

Written hexadecimal, separated by colons

21DA:00D3:0000:0000:02AA:00FF:FE28:9C5A or(without leading zeros)

21DA:D3:0:0:2AA:FF:FE28:9C5A or(omit blocks of zeros)

21DA:D3::2AA:FF:FE28:9C5A

Network layer:

Comparison of IPv4 and IPv6

IPv4 Header

0	4	8	12	16	20	24	28	31
Version	IHL	Type of Service	Total Length					
Identification				Flags	Fragment Offset			
Time to Live		Protocol		Header Checksum				
Source Address								
Destination Address								

IPv6 Header

0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	63
Version	Traffic Class		<i>Flow Label</i>					Payload Length			Next Header	Hop Limit				
Source Address																
Destination Address																

Network layer:

IPv6 - types of addressing



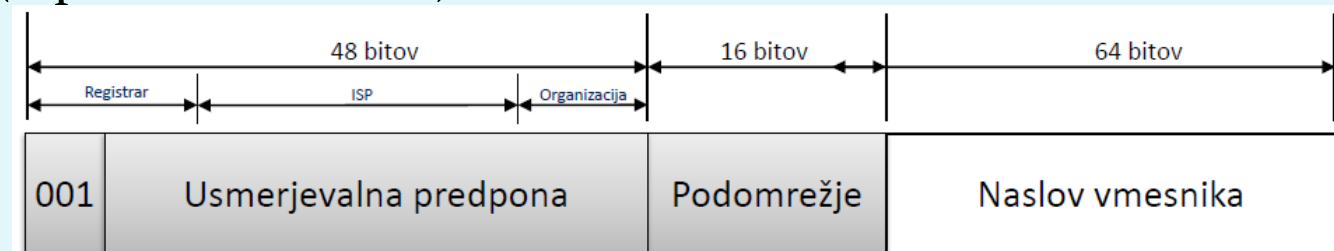
- **UNICAST:**
addressing each network interface
- **MULTICAST:**
addressing of a group of network interfaces, delivery to all interfaces in the crowd
- **ANYCAST:**
is the address of the crowd of the interfaces, the delivery is performed to one of the interfaces of the crowd(the closest one?)

Each interface can have multiple addresses of various types.
(BROADCAST addresses – in IPv6 they are no longer there!)

Network layer:

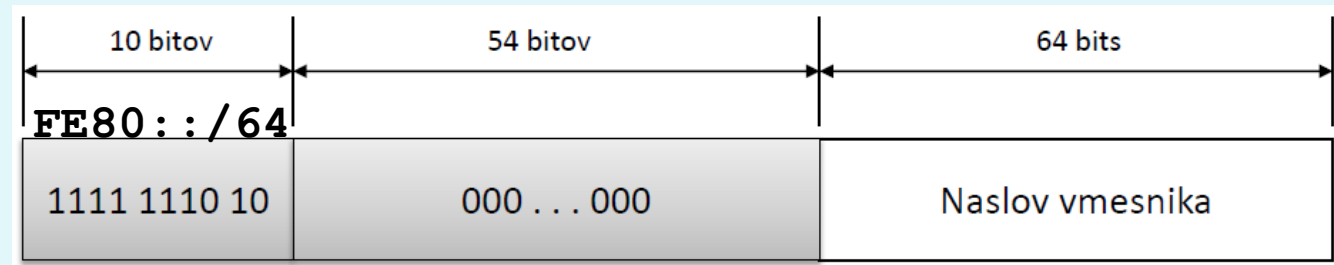
IPv6 - types of unicast addresses

1.) **global unicast** (= public addresses)



2.) **specific addresses** (localhost `::1`, undefined `o::o`, IPv4 addresses)

3.) **link-local addresses** (within 1 connection, adhoc network)



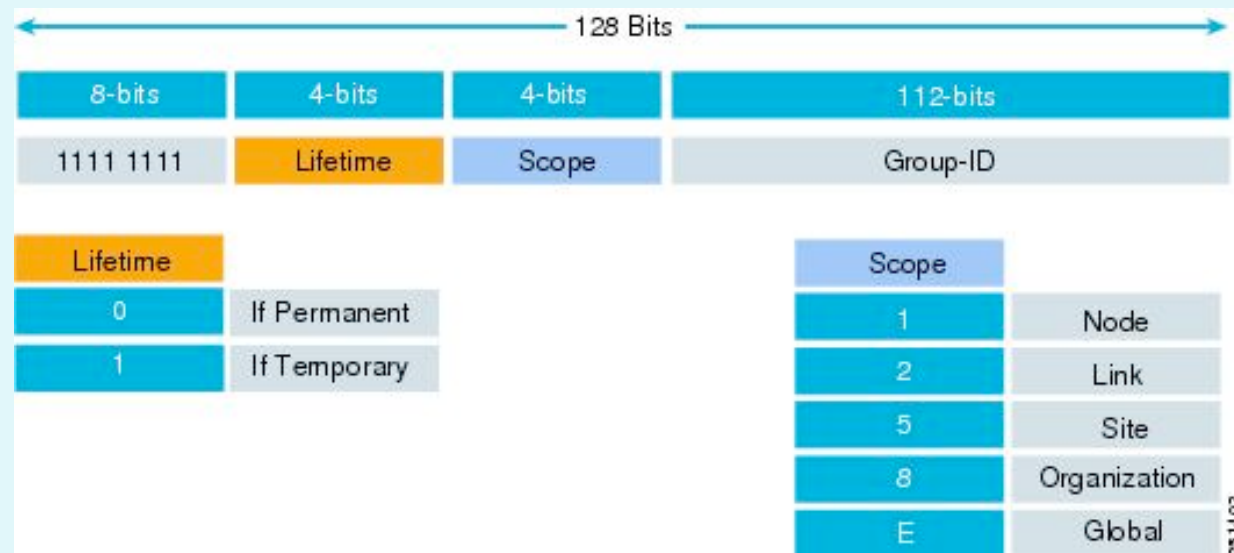
4.) **site-local** (Private addresses, within the org. they are not performed, `FEC0::/10`)

5.) **unique-local** (private addresses, allocated by the registrar, they're better structured, `FC00::/7`)

Network layer:

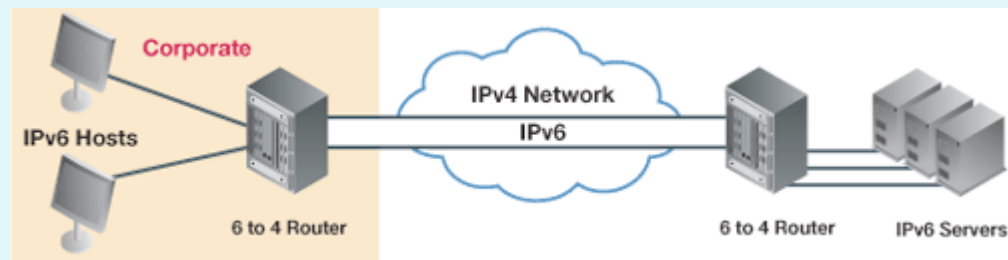
IPv6 – distribution (*multicast*)

- 1.) FF02::1 (link local: all interfaces)
- 2.) FF02::2 (link local all routers)
- 3.) address structure :



Network layer: IPv6 in IPv4 networks

- 1.) **dual-stack**: routers know IPv4 and IPv6
- 2.) **tunneling**: IPv6 packet packed in one or more IPv4 packets as data.



