

PEER TO PEER AND CLIENT SERVER ARCHITECTURE

Mohammad S. Hasan

Peer-to-peer network

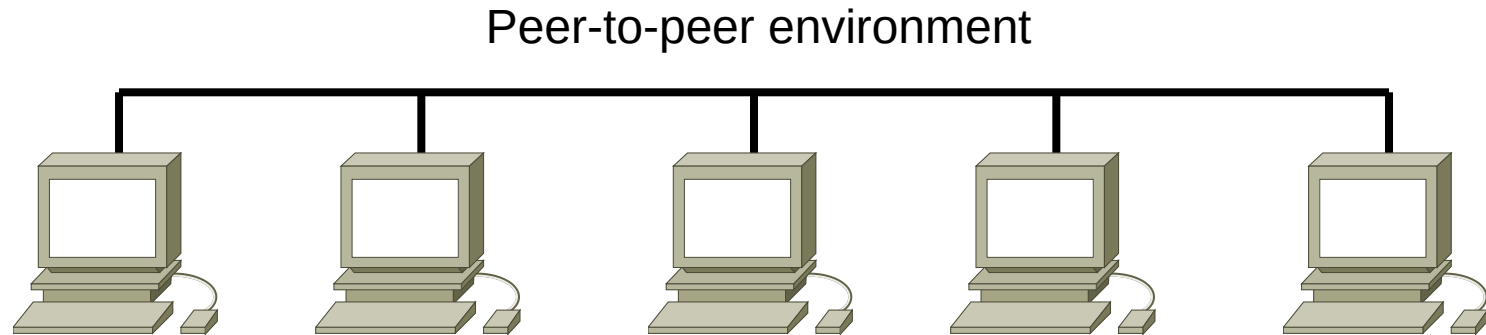
2

- ❑ No computer has control over the network
- ❑ All nodes are equal, it is based on sharing data either as directories or as files
- ❑ Simple easy structure
- ❑ Should not really exceed 10 nodes
- ❑ Normally based around a hub or switch

Peer-to-peer

3

- It appears if all nodes are connected directly to each other
 - ▣ If this were true you would need multiple interface cards

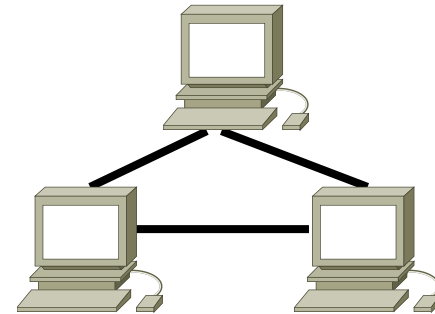


Number of connections

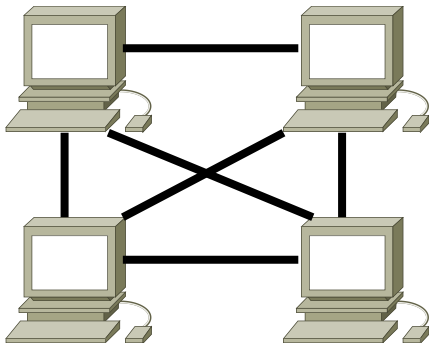
4



1 node = 1 link

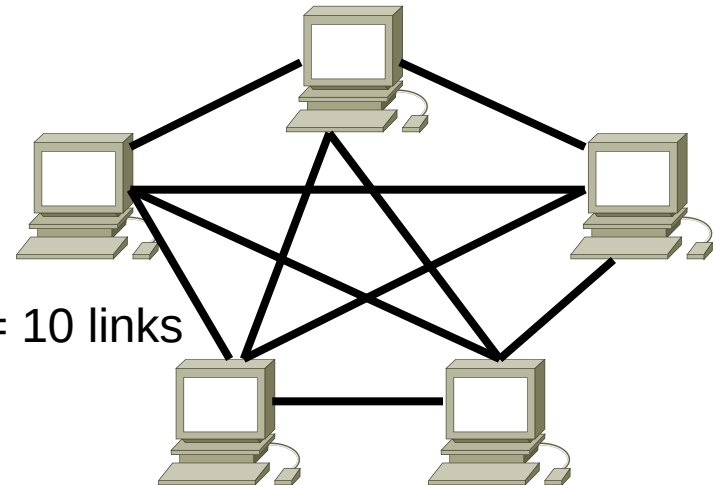


3 nodes = 3 links



4 nodes = 6 links

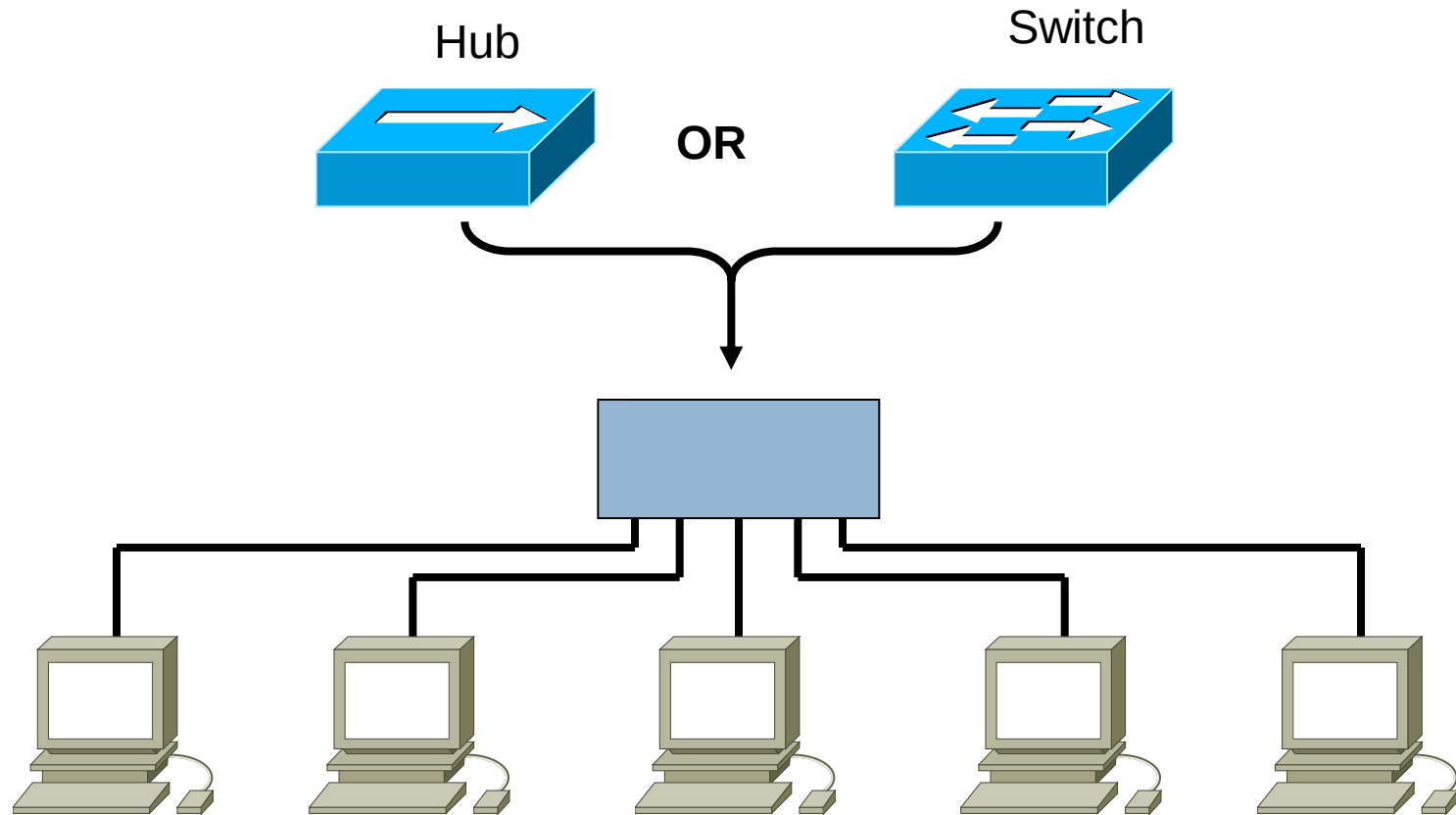
**If N = number of
nodes then
 $N(N-1)/2$ =
number of
connections**



