Computer Networks

Lecture 9

Network and transport layers,

IP, TCP, UDP protocols

Network layer

The Network layer, or OSI Layer 3, provides services to exchange the individual pieces of data over the network between identified end devices.

Layer 3 uses four basic processes:

- Addressing
- Encapsulation
- Routing
- Decapsulation

Network Layer Protocols

Protocols implemented at the Network layer that carry user data include:

- Internet Protocol version 4 (IPv4)
- Internet Protocol version 6 (IPv6)
- Novell Internetwork Packet Exchange (IPX)
- AppleTalk
- Connectionless Network Service (CLNS/DECNet)

The Internet Protocol (IPv4 and IPv6) is the most widely-used Layer 3 data carrying protocol.

IP protocol

The Internet Protocol (IP) is the principal communications protocol in the Internet protocol suite for relaying datagrams across network boundaries. Its routing function enables internetworking, and essentially establishes the Internet.

IP, as the primary protocol in the Internet layer of the Internet protocol suite, has the task of delivering packets from the source host to the destination host solely based on the IP addresses in the packet headers.

For this purpose, IP defines packet structures that encapsulate the data to be delivered. It also defines addressing methods that are used to label the datagram with source and destination information.

IP protocol

IPv4 basic characteristics:

- Connectionless - No connection is established before sending data packets.
- Best Effort (unreliable) - No overhead is used to guarantee packet delivery.
- Media Independent - Operates independently of the medium carrying the data.

IPv4 header

Version	IHL	Service type	Length
ID	Flags	Fragment Offset	
TTL	Protocol	Checksum	
Source address			
Destination address			
TCP data			

Header fields contain binary values that the IPv4 services reference as they forward packets across the network.

IPv4 packet

IPv4 packet consists of:

- Header
- Data

Header size is 20 - 60 bytes.

IPv4 packet size is:

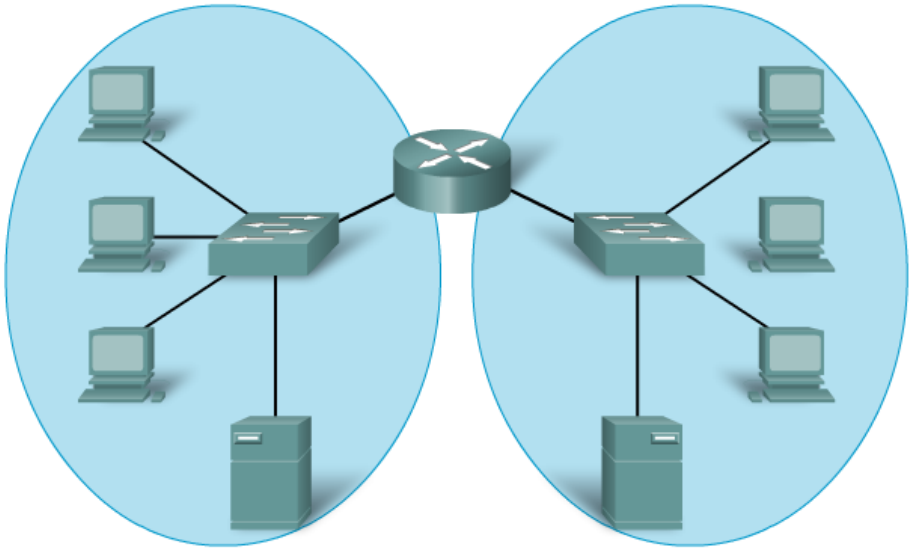
- Min 20 bytes (20-bytes header + 0 bytes data)
- Max 65.535 bytes.

IP packet can be fragmented, therefore packet ID and Fragment Offset are used.

Dividing Networks

Common issues with large networks are:

- Performance degradation
- Security issues
- Address Managements

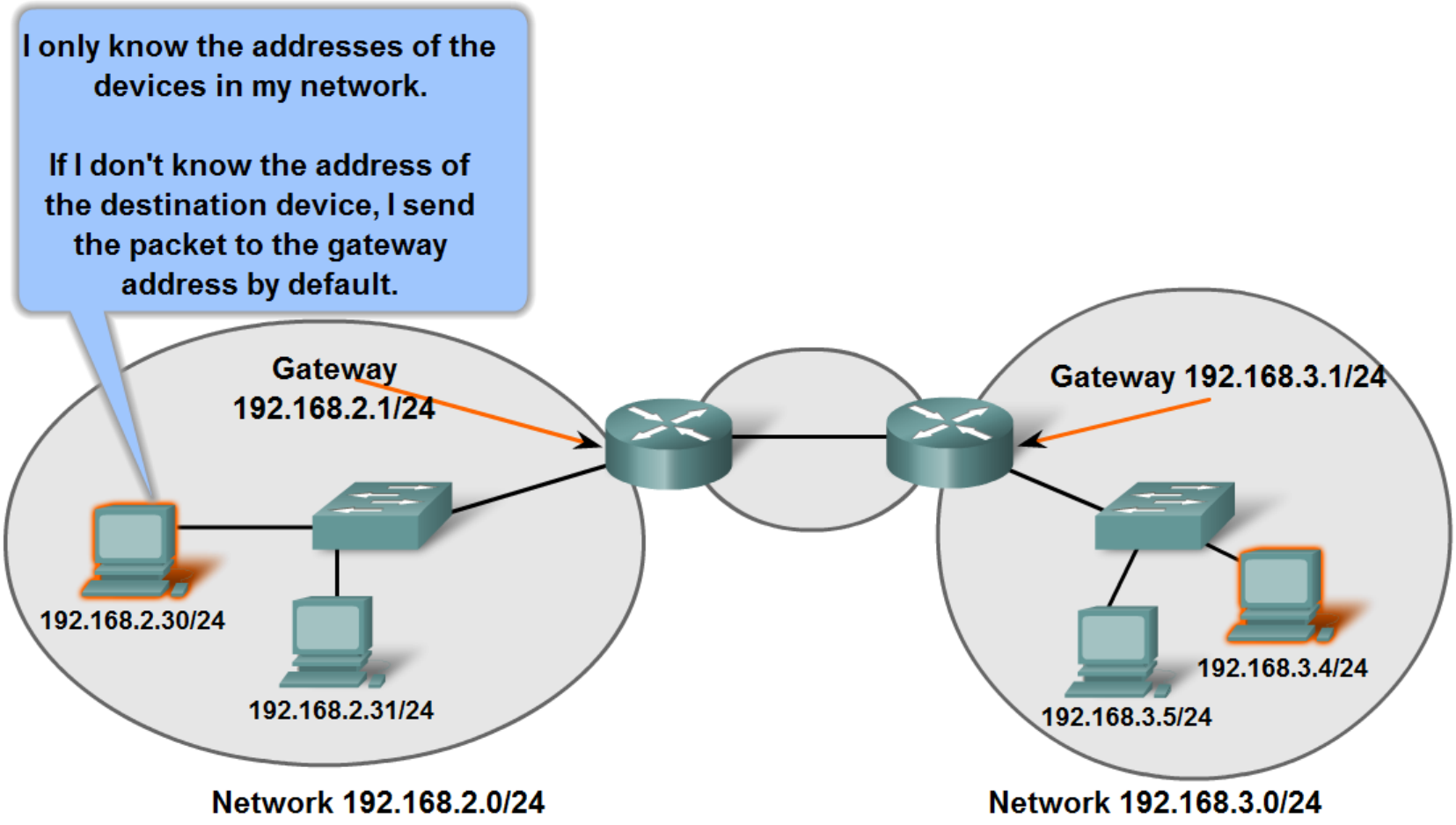


Routers

Within a network or a subnetwork, hosts communicate with each other without the need for any Network layer intermediary device. When a host needs to communicate with another network, an intermediary device, or router, acts as a gateway to the other network.

As a part of its configuration, a host has a default gateway address defined. As shown in the figure, this gateway address is the address of a router interface that is connected to the same network as the host.

Routers


I only know the addresses of the devices in my network.
If I don't know the address of the destination device, I send the packet to the gateway address by default.
Gateway
192.168.2.1/24
Gateway 192.168.3.1/24
192.168.2.30/24
192.168.2.31/24
192.168.3.4/24
192.168.3.5/24
Network 192.168.2.0/24
Network 192.168.3.0/24

Addressing

Type of host addresses in TCP/IP networks:

- **MAC** (Ethernet, Token Ring, FDDI)
- **IP (IPv4; IPv6)**
- **FQDN**

Address Resolution Protocol (ARP) is used for resolution of network layer addresses into link layer addresses, a critical function in multiple-access networks..

DNS is used to translates domain names, which can be easily memorized by humans, to the numerical IP addresses needed for the purpose of computer services and devices worldwide.