Network layer: Routing



- **MODES**

- static / dynamic (consideration of conditions in the network)
- centralized / distributed (according to the knowledge of the whole network status)
- one way / by multiple pathways

- **IMPLEMENTATION :**

- With the distance vector (RIP, IGRP, EIGRP)
- according to the network status (OSPF, IS-IS)

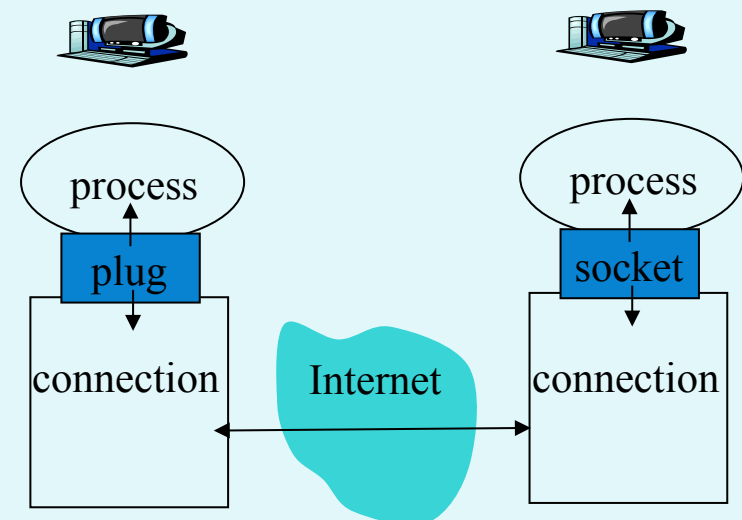
Transport layer: Functionalities

- **Task :**

- Receiving a message from application
- Assembling segments in the message to the network layer
- Transferring to application layer

- **plug**



- interface between the transport and application layer,
- We address the process with the IP number and the port number (www: 80, SMTP: 25, DNS: 53, POP3: 110).



Transport layer:

Connection and connectionless oriented

- **Connection and connectionless oriented communication**

- TCP and UDP, and other protocols  
- establishment, transmission, demolition – connection

Validation

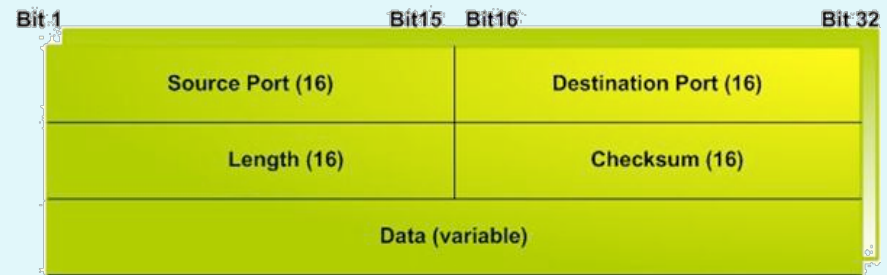
- in the protocol(TCP)
- in the application(UDP)
- directly(ACK and NACK)
- indirectly(only ACK, we conclude according to the number of packages)
- Simultaneous confirmation: the next package is sent only after the receipt of the confirmation
- Fluent sending: no waiting for the confirmation

Transport layer: TCP and UDP

The TCP Segment Format



The UDP Segment Format

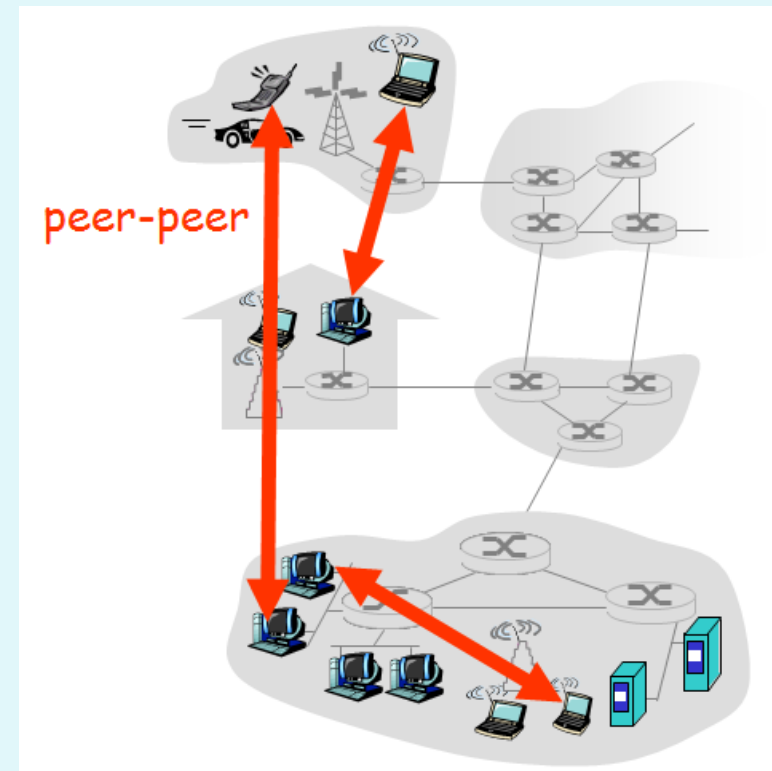


Application layer:

- **Classic services – client-server**
 - telnet, ssh; rdesktop
 - ftp, sftp
 - WWW in HTTP,
 - SMTP, POP₃, IMAP, MAPI
 - DNS,
 - SNMP, LDAP, RADIUS, ...
 - ...

Application layer:

- **newer services– P2P:**
 - Communication of two random final systems
 - servers are not constantly switched on,
 - broken connections/ changes to IP addresses,
 - examples: BitTorrent, Skype



Network and transport layer:

From the past to the future

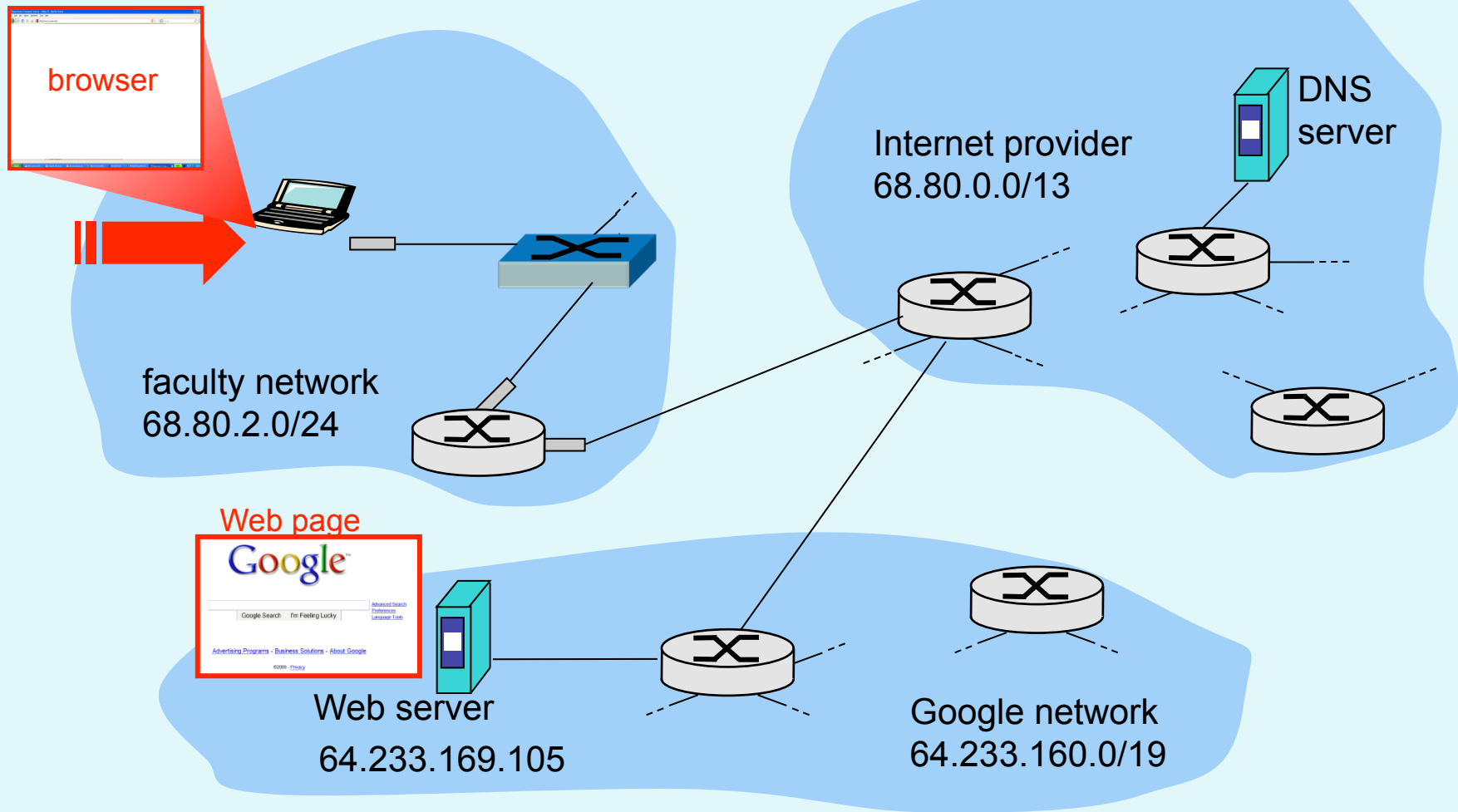
- **Problem:** lack of IPv4 addresses
 - The efficiency of private address spaces
 - NAT gateways - usually at the same time firewalls too
 - simply in client-server systems
 - In P2P we need a copy address in the outer world