5 nodes = 10 links

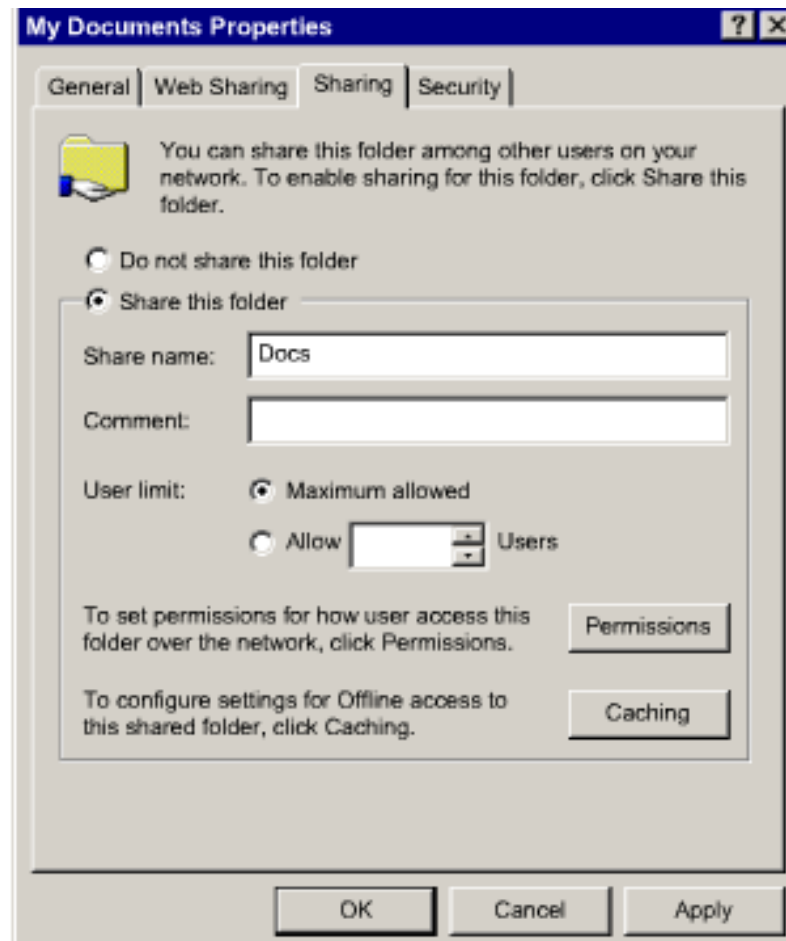
Peer-to-peer physical setup

5



Sharing data

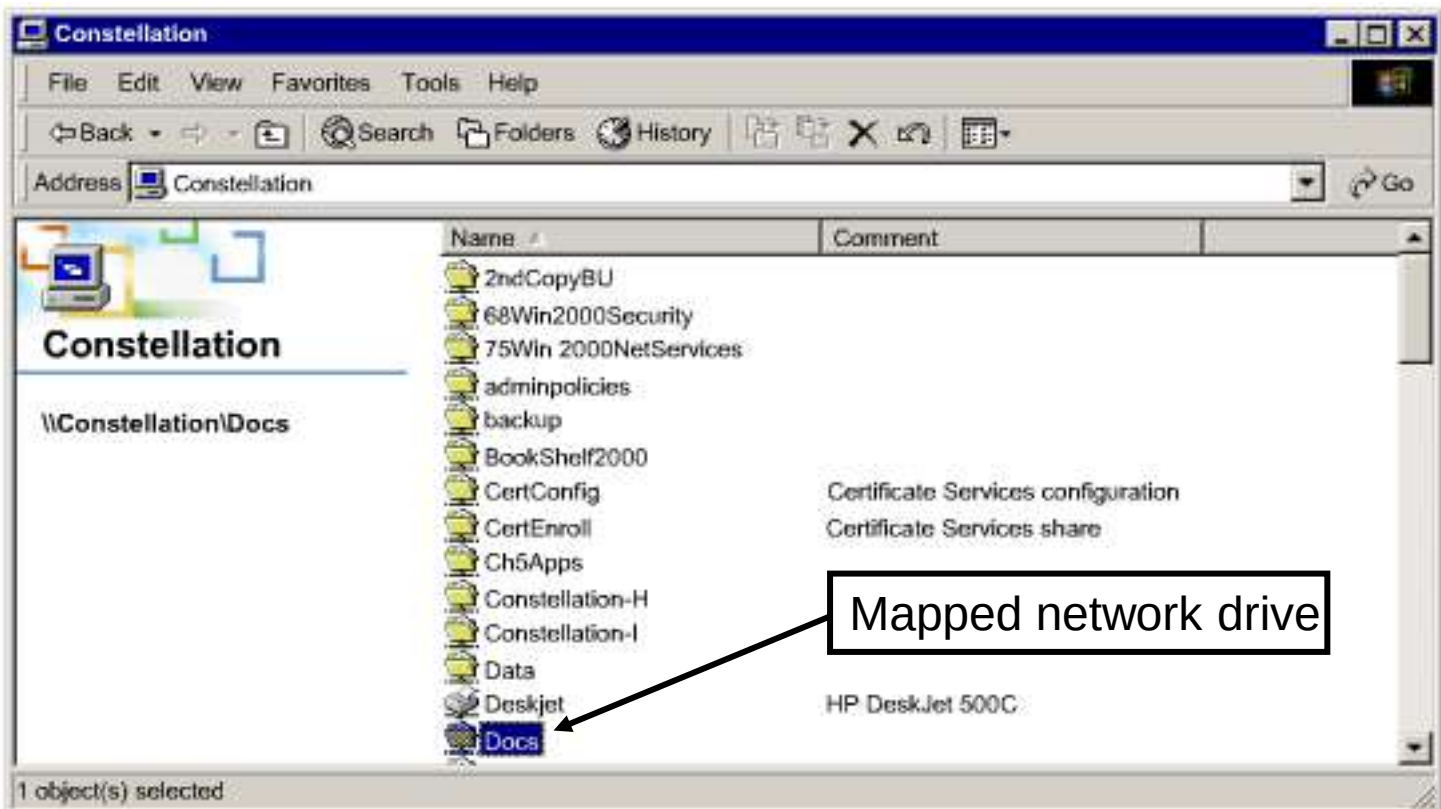
6



Peer To Peer and Client Server Architecture

Shared folders

7



HSN1

Shared folder symbol used in MS Windows XP explorer

Using a peer-to-peer network

8

- Each user must have a local account on each node
- If each user has an account on each node and have their own profile (wallpaper, screen-saver etc...)
 - ▣ then the registry file will become large and may impede performance of that machine

Client Server Paradigm

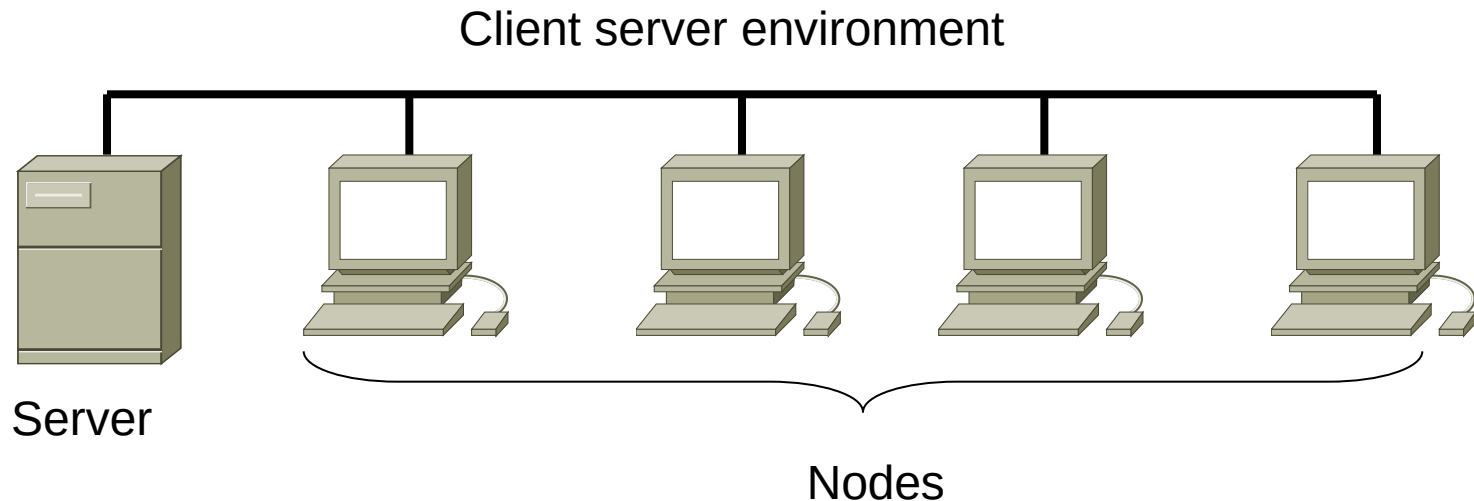
9

- Network applications use a form of communication known as the client-server paradigm.
 - ▣ A server application waits *passively* for contact
 - ▣ a client application initiates communication *actively*