IPv4 address

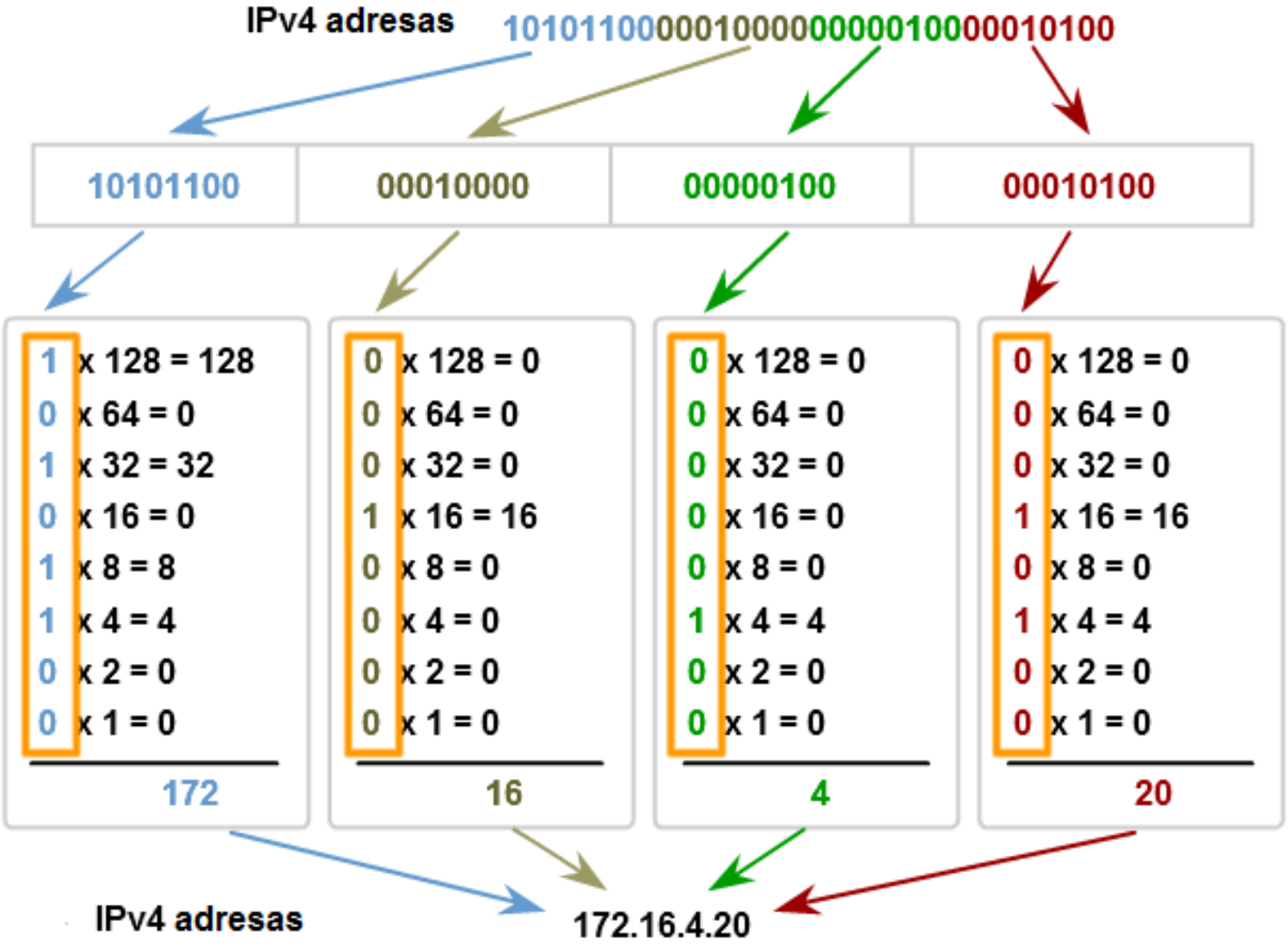
32-bit IPv4 address is hierarchical and is made up of two parts. The first part identifies the network and the second part identifies a host on that network.

Both parts are required for a complete IP address.

Example

192	.	168	.	10	.	1
11000000		10101000		00001010		00000001

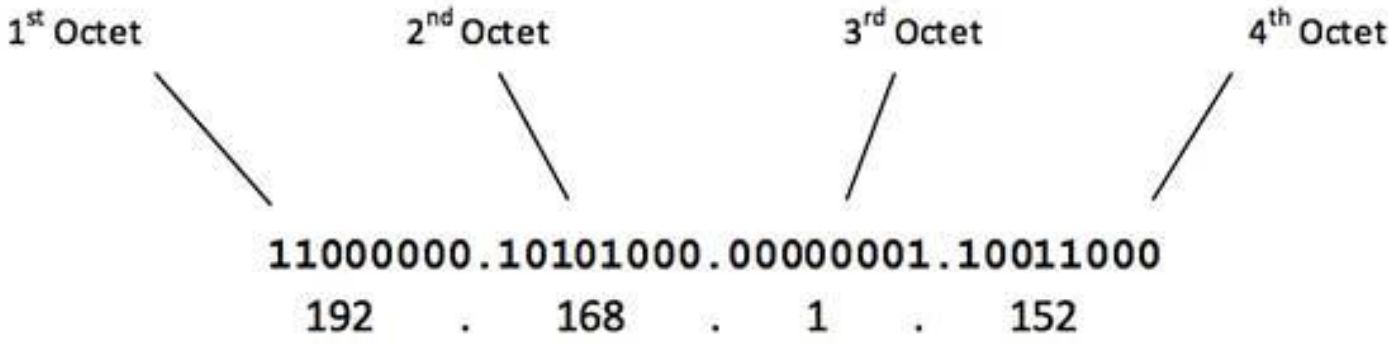
IP address


IPv4 adresas
10101100000100000000010000010100
10101100
00010000
00000100
00010100
1 x 128 = 128
0 x 64 = 0
1 x 32 = 32
0 x 16 = 0
1 x 8 = 8
1 x 4 = 4
0 x 2 = 0
0 x 1 = 0
172
0 x 128 = 0
0 x 64 = 0
0 x 32 = 0
1 x 16 = 16
0 x 8 = 0
0 x 4 = 0
0 x 2 = 0
0 x 1 = 0
16
0 x 128 = 0
0 x 64 = 0
0 x 32 = 0
0 x 16 = 0
0 x 8 = 0
1 x 4 = 4
0 x 2 = 0
0 x 1 = 0
4
0 x 128 = 0
0 x 64 = 0
0 x 32 = 0
1 x 16 = 16
0 x 8 = 0
1 x 4 = 4
0 x 2 = 0
0 x 1 = 0
20
IPv4 adresas
172.16.4.20

IPv4 address

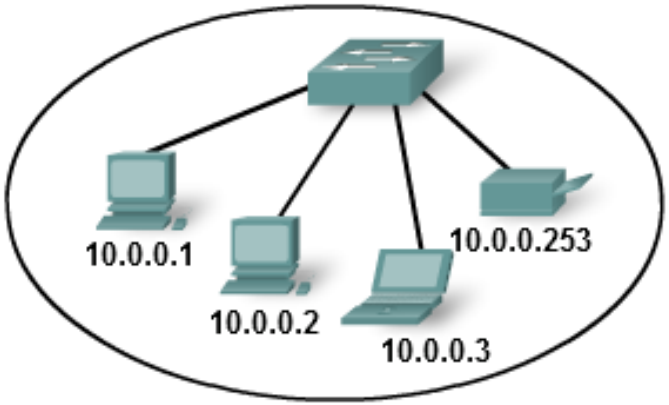
IP address type:

- Network address
- Host address
- Broadcast address



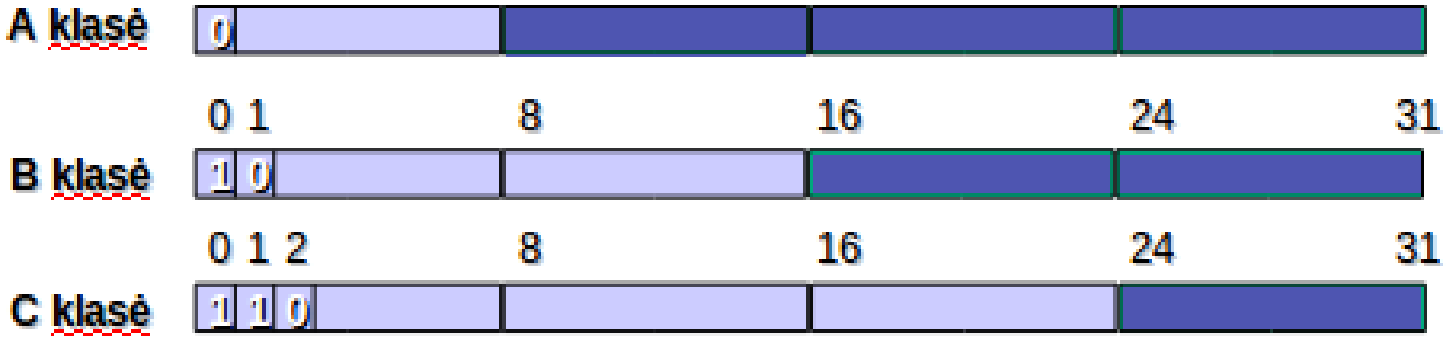
Example

	Tinklo dalis			Mazgo dalis
Tinklo adresas	10	0	0	0
	00001010	00000000	00000000	00000000
Transliacinis adresas	10	0	0	255
	00001010	00000000	00000000	11111111
Mazgo adresas	10	0	0	1
	00001010	00000000	00000000	00000001



IP address class

IPv4 Addressing system is divided into five classes (A,B,C,D,E) of IP addresses. All the five classes are identified by the first octet of IP Address.



IP address classes

Class	Bits	Network address (smallest)	Network address (biggest)	Number of networks	Number of hosts
A	0	1.0.0.0	126.0.0.0	126	2^{24} -2
B	10	128.0.0.0	191.255.0.0	16384	2^{16}-2
C	110	192.0.0.0	223.255.255.0	2097150	2^{8}-2
D	1110	224.0.0.0	239.255.255.255		Grupiniai adresai
E	11110	240.0.0.0	247.255.255.255		Rezerved