In IPv6 NAT gateways are not required

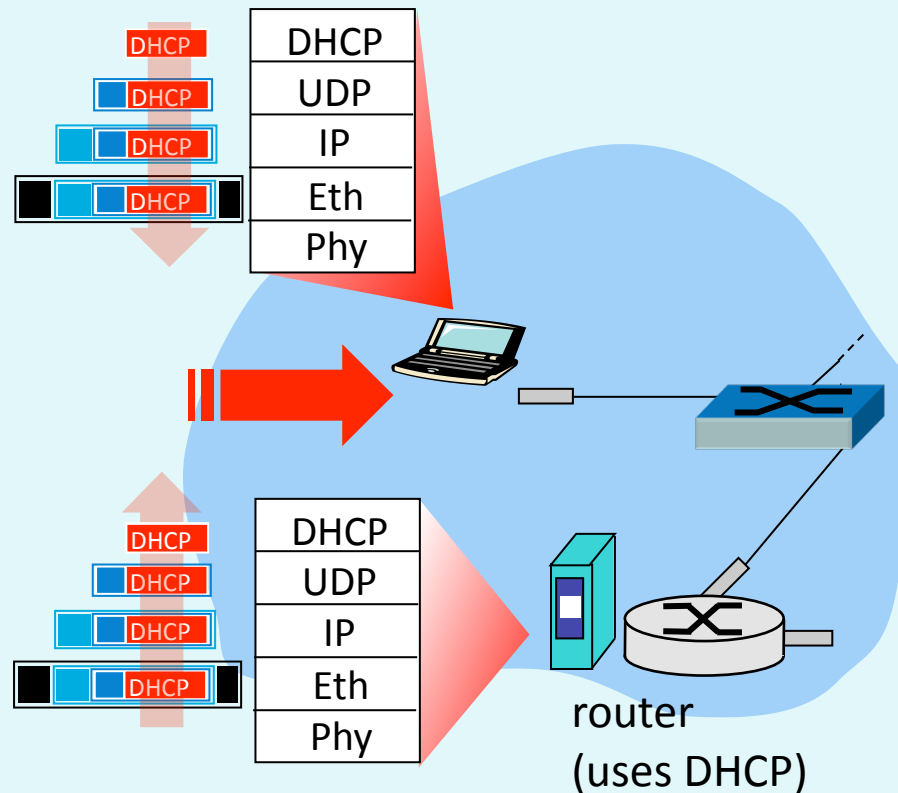


An example of communication

Example of communication: Web browsing

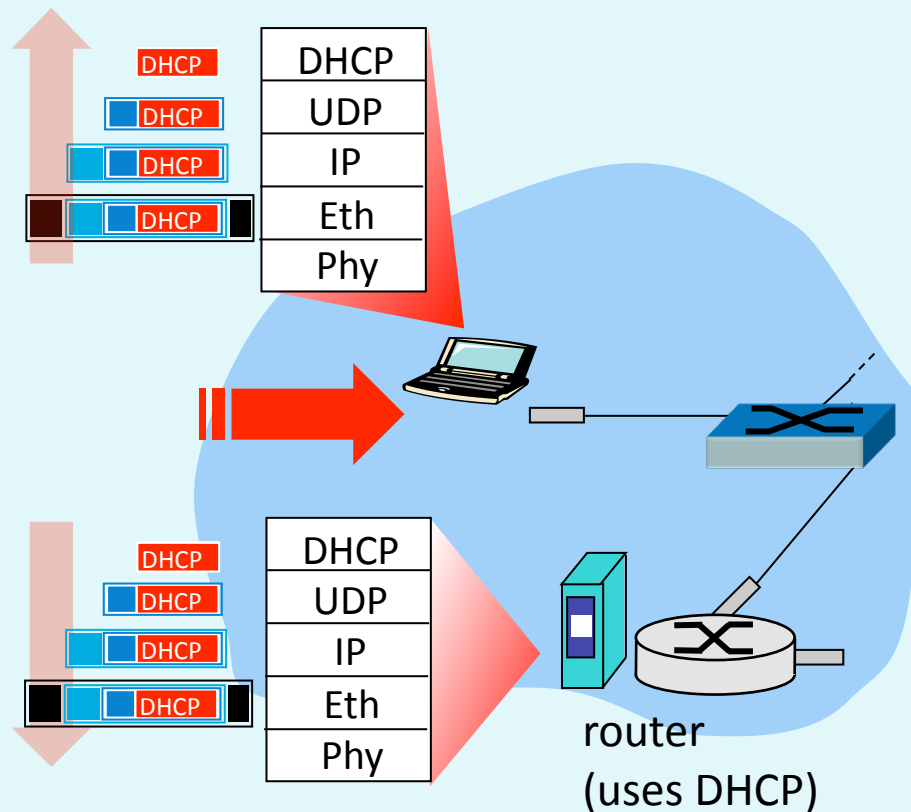


Example of communication: Web browsing



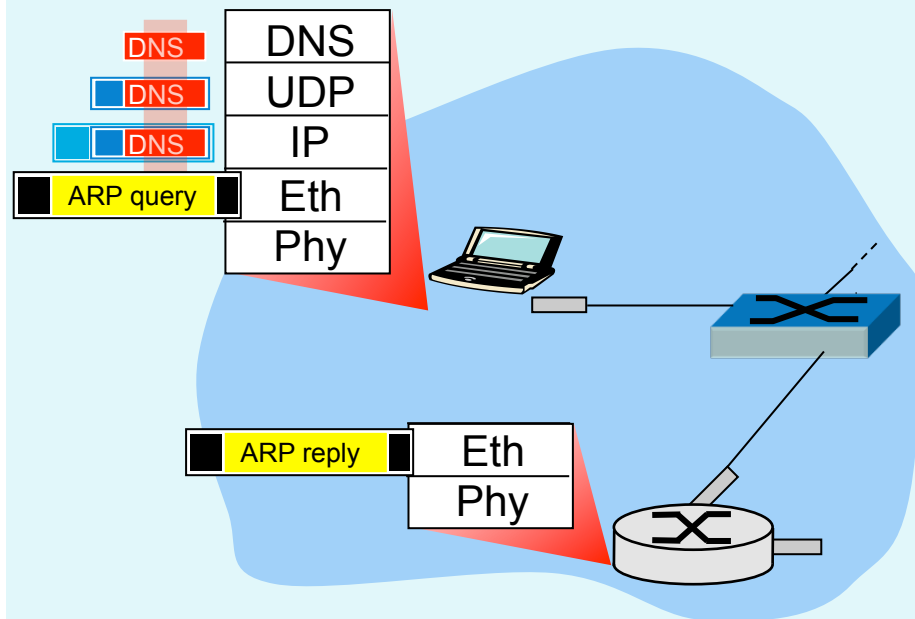
- When it connects to the network, the laptop needs an IP address, and the data of connection and DNS server: It uses DHCP.
- The request DHCP encapsulates: UDP -> IP -> 802.1 Ethernet
- ethernet frame transmits(broadcast) itself to the network, it is received by the router, which carries out the DHCP server's task
- DHCP server reads the content of DHCP request

Example of communication: Web browsing



- DHCP answers to the client (laptop) with the DHCP ACK package, which contains its IP address and the addresses of the gateway and DNS server
- The answer encapsulates the DHCP server (router) and passes it on to the client which decapsulates.
- The DHCP client receives the answer DHCP ACK
- The result: The client is ready for communication