Client Server Paradigm

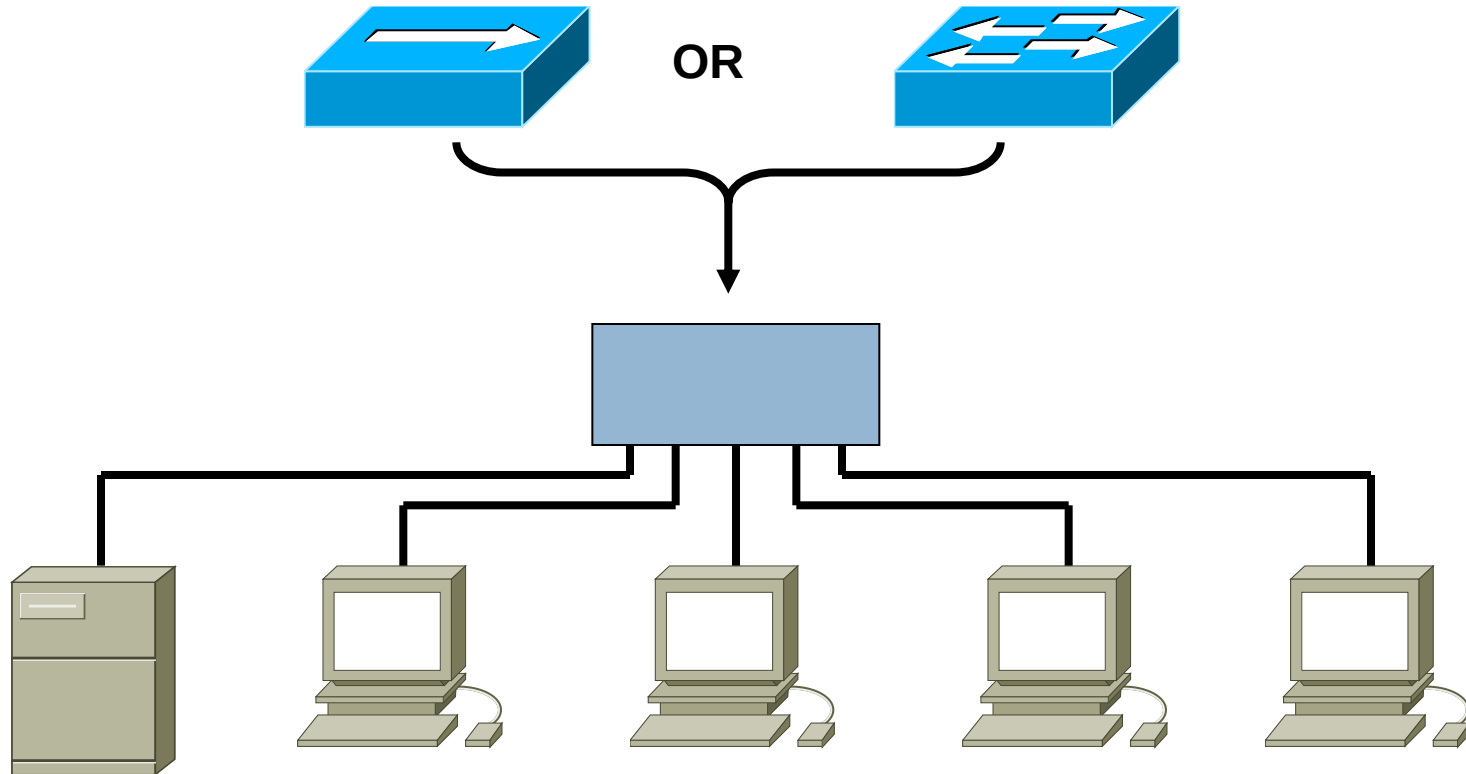
10

- It appears if all of the nodes are connected directly to each other
- If this were true you would need multiple interface cards