Subnet mask

- IP address **129.64.134.5**

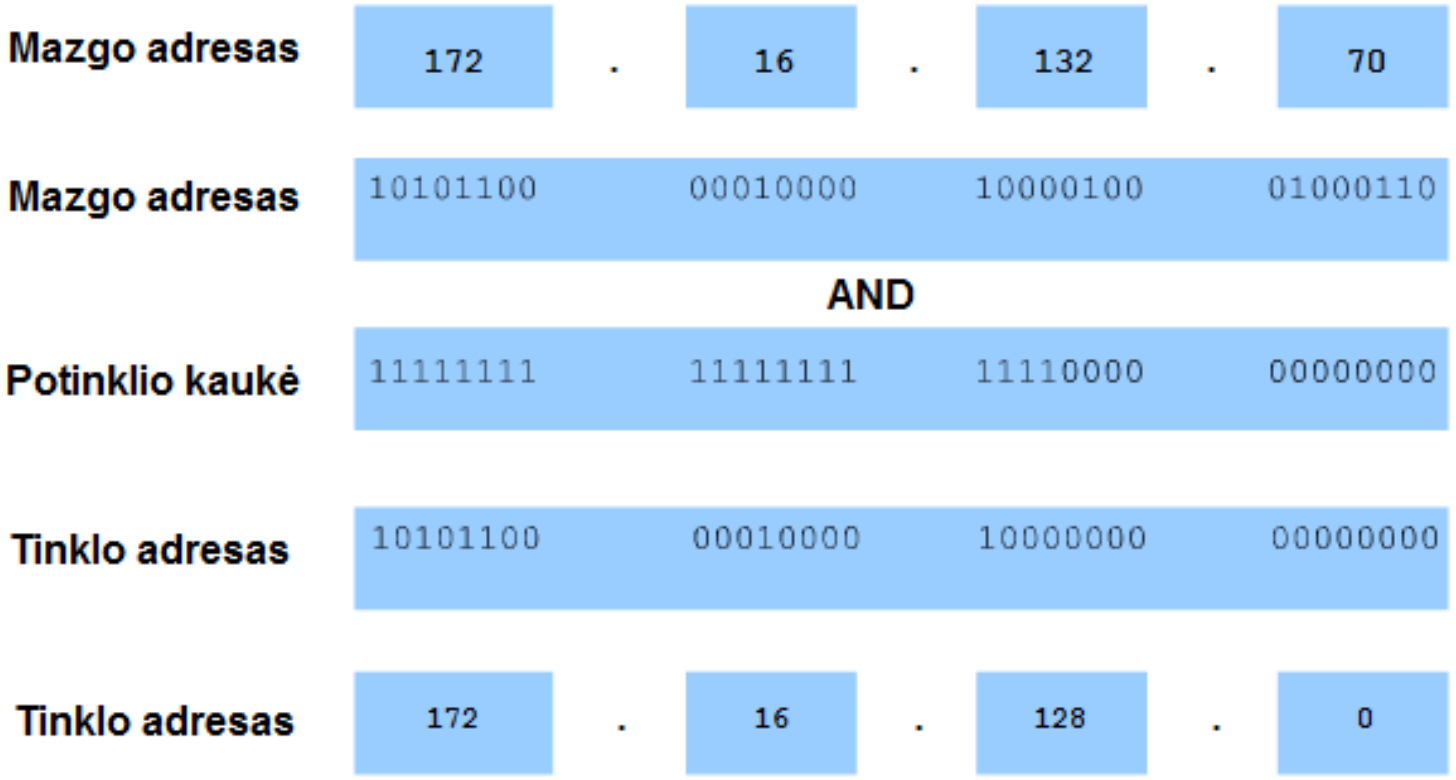
  10000001. 01000000. 10000110. 00000101
- Mask **255.255.128.0**

  11111111. 11111111. 10000000. 00000000

Network part: 10000001.01000000.1 -> 129.64.128.0

Host part: 0000110.00000101

Addressing


Mazgo adresas
172 . 16 . 132 . 70
Mazgo adresas
10101100 00010000 10000100 01000110
AND
Potinklio kaukė
11111111 11111111 11110000 00000000
Tinklo adresas
10101100 00010000 10000000 00000000
Tinklo adresas
172 . 16 . 128 . 0

Subnet mask

***Default subnet masks*:**

- A class 255.0.0.0
- B class 255.255.0.0
- C class 255.255.255.0

CIDR (classles inter-domain routing) format:

- 184.22.41.201/16 -> 16 bits are 1
- 193.219.146.24/24 -> 24 bits are 1

Examples

Network address

172 . 16. 20. 0 /25

10101100.00010000.00010100.0000000

|----------Network ----------|- host -|

0+0+0+0+0+0+0+0=0

Network address = 172.16.20.0

First host address

172 . 16. 20. 1

10101100.00010000.00010100.0000001

|----------Network ----------|- host -|

0+0+0+0+0+0+0+1=1

Lowest host address = 172.16.20.1

Broadcast address

172 . 16. 20. 127

10101100.00010000.00010100.0111111

|----------Network ----------|- host -|

0+64+32+16+8+4+2+1=127

Broadcast address = 172.16.20.127

Last host address

172 . 16. 20. 126

10101100.00010000.00010100.0111110

|----------Network ----------|- host -|

0+64+32+16+8+4+2+0=126

Highest host address = 172.16.20.126

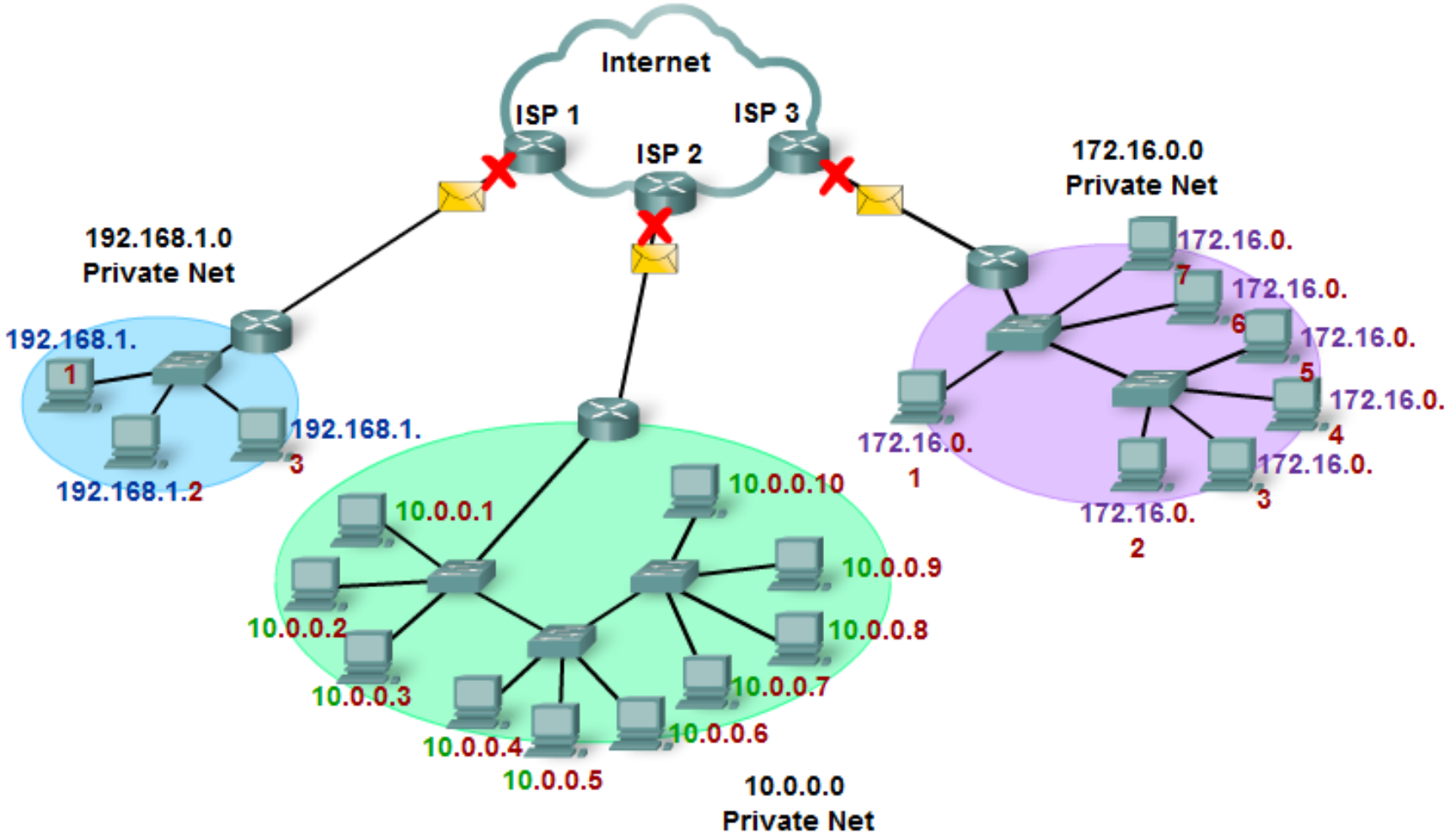
Private IP addresses

The Internet Engineering Task Force (IETF) has directed the Internet Assigned Numbers Authority (IANA) to reserve the following IPv4 address ranges for private networks, as published in RFC 1918.

Private addresses

10.0.0.0 - 10.255.255.255 (10.0.0.0 /8)	A class
172.16.0.0 - 172.31.255.255 (172.16.0.0 /12)	B class
192.168.0.0 - 192.168.255.255 (192.168.0.0 /16)	C class

Private Networks



Reserved IP addresses

In the Internet addressing architecture, the Internet Engineering Task Force (IETF) and the Internet Assigned Numbers Authority (IANA) have reserved various Internet Protocol (IP) addresses for special purposes.

These IP addresses may be used for maintenance of routing tables, multicast, operation under failure modes, or to provide addressing space for public, unrestricted uses.

More details:
http://en.wikipedia.org/wiki/Reserved_IP_addresses

IPv6

Internet Engineering Task Force (IETF) aproved IPv6.

IPv6 used 128 bits for addressing. Main advantages of IPv6

Larger Address Space

- In contrast to IPv4, IPv6 uses 4 times more bits to address a device on the Internet. This much of extra bits can provide approximately 3.4×1038 different combinations of addresses. This address can accumulate the aggressive requirement of address allotment for almost everything in this world. According to an estimate, 1564 addresses can be allocated to every square meter of this earth.

Simplified Header

- IPv6's header has been simplified by moving all unnecessary information and options (which are present in IPv4 header) to the end of the IPv6 header. IPv6 header is only twice as bigger than IPv4 provided the fact that IPv6 address is four times longer.

End-to-end Connectivity

- Every system now has unique IP address and can traverse through the Internet without using NAT or other translating components. After IPv6 is fully implemented, every host can directly reach other hosts on the Internet, with some limitations involved like Firewall, organization policies, etc.

Auto-configuration

- IPv6 supports both stateful and stateless auto configuration mode of its host devices. This way, absence of a DHCP server does not put a halt on inter segment communication.

IPv6

Faster Forwarding/Routing

- Simplified header puts all unnecessary information at the end of the header. The information contained in the first part of the header is adequate for a Router to take routing decisions, thus making routing decision as quickly as looking at the mandatory header.

Anycast Support

- This is another characteristic of IPv6. IPv6 has introduced Anycast mode of packet routing. In this mode, multiple interfaces over the Internet are assigned same Anycast IP address. Routers, while routing, send the packet to the nearest destination.

Mobility

- IPv6 was designed keeping mobility in mind. This feature enables hosts (such as mobile phone) to roam around in different geographical area and remain connected with the same IP address. The mobility feature of IPv6 takes advantage of auto IP configuration and Extension headers.

Enhanced Priority Support

- IPv4 used 6 bits DSCP (Differential Service Code Point) and 2 bits ECN (Explicit Congestion Notification) to provide Quality of Service but it could only be used if the end-to-end devices support it, that is, the source and destination device and underlying network must support it.
- In IPv6, Traffic class and Flow label are used to tell the underlying routers how to efficiently process the packet and route it.