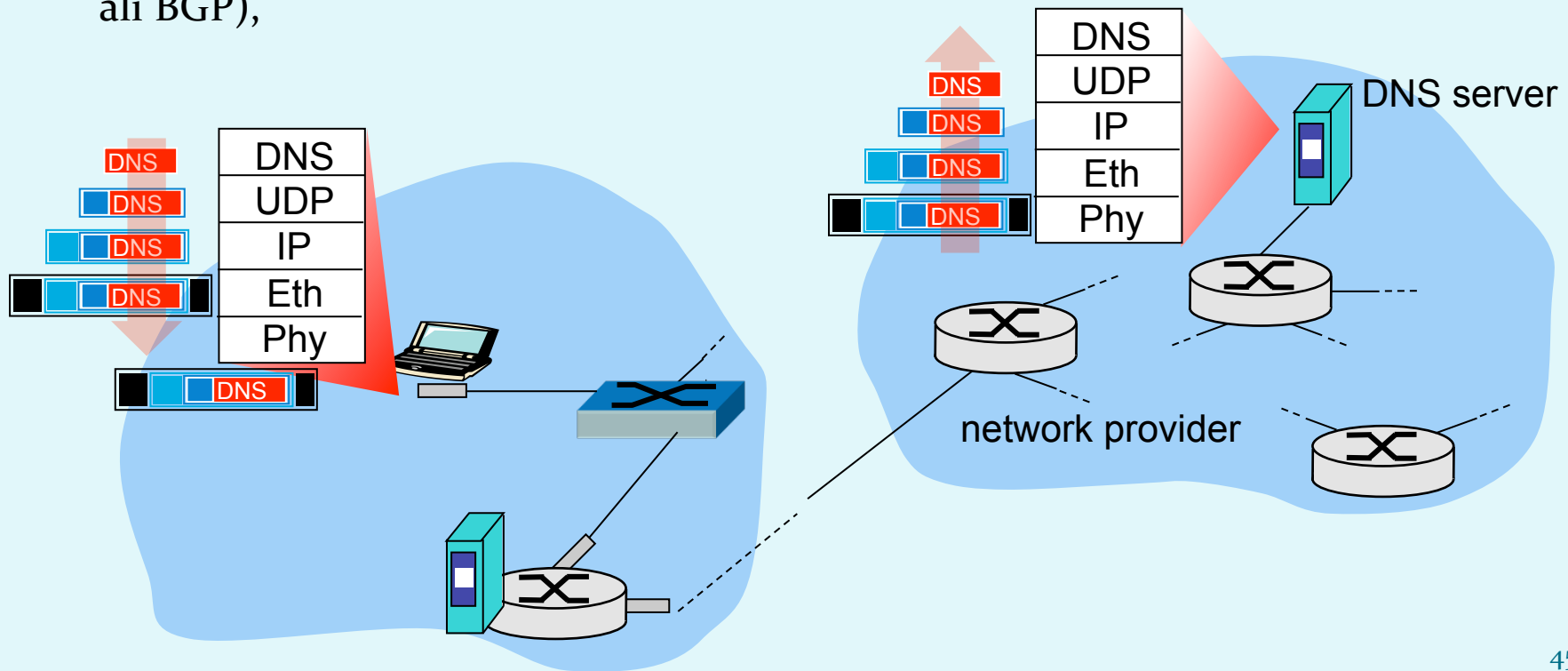
Example of communication: Web browsing



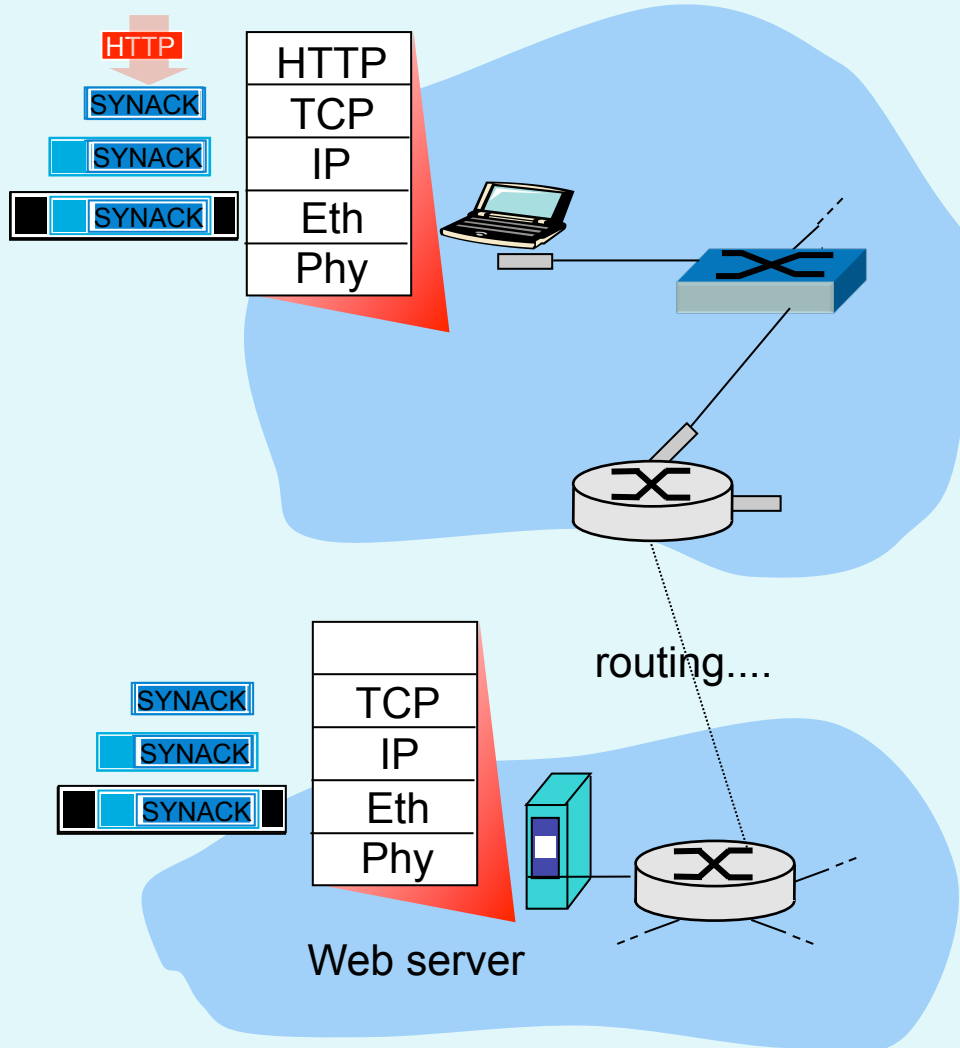
- Before sending off the http request we need the IP of the server
www.google.com: use DNS
- Encapsulation of the DNS request: UDP ->IP -> Ethernet. We need MAC address of the router: use ARP
- We send off the ARP request, the router answers with the ARP answer, which keeps its MAC address
- The client now knows the MAC address of the gateway, which can send the DNS request to it.

Example of communication: Web browsing

- The IP datagram with **DNS request** is passed on the router.
- IP datagram is passed on the **DNS server**, which is in the network of internet provider (RIP, OSPF, IS-IS ali BGP),
- DNS server **decapsulates** the request and sends to user the IP address of the network server www.google.com