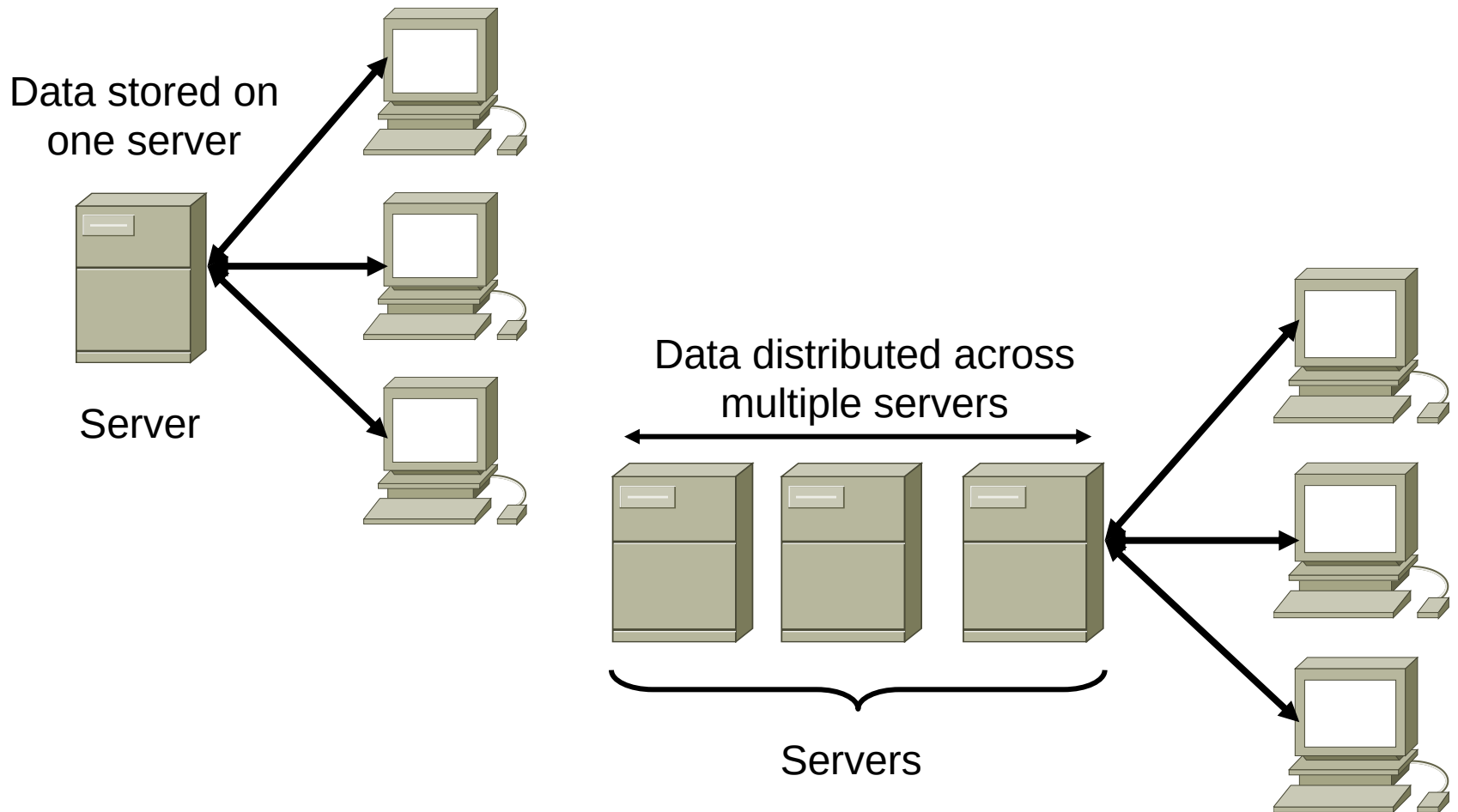
Client server physical setup

11



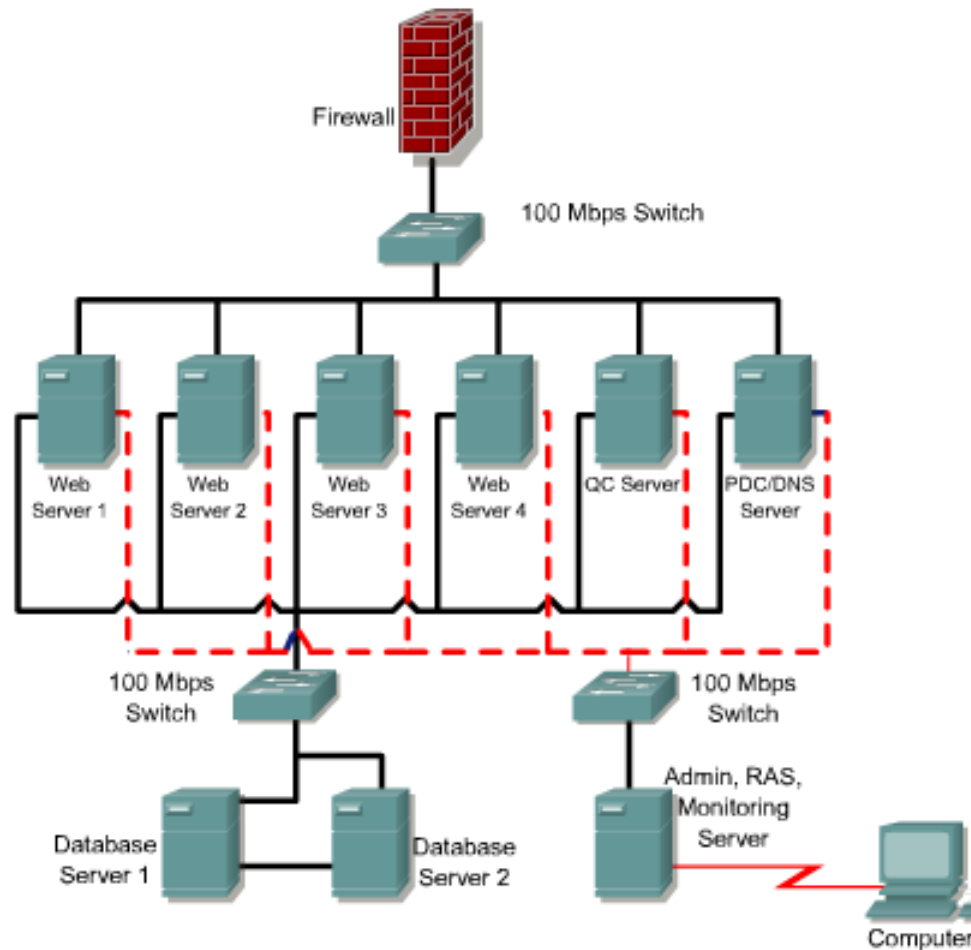
One or more servers

12



Server farm

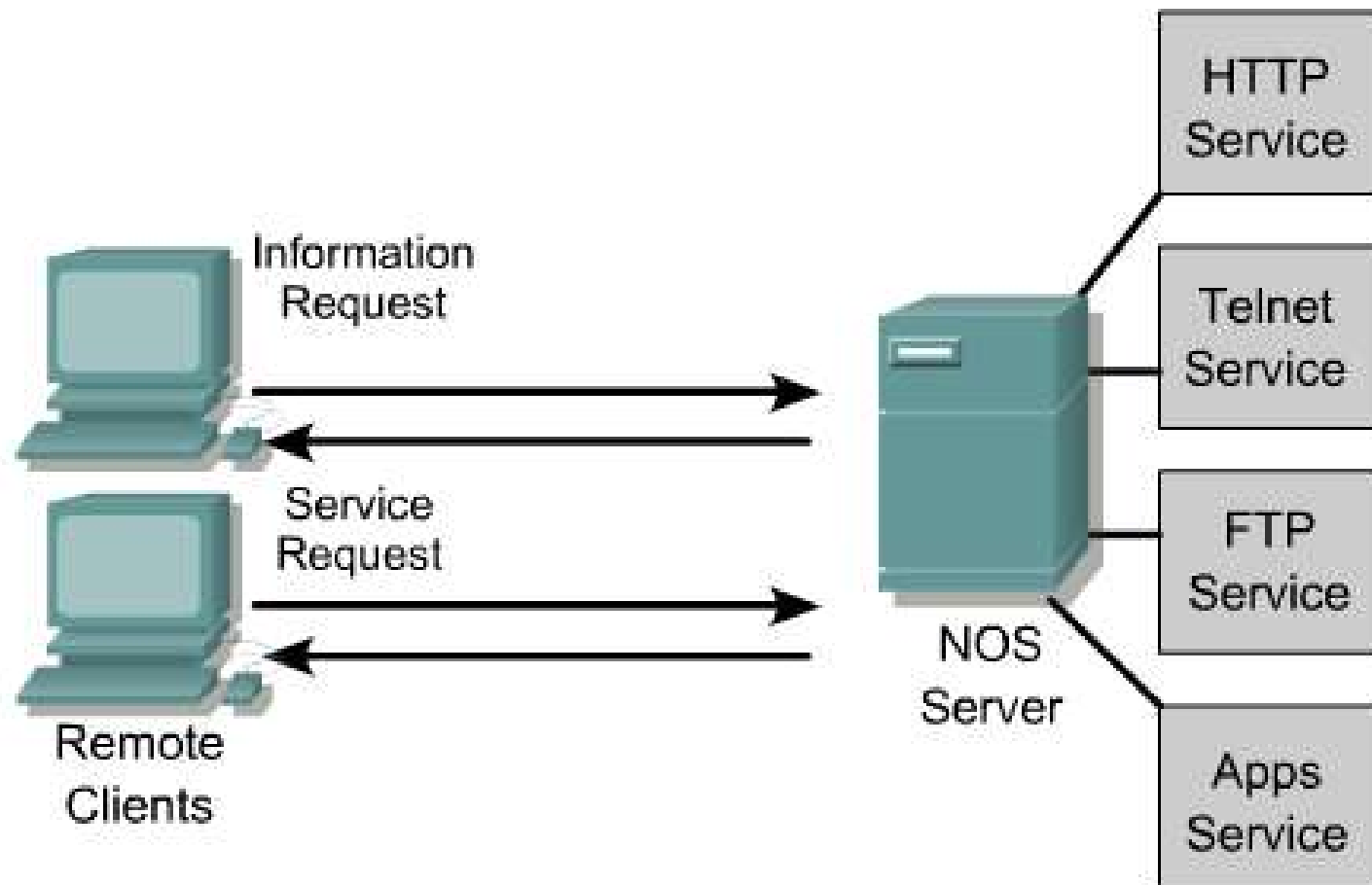
13



Peer To Peer and Client Server Architecture

Client server interaction

14



Server?

15

- A server is NOT a computer
- A server is a piece of software or a program that waits passively for communications
- A computer with a fast CPU and lots of memory that runs multiple server programs is really a *Server-Class Computer*
 - ▣ we normally refer to this hardware as “the server”