IPv6 address

IPv6 address consists of 8 groups of hexadecimal numbers separated by colon.

Example: 2001:08e0:7d83:7d88:4f84:4c74:1d83:22b4.
Prefix in CIDR format: /64.

Example: 2001:000:7d83:7d88:4f84:4c74:1d83:22b4

2001::7d83:7d88:4f84:4c74:1d83:22b4

IPv6 adresas

IPv6 address has network part and host part.

64 bits are used of network part and 64 bits for host identification.

IPv6 address structure			
Field	Routing prefix	Subnet ID	Host ID
Bits	48 (ar daugiau)	16 (ar mažiau)	64

FQDN

Fully qualified domain name (**FQDN**) is a domain name that specifies its exact location in the tree hierarchy of the Domain Name System (DNS). It specifies all domain levels, including the top-level domain and the root zone

FQDN structure:

<kompiuterio vardas>.<domenų srities vardas (vardai)>

<kompiuterio vardas> :

- Host name: *reda, goda*, *niujorkas*
- Internet service name: *www, mail, news*

Example: *reda.vgtu.lt, mail.takas.lt, www.vgtu.lt*

<domain name>

vgtu.lt, ktu.lt, omnitel.lt bite.lt

Domain names (top level)

.com – commercial

.edu - education

.net – organizations (Internet)

.gov – USA state enterprises

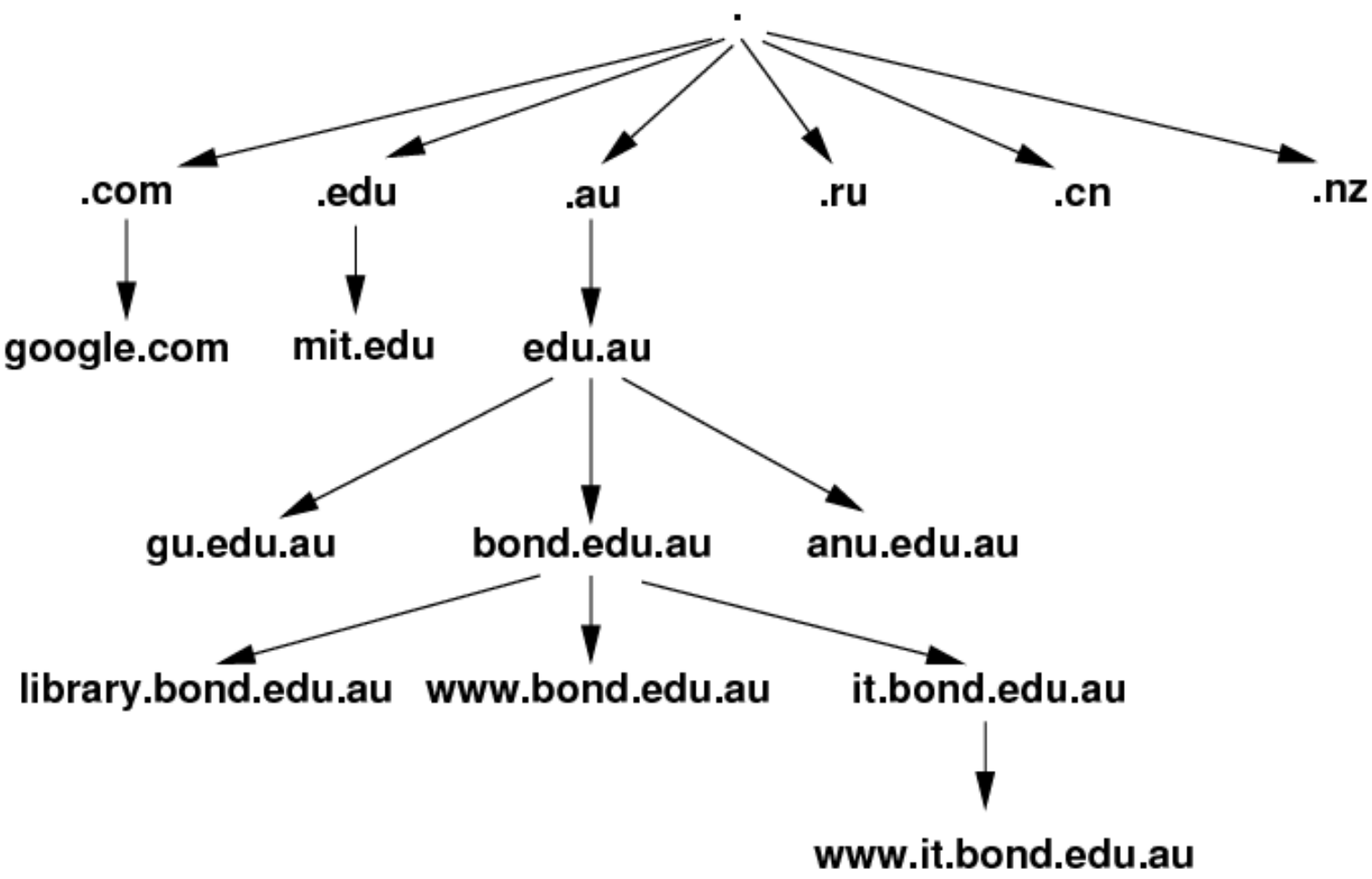
.mil – USA military

.org – non profit organizations

.CC – country code,

.lt – Lietuva, .fr – France, .dk – Denmark …

DNS tree


.
.com
.edu
.au
.ru
.cn
.nz
google.com
mit.edu
edu.au
gu.edu.au
bond.edu.au
anu.edu.au
library.bond.edu.au
www.bond.edu.au
it.bond.edu.au
www.it.bond.edu.au

DNS tool

nslookup is a network administration command-line tool available for many computer operating systems for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or for any other specific DNS record.

nslookup regimes:

- Interactive
- Non-interactive

nslookup syntax:

nslookup [-option] [hostname] [server]

Transport layer

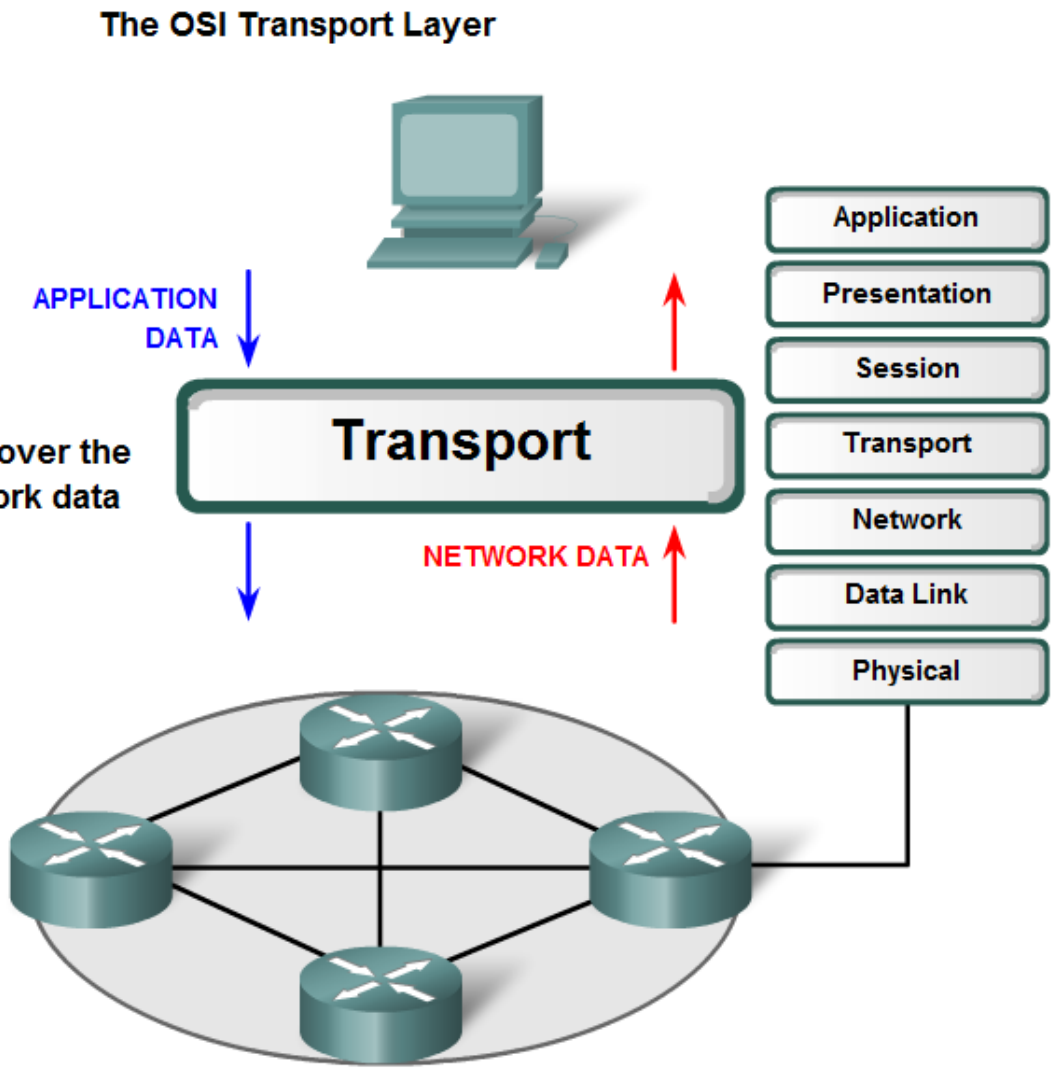
The Transport layer provides for the segmentation of data and the control necessary to reassemble these pieces into the various communication streams.