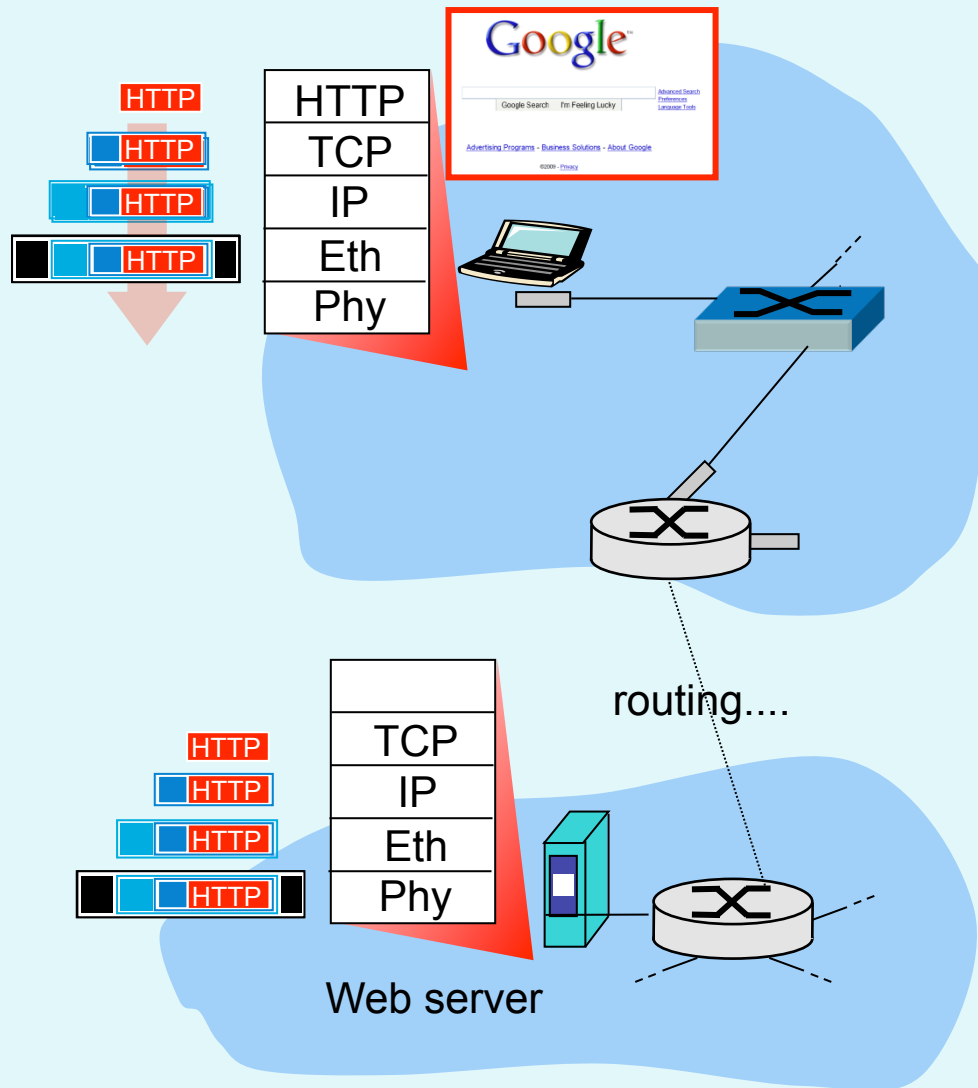


Example of communication: Internet browsing



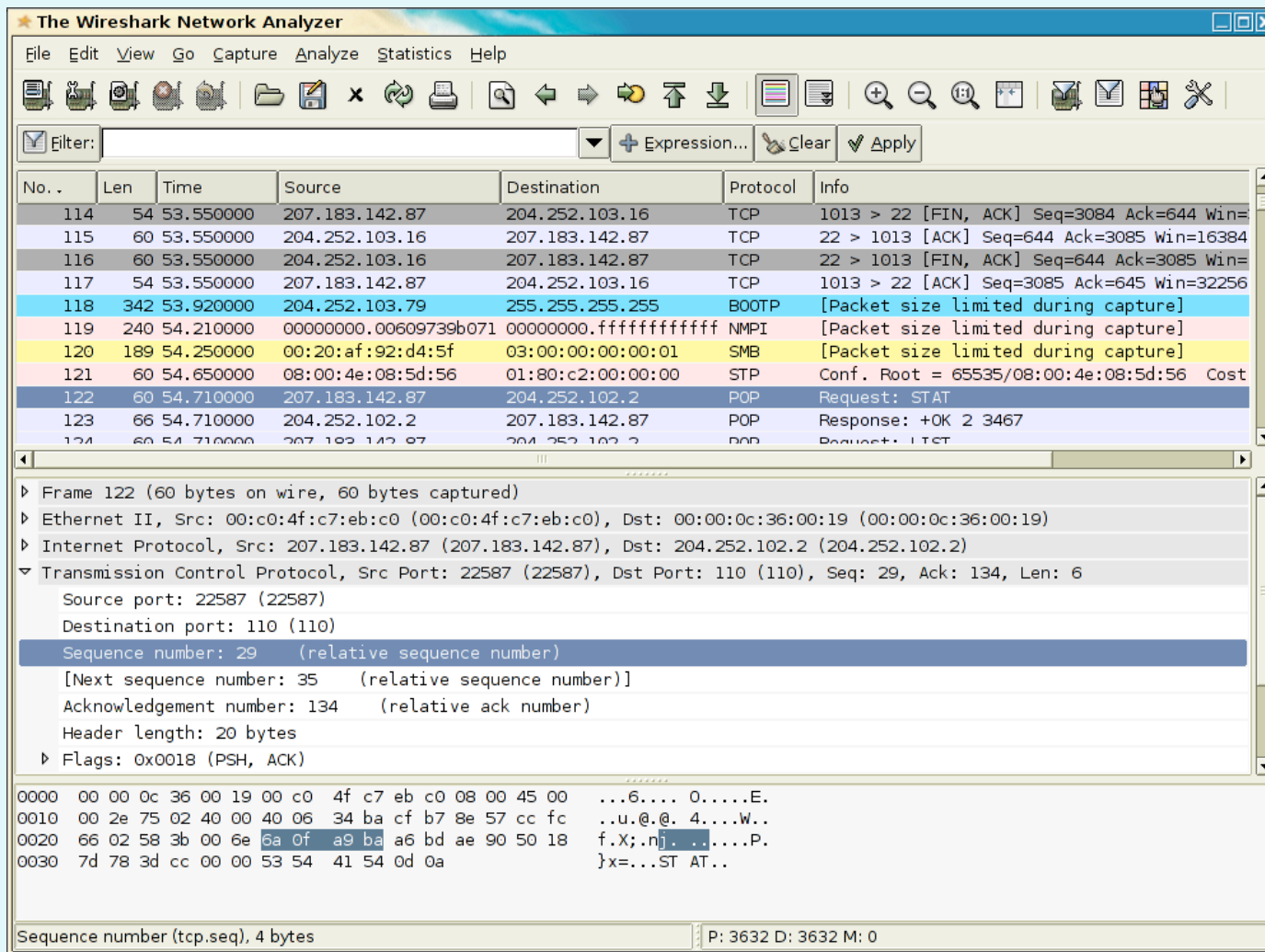
- To send the **HTTP request**, to the client first addresses the **TCP plug** of the web server
- **TCP SYN** segment direct itself through the network to the web server
- The web server answers with **TCP SYNACK** (confirmation of the handshake),
- The TCP connection is now established!

Example of communication: Internet browsing



- **HTTP request** is sent to the **TCP plug** of the web server,
- **IP datagram**, which contains the internet request for the website www.google.com is directed to the web server
- The internet server answers with **HTTP REPLY**, which contains the contents of the webpage
- The IP datagram with the webpage is directed to the client,
- **WWW page is finally shown!**

Capturing data from the network



The screenshot displays the Wireshark Network Analyzer interface. The main window shows a list of captured packets with columns for No., Len, Time, Source, Destination, Protocol, and Info. Packet 122 is selected, and its details are shown in the lower pane. The details pane shows the following information:

- Frame 122 (60 bytes on wire, 60 bytes captured)
- Ethernet II, Src: 00:c0:4f:c7:eb:c0 (00:c0:4f:c7:eb:c0), Dst: 00:00:0c:36:00:19 (00:00:0c:36:00:19)
- Internet Protocol, Src: 207.183.142.87 (207.183.142.87), Dst: 204.252.102.2 (204.252.102.2)
- Transmission Control Protocol, Src Port: 22587 (22587), Dst Port: 110 (110), Seq: 29, Ack: 134, Len: 6
 - Source port: 22587 (22587)
 - Destination port: 110 (110)
 - Sequence number: 29 (relative sequence number)
 - [Next sequence number: 35 (relative sequence number)]
 - Acknowledgement number: 134 (relative ack number)
 - Header length: 20 bytes
 - Flags: 0x0018 (PSH, ACK)

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```

0000 00 00 0c 36 00 19 00 c0 4f c7 eb c0 08 00 45 00  ...6... 0....E.
0010 00 2e 75 02 40 00 40 06 34 ba cf b7 8e 57 cc fc  ..u.@.@. 4....W..
0020 66 02 58 3b 00 6e 6a 0f a9 ba a6 bd ae 90 50 18  f.X;.n]. ....P.
0030 7d 78 3d cc 00 00 53 54 41 54 0d 0a                }x=...ST AT..
    