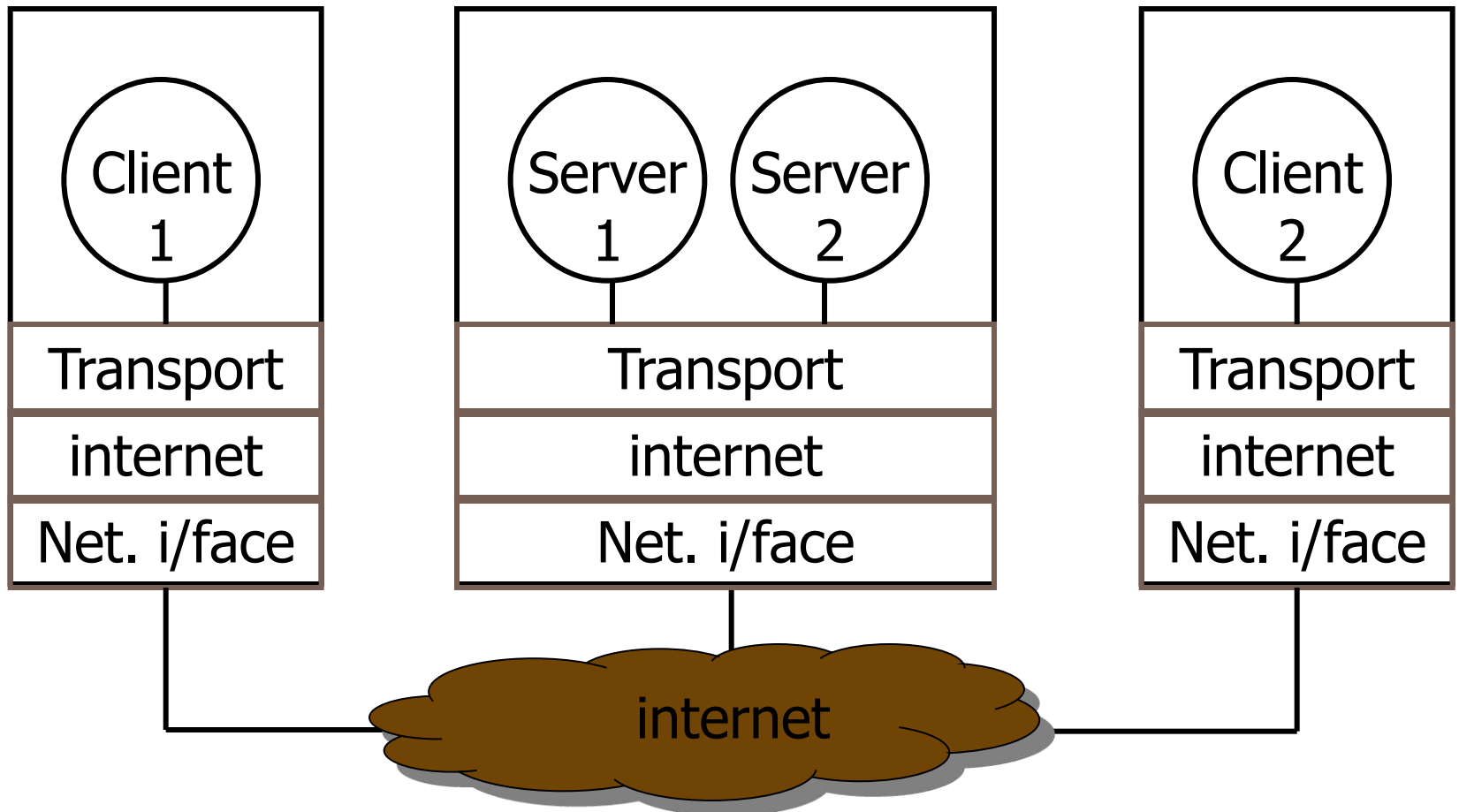
Server example

16

- <http://www.fujitsu.com/uk/products/computing/servers/primergy/scale-out/cx1000/index.html>
- <http://www.supermicro.nl/products/system/5U/>

Multiple services on one computer

17



Concurrency

18

- Support of multiple application programs
- A program that has more than one thread of control is called a concurrent program
- Concurrency is fundamental as it allows the server to service multiple clients at the same time
- If N clients using a given service on a single computer, there are $N+1$ threads.
 - ▣ N threads are interacting with the client, the main thread awaiting additional requests

Client/Server model

19

- ❑ Data stored on fast server-class computer
- ❑ Program logic & user interface is on client machine
- ❑ Normally a client initiates a conversation
- ❑ A server waits for a conversation
- ❑ Application Server
 - ▣ A server which processes & analyses the data before sending the results

Why Client/Server?

20

- Resource sharing
- High reliability (alternate sources of supply)
- Saving money
 - ▣ Small computers have much better price/performance ratio than large ones
 - ▣ Mainframes are approx 10x faster than PC's but 1000x more expensive
 - ▣ Store data on 1 or more servers

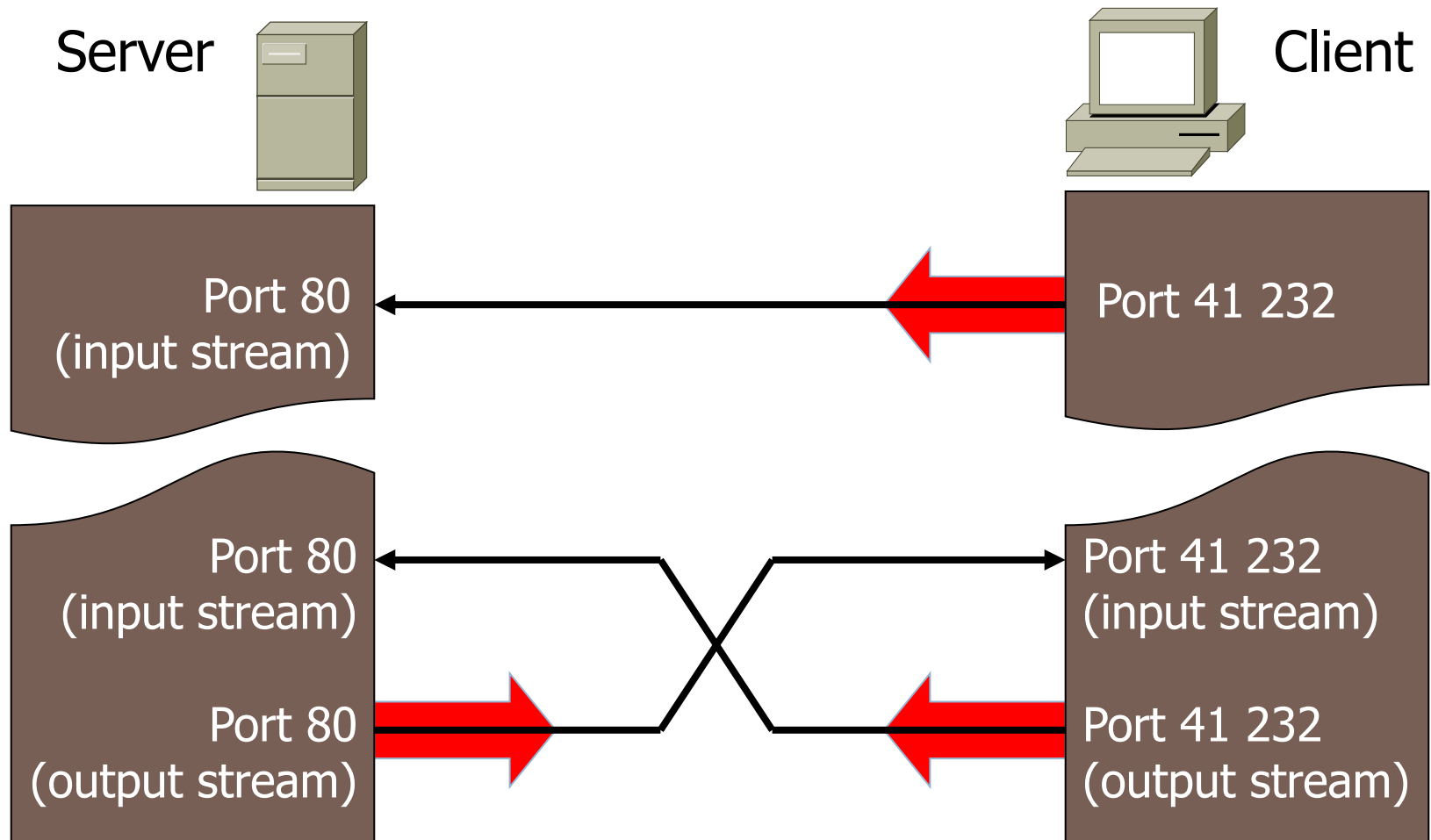
Why Client/Server?