Its primary responsibilities to accomplish this are:

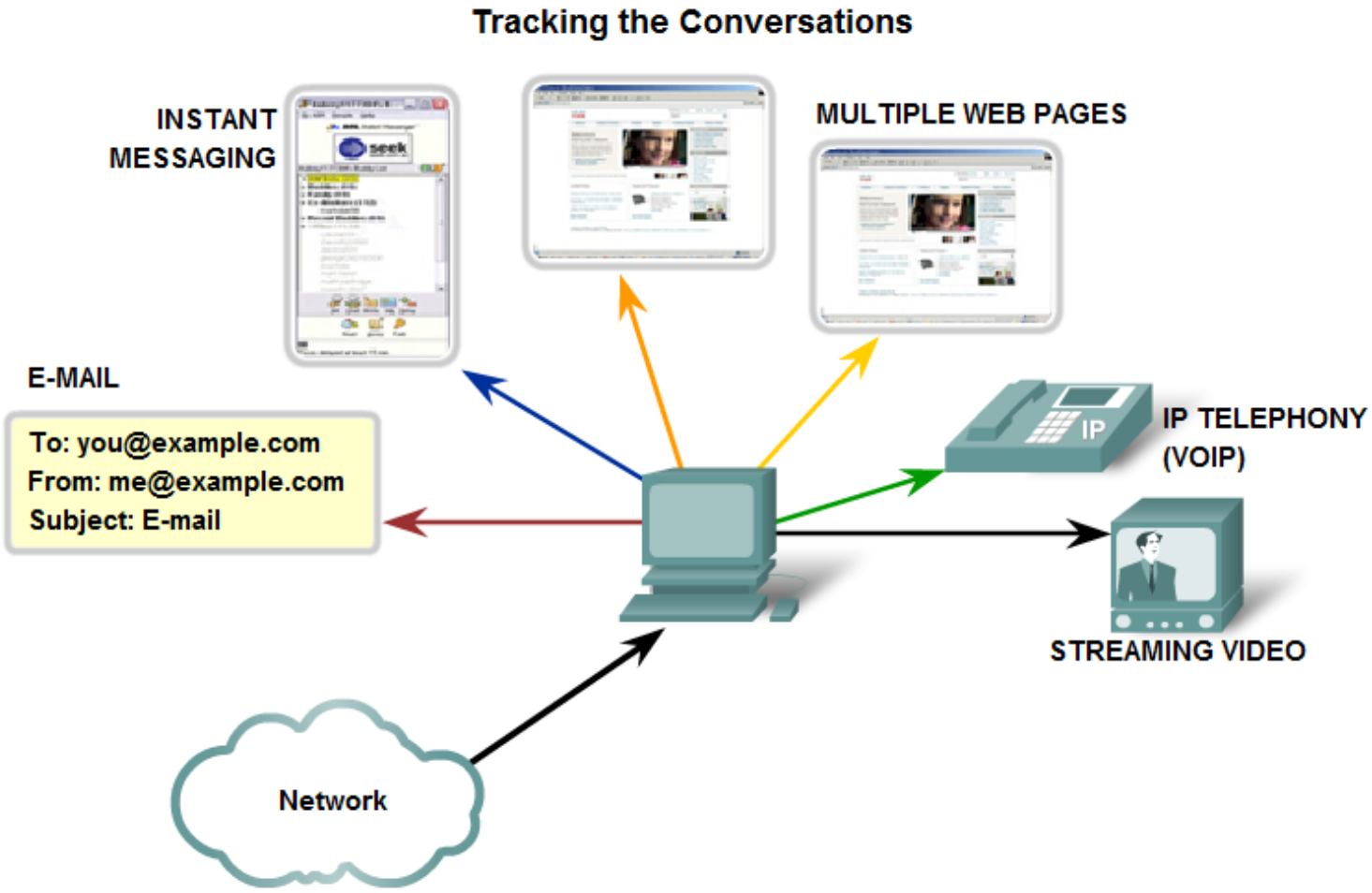
- Tracking the individual communication between applications on the source and destination hosts
- Segmenting data and managing each piece
- Reassembling the segments into streams of application data
- Identifying the different applications

Transport layer

The Transport layer prepares application data for transport over the network and processes network data for use by applications.

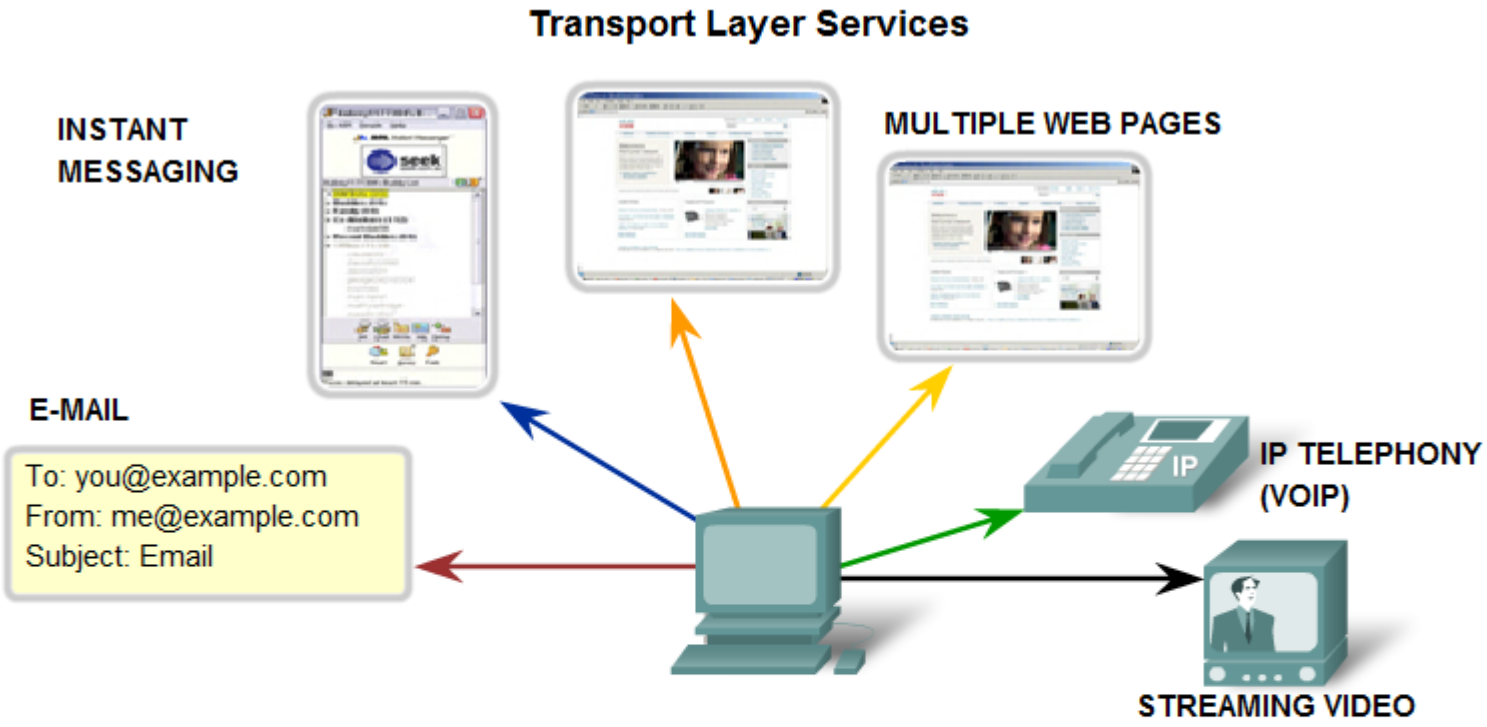


Transport layer



The Transport layer segments the data and manages the separation of data for different applications. Multiple applications running on a device receive the correct data.

Transport layer



Establishing a Session ensures the application is ready to receive the data.

Reliable delivery means lost segments are resent so the data is received complete.

Same order delivery ensures data is delivered sequentially as it was sent.

Flow Control manages data delivery if there is congestion on the host.

Transport layer

Some applications transmit large amounts of data - in some cases, many gigabytes. It would be impractical to send all of this data in one large piece. No other network traffic could be transmitted while this data was being sent. A large piece of data could take minutes or even hours to send. In addition, if there were any error, the entire data file would have to be lost or resent. Network devices would not have memory buffers large enough to store this much data while it is transmitted or received. The limit varies depending on the networking technology and specific physical medium being in use.

Dividing application data into pieces both ensures that data is transmitted within the limits of the media and that data from different applications can be multiplexed on to the media.