```

At the bottom of the interface, the status bar indicates: Sequence number (tcp.seq), 4 bytes | P: 3632 D: 3632 M: 0

Capturing data from the network: DHCP example

request

```
Message type: Boot Request (1)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP Request
Option: (61) Client identifier
    Length: 7; Value: 010016D323688A;
    Hardware type: Ethernet
    Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Option: (t=50,l=4) Requested IP Address = 192.168.1.101
Option: (t=12,l=5) Host Name = "nomad"
Option: (55) Parameter Request List
    Length: 11; Value: 010F03062C2E2F1F21F92B
    1 = Subnet Mask; 15 = Domain Name
    3 = Router; 6 = Domain Name Server
    44 = NetBIOS over TCP/IP Name Server
.....
```

response

```
Message type: Boot Reply (2)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 192.168.1.101 (192.168.1.101)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 192.168.1.1 (192.168.1.1)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP ACK
Option: (t=54,l=4) Server Identifier = 192.168.1.1
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
Option: (t=3,l=4) Router = 192.168.1.1
Option: (6) Domain Name Server
    Length: 12; Value: 445747E2445749F244574092;
    IP Address: 68.87.71.226;
    IP Address: 68.87.73.242;
    IP Address: 68.87.64.146
Option: (t=15,l=20) Domain Name = "hsdl.ma.comcast.net."
```

Network security



Network security

- **Is an area that :**
 - analyzes the potential attacks on systems,
 - Plans the techniques of the defence from the attacks,
 - Forms safe architectures, which are resistant to the invasions
- **The Internet wasn't build considering the safety!**
 - First the vision of the internet was: "This was a group of people, that trust each other and are connected to a common network"
 - At the making of the protocol, the manufacturers made it with the methodology of „ patching”,
 - The safety mechanisms should be considered at all layers of OSI model

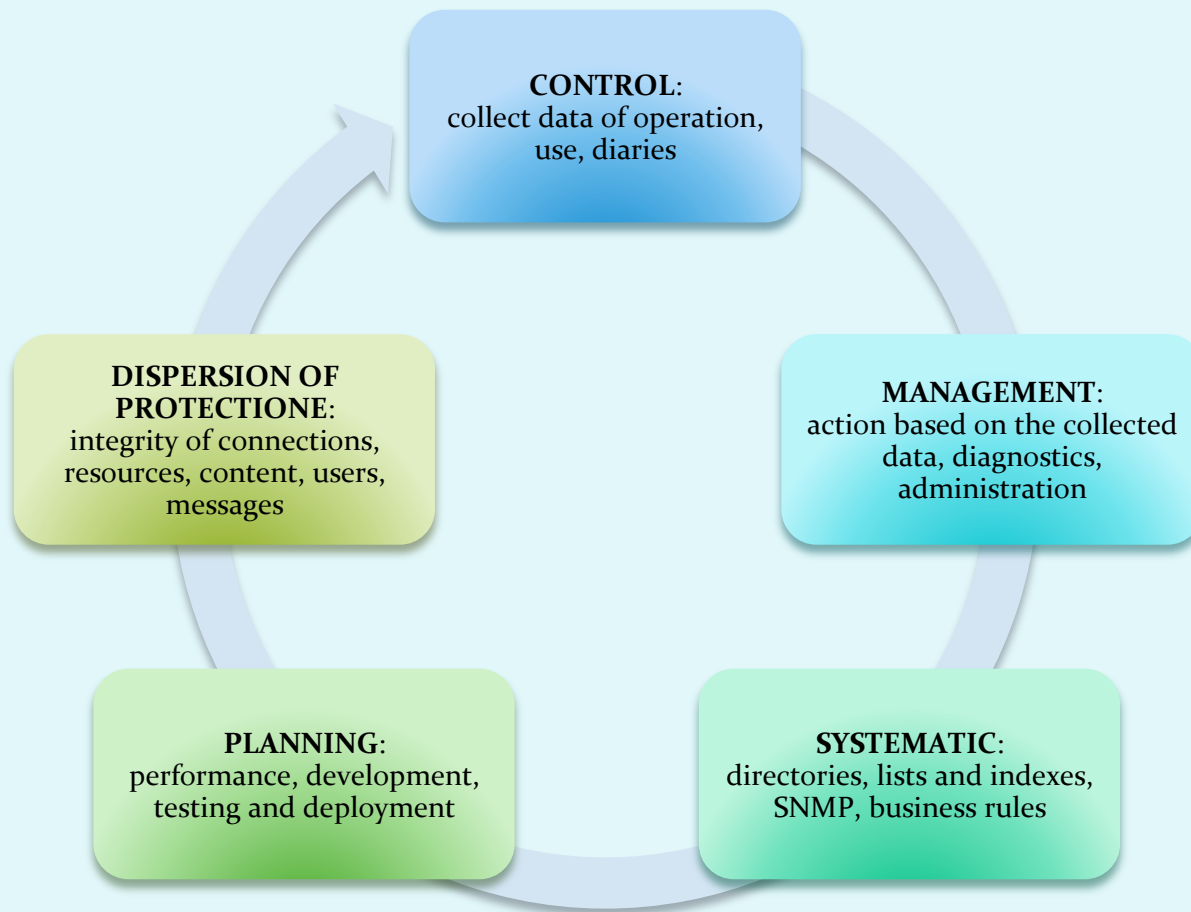
How can the intruder harms the system?

There are many possible approaches and techniques!

- **tapping** : intercepting of messages,
- Active **forging** of messages in some communication,
- **Theft the identity**(impersonization): forges the original address or any other content of the package
- **Taking-over the connection (hijacking)**: removes the real sender or receiver from the communication and takes-up his role
- **Disabling of the service providence (denial of service)**: Disables the use of the regular service (ex. With overloading it)



Security: ensure the reliability



Elements of safe communication:

- **Confidentiality** – who is allowed to read? (encryption)
- **Authentication** – prove that it is really you (identification, tell who you are, without proof)
- **Availability and control of access** – prevention of illegitimate use of sources (authorization – finding out if you can do something, accounting – who used what)
- **Integrity of the message** – was it changed during the transmission?
- **Disabling of disclaiming** (*nonrepudiation*) you really sent/received it

- Practice showed:
 - firewalls, intrusion detection systems,
- Safety on application, transport, network and data link layer

Authentication

We make sure of the true identity of the person – co-speaker.

APPROACHES :

- *Challenge-response,*
- We trust the third side,
- Authentication with the system of public keys



Confidentiality of messages: crypting (concealing) the content