21

- Scalability
 - ▣ As more speed required add more processors to the servers or more servers
 - ▣ Cannot be done on a mainframe
- Improved communications

Client/Server Connection

22



Peer To Peer and Client Server Architecture

Advantages of peer-to-peer & client server networks

23

Advantages of a peer-to-peer network	Advantages of a client server network
Less expensive to implement	Provides for better security
Does not require additional specialised network administration software	Easier to administer when the network is large because administration is centralised
Does not require a dedicated network administrator	All data can be backed up on one central location

Disadvantages of peer-to-peer & client server networks

24

Disadvantages of a peer-to-peer network	Disadvantages of a client server network
Does not scale well to large networks and administration becomes unmanageable	Requires expensive specialised network administrative and operational software
Each user must be trained to perform administrative tasks	Requires expensive, more powerful hardware for the server machine
Less secure	Requires a professional administrator
All machines sharing the resources negatively impact the performance	Has a single point of failure. User data is unavailable if the server is down

Thin client

25

- Based on a system where nodes talk to one or more servers
- Similar idea to client server
- HOWEVER
 - ▣ The nodes are typically dumb terminals
 - ▣ That is may have no storage, mono-chrome displays, may not have a GUI based OS

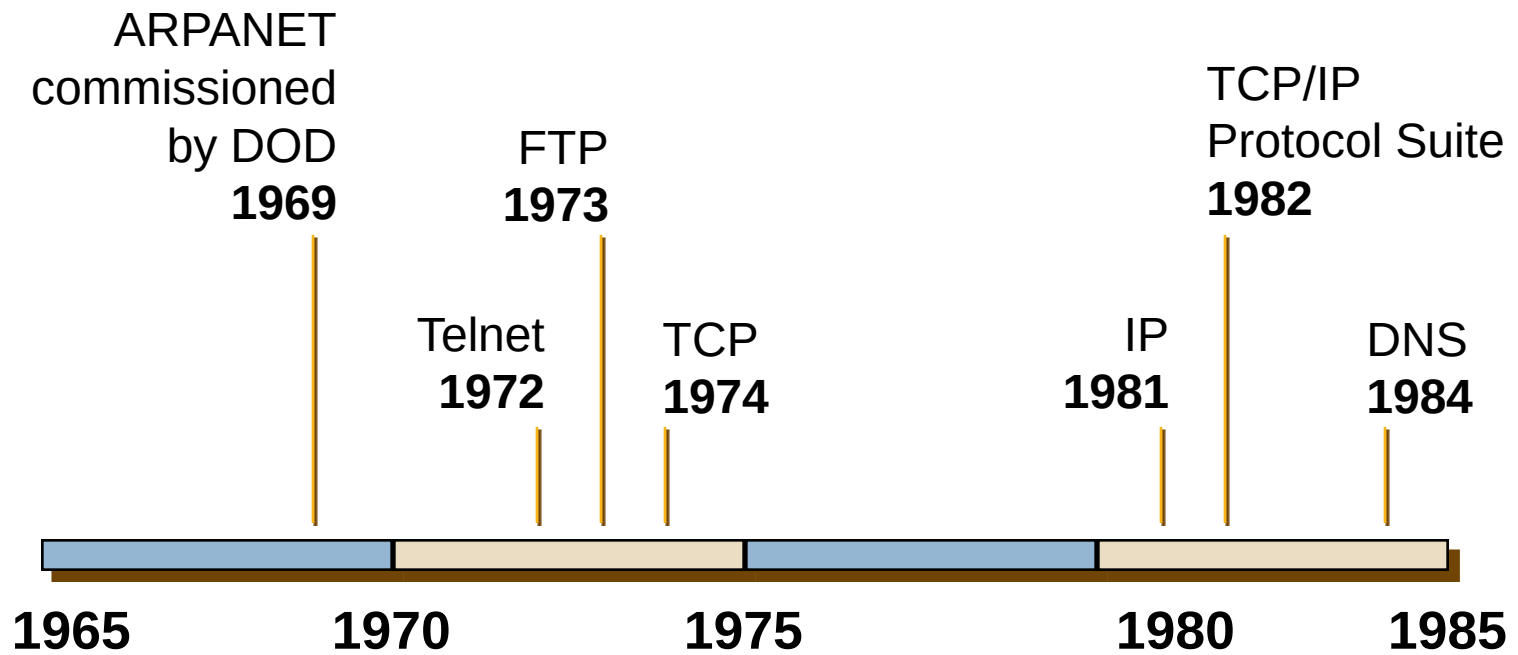
Thin client

26

- All the processing is executed by the server
- The client makes a request to the server
- The server process the result
- The result is sent back to the client and displayed
- Example
 - ▣ A terminal in a bank querying a central database about an account