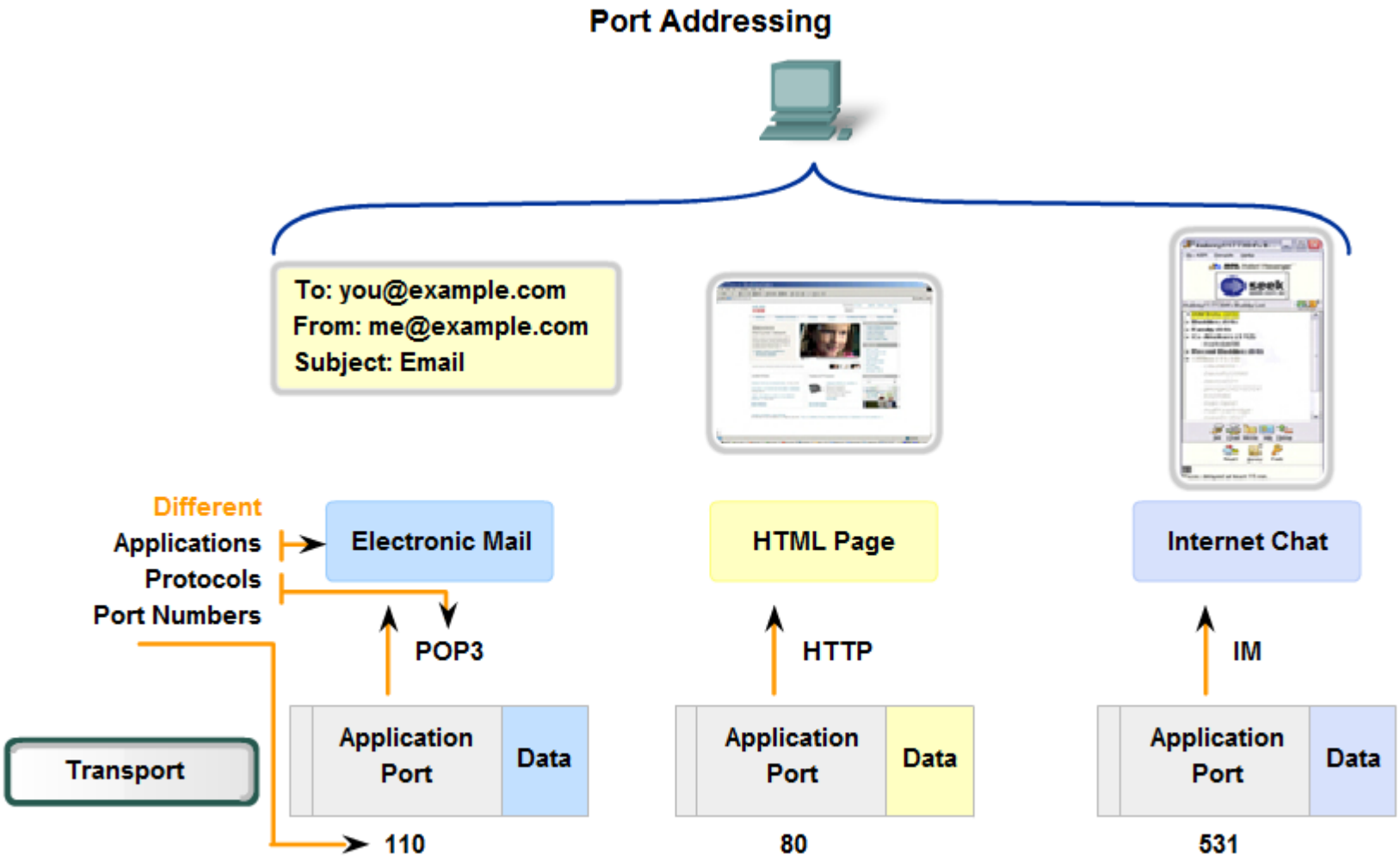
Transport layer

A Transport layer protocol can implement is a method to ensure reliable delivery of the data. In networking terms, reliability means ensuring that each piece of data that the source sends arrives at the destination.

At the Transport layer the three basic operations of reliability are:

- tracking transmitted data
- acknowledging received data
- retransmitting any unacknowledged data

Ports



Data for different applications is directed to the correct application because each application has a unique port number.

Ports

Sometimes it is necessary to know which active TCP connections are open and running on a networked host. **netstat** is an network utility that can be used to verify those connections. It lists the protocol in use, the local address and port number, the foreign address and port number, and the state of the connection. Unexplained TCP connections can pose a major security threat. This is because they can indicate that something or someone is connected to the local host.

```
C:\>netstat

Active Connections

Proto    Local Address          Foreign Address        State
TCP      kenpc:3126             192.168.0.2:netbios-ssn    ESTABLISHED
TCP      kenpc:3158             207.138.126.152:http       ESTABLISHED
TCP      kenpc:3159             207.138.126.169:http       ESTABLISHED
TCP      kenpc:3160             207.138.126.169:http       ESTABLISHED
TCP      kenpc:3161             sc.msn.com:http            ESTABLISHED
TCP      kenpc:3166             www.cisco.com:http         ESTABLISHED

C:\>
```

TCP protocol

TCP is a connection-oriented protocol, described in RFC 793. TCP incurs additional overhead to gain functions.

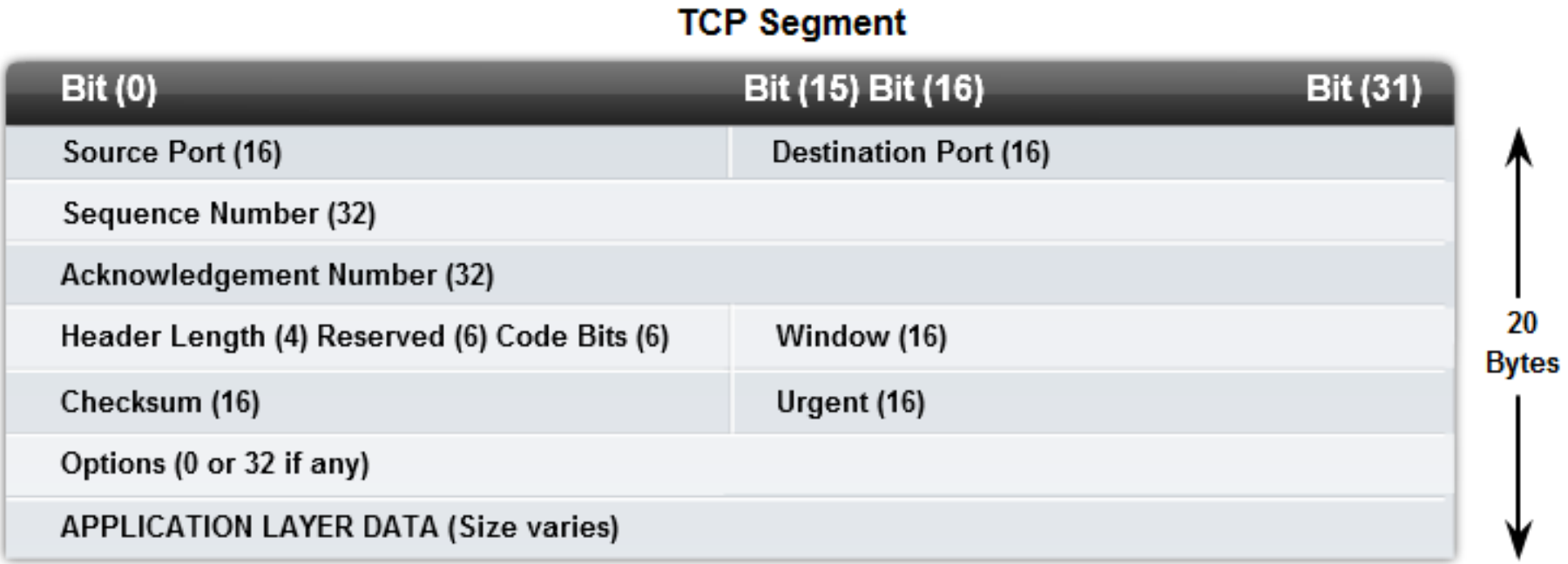
Additional functions specified by TCP are the same order delivery, reliable delivery, and flow control.

Each TCP segment has 20 bytes of overhead in the header encapsulating the Application layer data, whereas each UDP segment only has 8 bytes of overhead.

Applications that use TCP are:

- Web Browsers
- E-mail

TCP header


TCP Segment
Bit (0)
Bit (15) Bit (16)
Bit (31)
Source Port (16)
Destination Port (16)
Sequence Number (32)
Acknowledgement Number (32)
Header Length (4) Reserved (6) Code Bits (6)
Window (16)
Checksum (16)
Urgent (16)
Options (0 or 32 if any)
APPLICATION LAYER DATA (Size varies)
20 Bytes

TCP

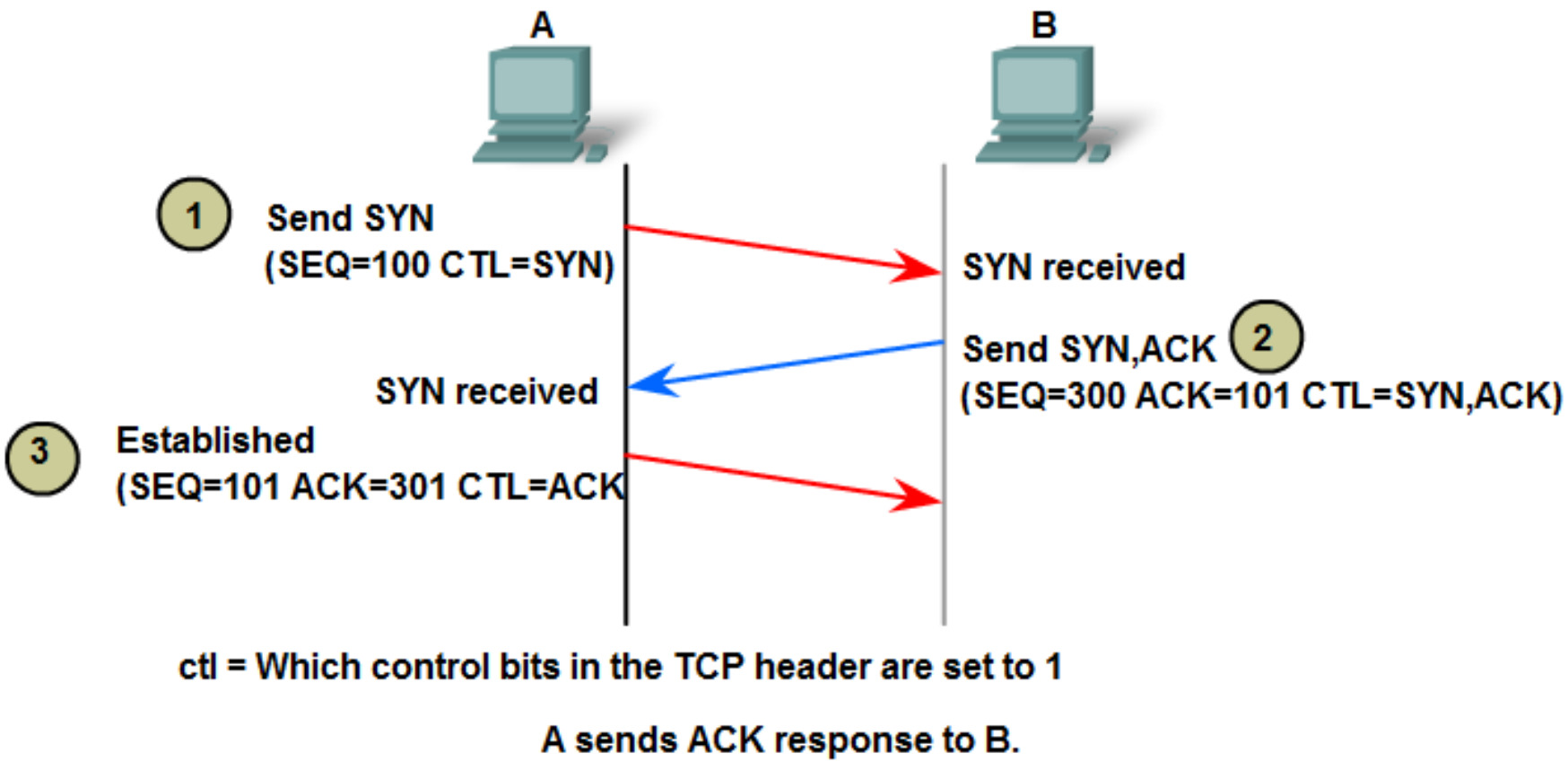
The host tracks each data segment within a session and exchanges information about what data is received by each host using the information in the TCP header.