This is a form of defence from **passive** intruders (eavesdroppers) and **active** intruders (forgers).

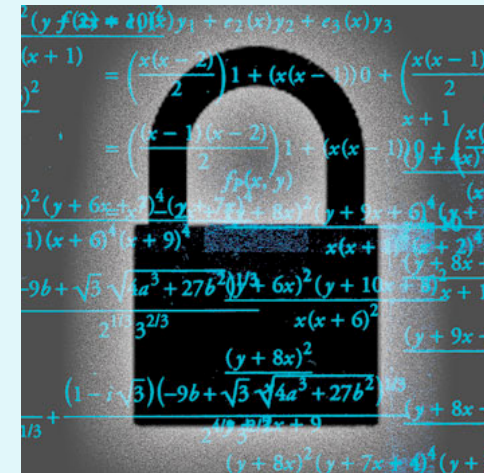
We **encrypt message P with the E key** – we get **cryptogram E(P)**. We process the cryptogram **E(P)** into the original for with the **D()** key and what we get is original message **D(E(P))-P**.

Different methods::

- **Substitution** (change of symbols) / **transposition** (sequence of the symbols)
- **Symmetric** (**E=D**, ex. DES, AES) / **asymmetric**(**E≠D** , ex. RSA, ECC)

Types of cryptography

- Cryptography that uses keys:
 - Algorithm is usually known to everybody,
 - Only the keys are secretive
 - encryption : hiding the content
 - Crypto-analysis(„crashing” of the code)
- Cryptography with public key
 - $E() \neq D()$: two keys– public and private
- Symmetric cryptography
 - $E() = D()$: only one key
- Thickening functions – they are not cryptography. Don't use keys. How can they be useful?

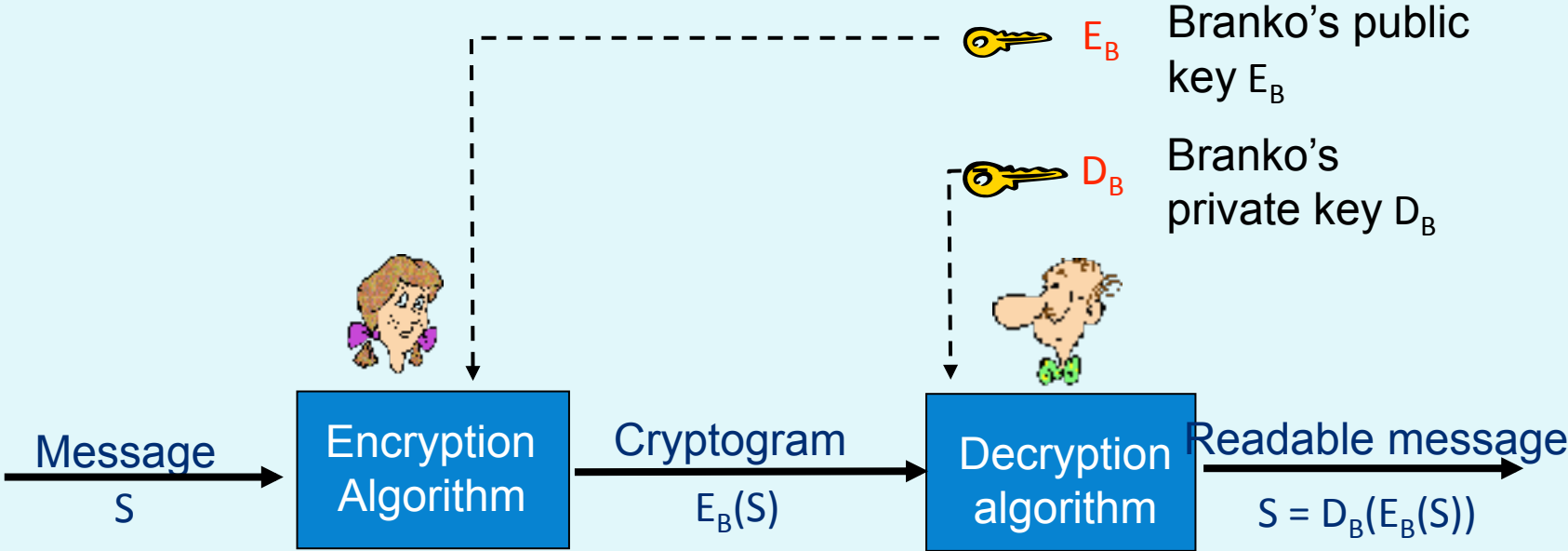


Cryptography with public keys

- The algorithms for encrypting with public keys are asymmetric, $E \neq D$
- Key E and D must satisfy the following requirements encryption of message S :
 1. $D(E(S)) = D(E(S)) = S$
 2. From known E and $E(S)$ it must be impossible to figure out D
 3. From E it must be very hard / impossible to figure out D
- The most known algorithm is **RSA**(Rivest, Shamir, Adelman). RSA uses big prime numbers to define D and E ; the procedure of encrypting/decrypting is the same as calculating the mod of divide by the product of these two numbers.

Problem: distribution of keys, slowness

Cryptography with public keys



Why is RSA safe?

- Let's say that we know the public key of some person (defined by a pair of numbers (n, e)). To figure out the private key we have to know the denominators of the number n . But searching the denominators of a large number is hard or impossible with current computational capacities.
- How to find big enough prime numbers?
 - We carry out “guessing” for several times: we generate a large number and test it, if it is a prime number,
To test the prime numbers there exist efficient algorithms.

Integrity

- **The integrity of users** : Proves who sent the message and that the message is read only by the real receiver. We encrypt the message S , which is sent by A to B

$$E_B(D_A(S)) = XXX$$

$$\text{and decrypt: } D_B(XXX) = D_B(E_B(D_A(S))) = D_A(S); E_A(D_A(S)) = S$$

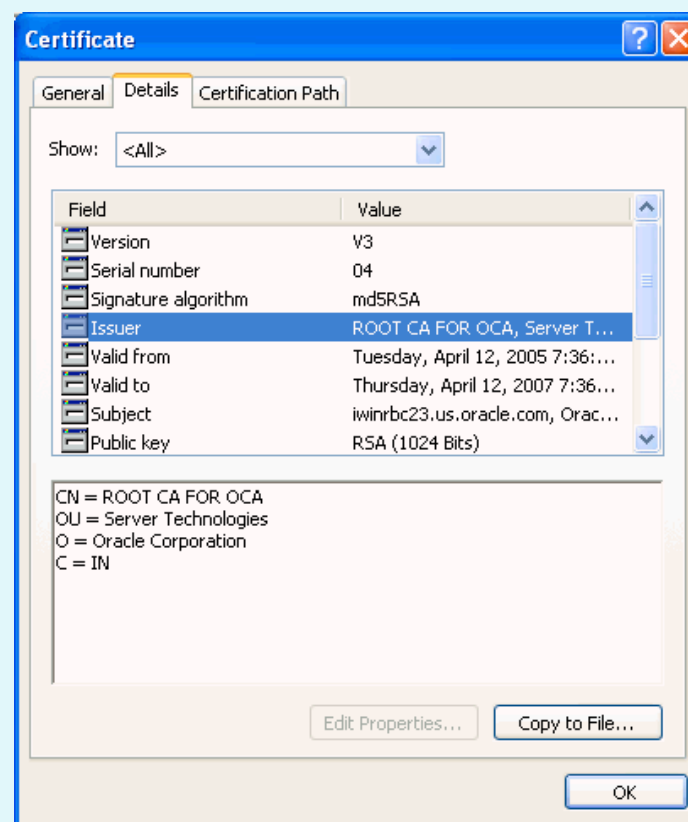
- **The integrity of message**: proves that the message (also not crypted!) hasn't been changed. To achieve that we use thickening functions, which calculate the signature of the message $SIG(S)$. We sign this value with the mechanism of electronic signing

$$D_A(\text{sig}(S)) = sss$$

And we send SSS along with the (encrypted) original message xxx : (xxx, sss) . The receiver decrypts XXX into S , recalculates the $\text{sig}(S)$ and checks if $SSS = \text{sig}(S)$

Certificates

- System PKI includes certification authorities, which issue, save and cancel the certificates.
- Certificates are defined by the standard X.509 (RFC 2459)
- The certificate contains:
 - The name of the Issuer,
 - The name of the person, the address, the domain name and other personal information,
 - The owners public key,
 - The digital signature(signed by the private key of the issuer)



Next time we move on!

- connect a computer to to the network
- boot your computer : protocols DHCP and BOOTP
- architecture server- client,
- protocol: operation, its functions,
- protocol trace