Each connection represents two one-way communication streams, or sessions.

To establish the connection, the hosts perform a **three-way handshake**. Control bits in the TCP header indicate the progress and status of the connection. The three-way handshake:

- Establishes that the destination device is present on the network
- Verifies that the destination device has an active service and is accepting requests on the destination port number that the initiating client intends to use for the session
- Informs the destination device that the source client intends to establish a communication session on that port number

Three-way handshake



ACK - Acknowledgement

SYN - Synchronize sequence numbers

Acknowledgement

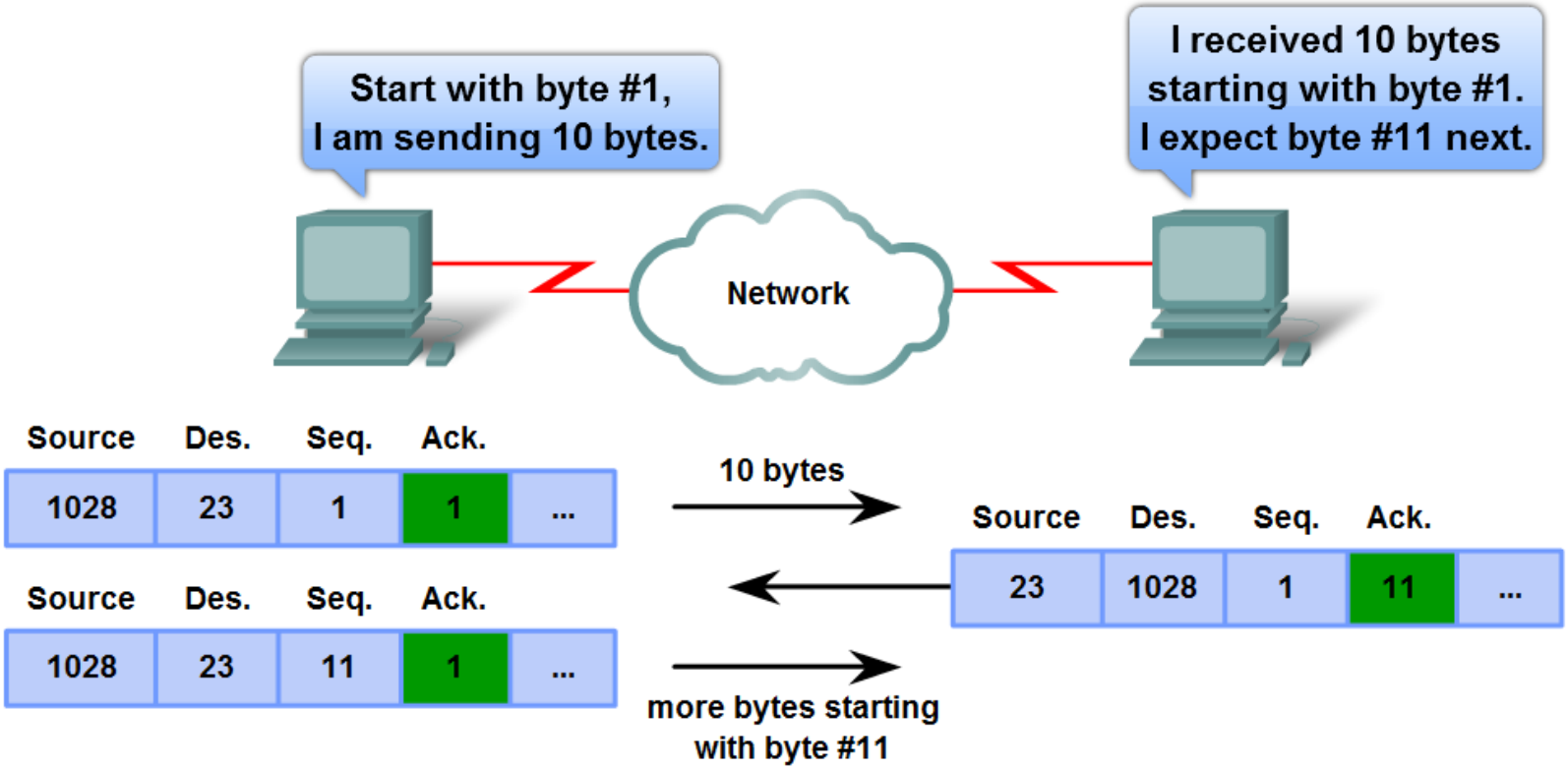
One of TCP's functions is making sure that each segment reaches its destination. The TCP services on the destination host acknowledge the data that it has received to the source application.

The segment header sequence number and acknowledgement number are used together to confirm receipt of the bytes of data contained in the segments. The sequence number indicates the relative number of bytes that have been transmitted in this session including the bytes in the current segment. TCP uses the acknowledgement number in segments sent back to the source to indicate the next byte in this session that the receiver expects to receive. This is called expectational acknowledgement.

The source is informed that the destination has received all bytes in this data stream up to, but not including, the byte indicated by the acknowledgement number. The sending host is expected to send a segment that uses a sequence number that is equal to the acknowledgement number.

Acknowledgement process

Source Port	Destination Port	Sequence Number	Acknowledgement Numbers	...



Handling Segment Loss

No matter how well designed a network is, data loss will occasionally occur. Therefore, TCP provides methods of managing these segment losses. Among these is a mechanism to retransmit segments with unacknowledged data.

A destination host service using TCP usually only acknowledges data for contiguous sequence bytes. If one or more segments are missing, only the data in the segments that complete the stream are acknowledged.

For example, if segments with sequence numbers 1500 to 3000 and 3400 to 3500 were received, the acknowledgement number would be 3001. This is because there are segments with the sequence numbers 3001 to 3399 that have not been received.

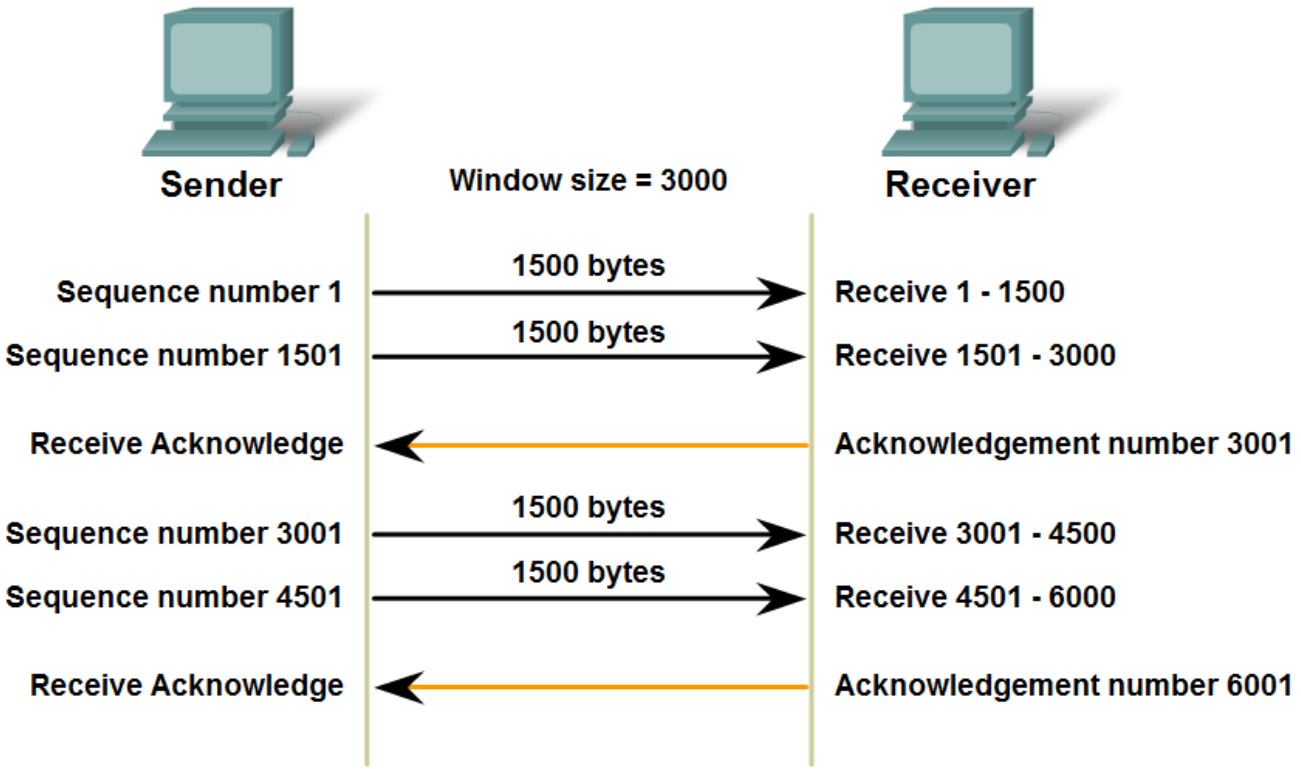
Flow control

TCP also provides mechanisms for flow control. Flow control assists the reliability of TCP transmission by adjusting the effective rate of data flow between the two services in the session. When the source is informed that the specified amount of data in the segments is received, it can continue sending more data for this session.

This **Window Size** field in the TCP header specifies the amount of data that can be transmitted before an acknowledgement must be received. The initial window size is determined during the session startup via the three-way handshake.

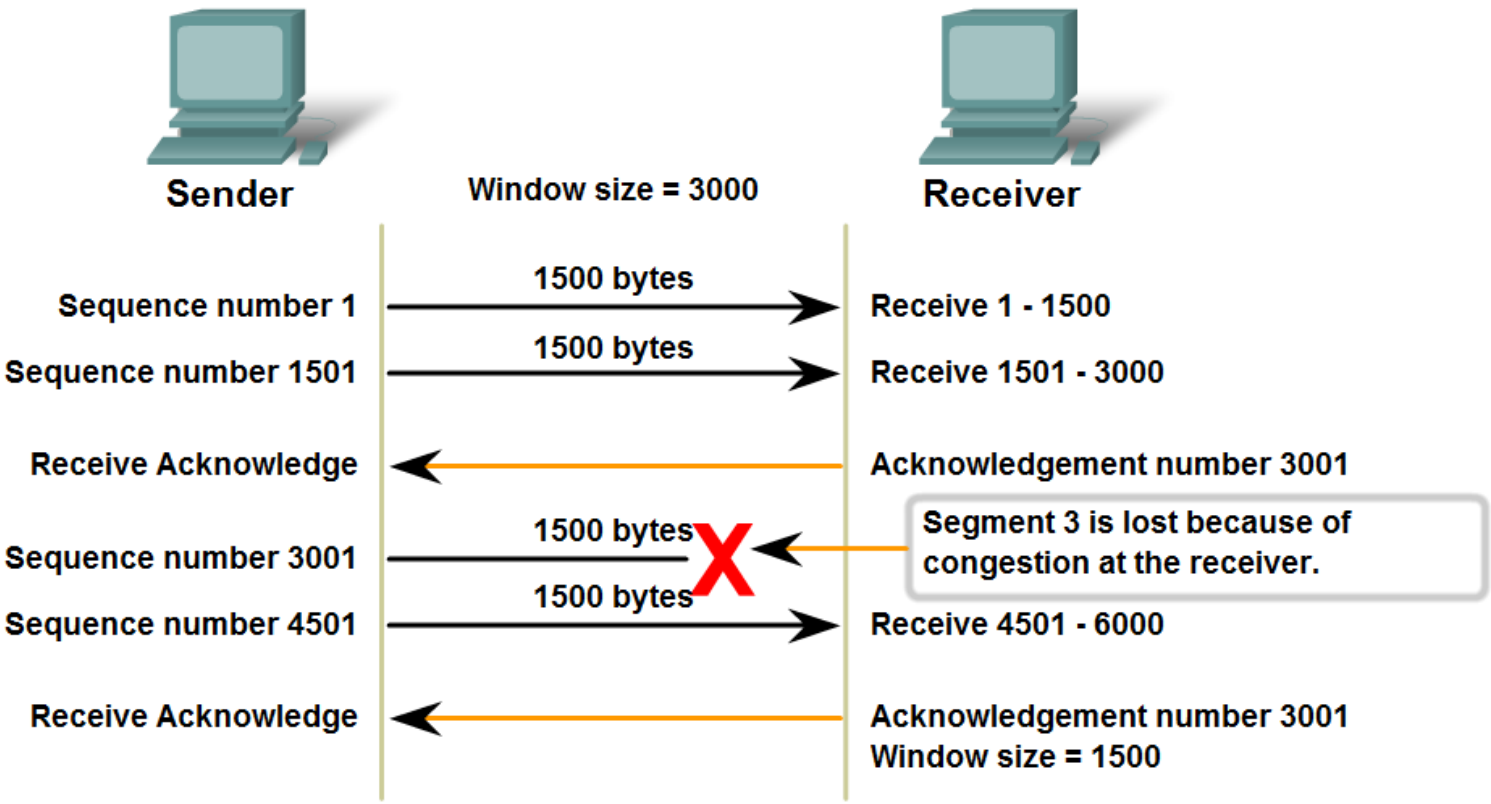
TCP feedback mechanism adjusts the effective rate of data transmission to the maximum flow that the network and destination device can support without loss. TCP attempts to manage the rate of transmission so that all data will be received and retransmissions will be minimized.

Window



The window size determines the number of bytes sent before an acknowledgment is expected.
The acknowledgement number is the number of the next expected byte.

Window



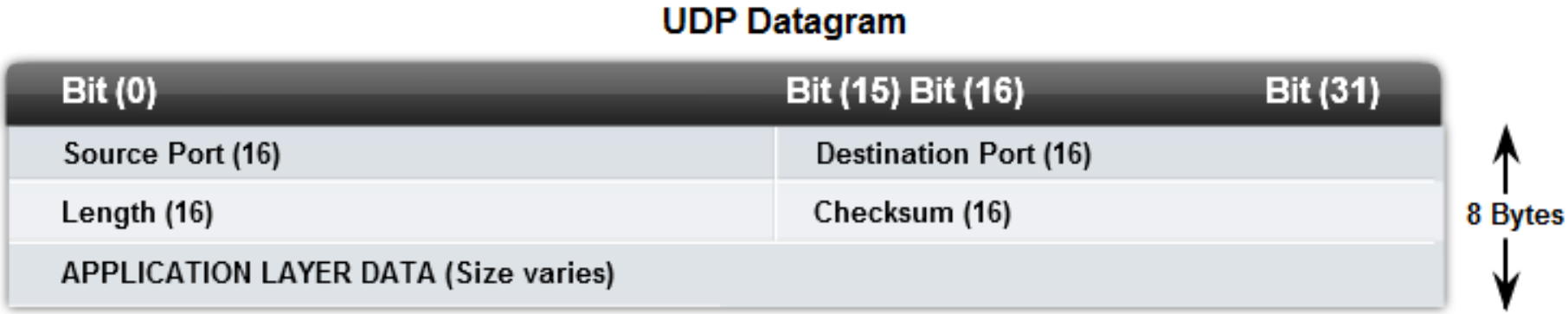
If segments are lost because of congestion, the Receiver will acknowledge the last received sequential segment and reply with a reduced window size.

UDP

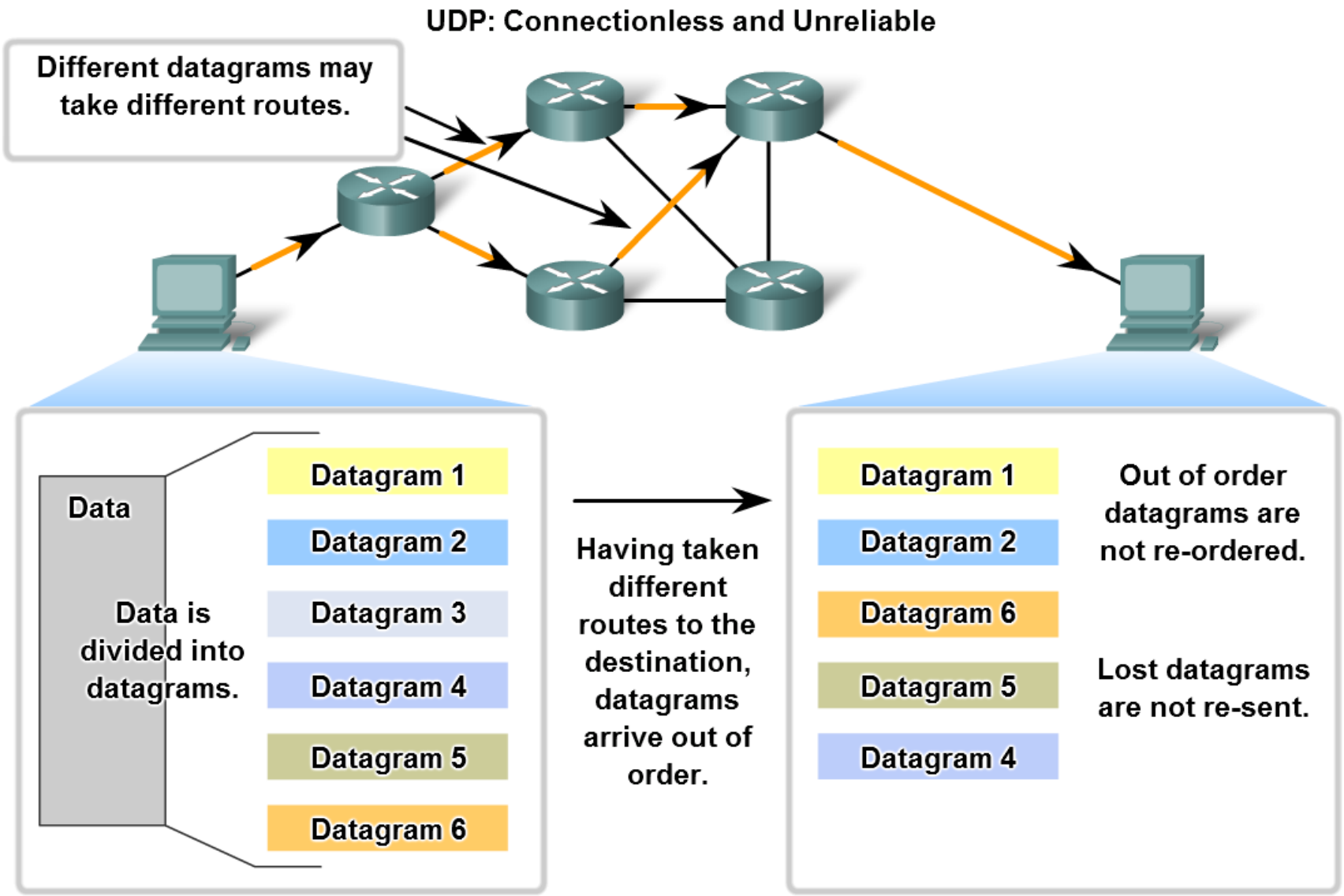
UDP is a simple, connectionless protocol, described in RFC 768. It has the advantage of providing for low overhead data delivery. The pieces of communication in UDP are called datagrams. These datagrams are sent as "best effort" by this Transport layer protocol.

Applications that use UDP include:

- Domain Name System (DNS)
- Video Streaming.



UDP


UDP: Connectionless and Unreliable
Different datagrams may take different routes.
Data
Data is divided into datagrams.
Datagram 1
Datagram 2
Datagram 3
Datagram 4
Datagram 5
Datagram 6
Having taken different routes to the destination, datagrams arrive out of order.
Datagram 1
Datagram 2
Datagram 6
Datagram 5
Datagram 4
Out of order datagrams are not re-ordered.
Lost datagrams are not re-sent.

UDP

Key Application layer protocols that use UDP include:

- Domain Name System (DNS)
- Simple Network Management Protocol (SNMP)
- Dynamic Host Configuration Protocol (DHCP)
- Routing Information Protocol (RIP)
- Trivial File Transfer Protocol (TFTP)

UDP

As with TCP, client/server communication is initiated by a client application that is requesting data from a server process. The UDP client process randomly selects a port number from the dynamic range of port numbers and uses this as the source port for the conversation. The destination port will usually be the Well Known or Registered port number assigned to the server process.

Randomized source port numbers also help with security. If there is a predictable pattern for destination port selection, an intruder can more easily simulate access to a client by attempting to connect to the port number most likely to be open.

Because there is no session to be created with UDP, as soon as the data is ready to be sent and the ports identified, UDP can form the datagram and pass it to the Network layer to be addressed and sent on the network.