TCP/IP development

27



TCP/IP Utilities

28

FTP

TFTP

RCP

Telnet

RSH

REXEC

LPR

LPQ

LPD

PING

IPCONFIG

NSLOOKUP

HOSTNAME

NETSTAT

NBTSTAT

ROUTE

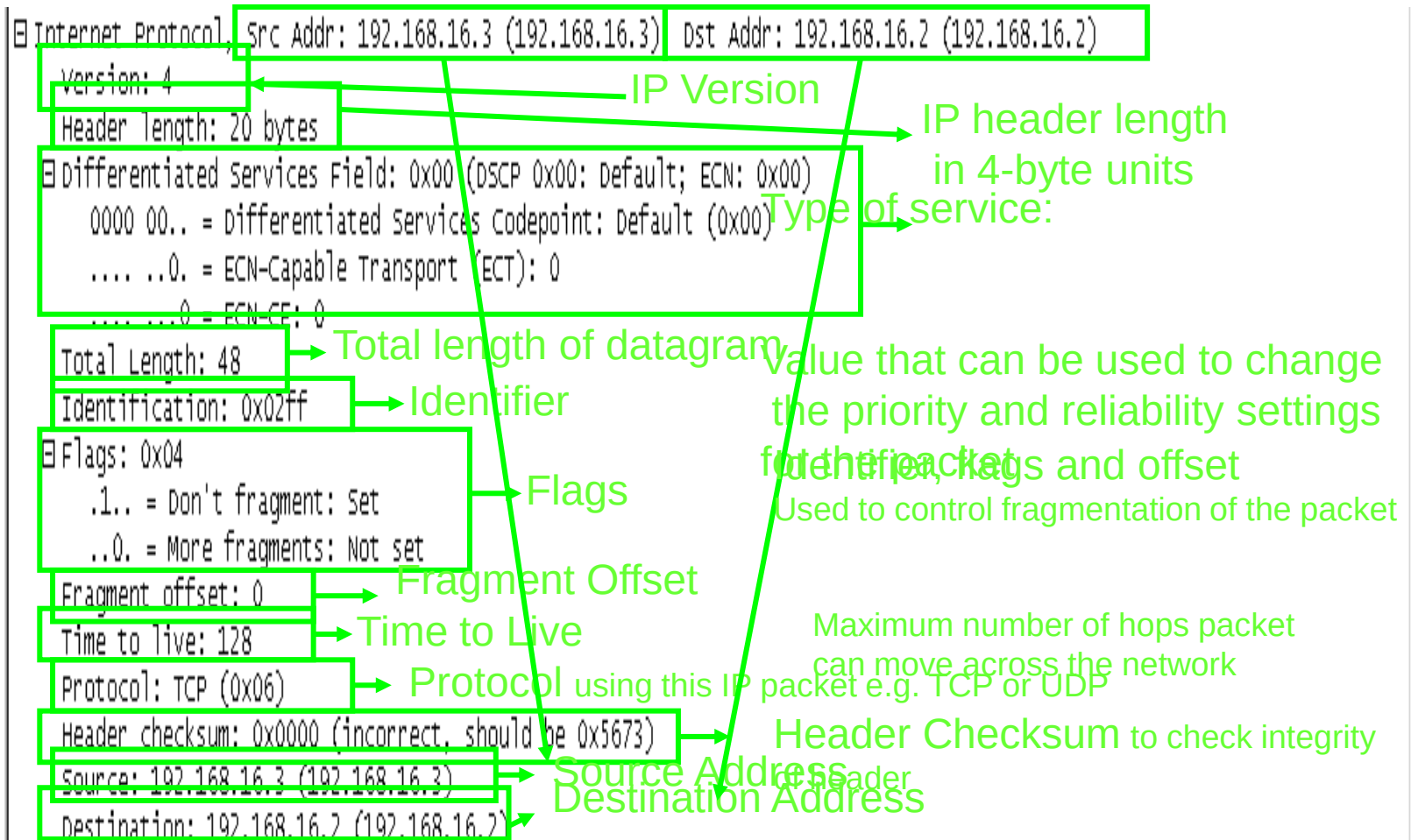
TRACERT

ARP

Finger

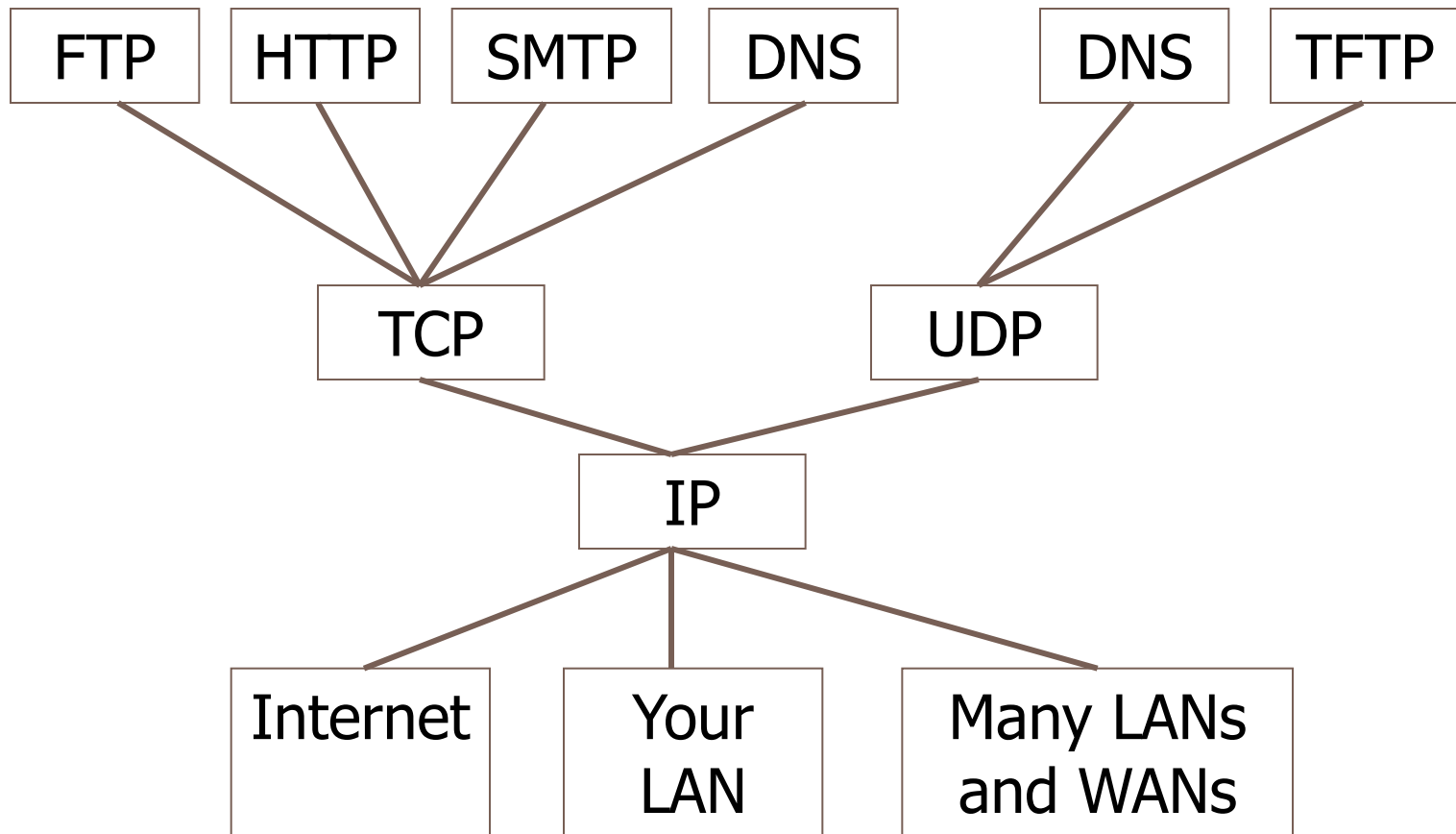
IP ver 4 (IPv4): Screen capture from Network Analyser

29



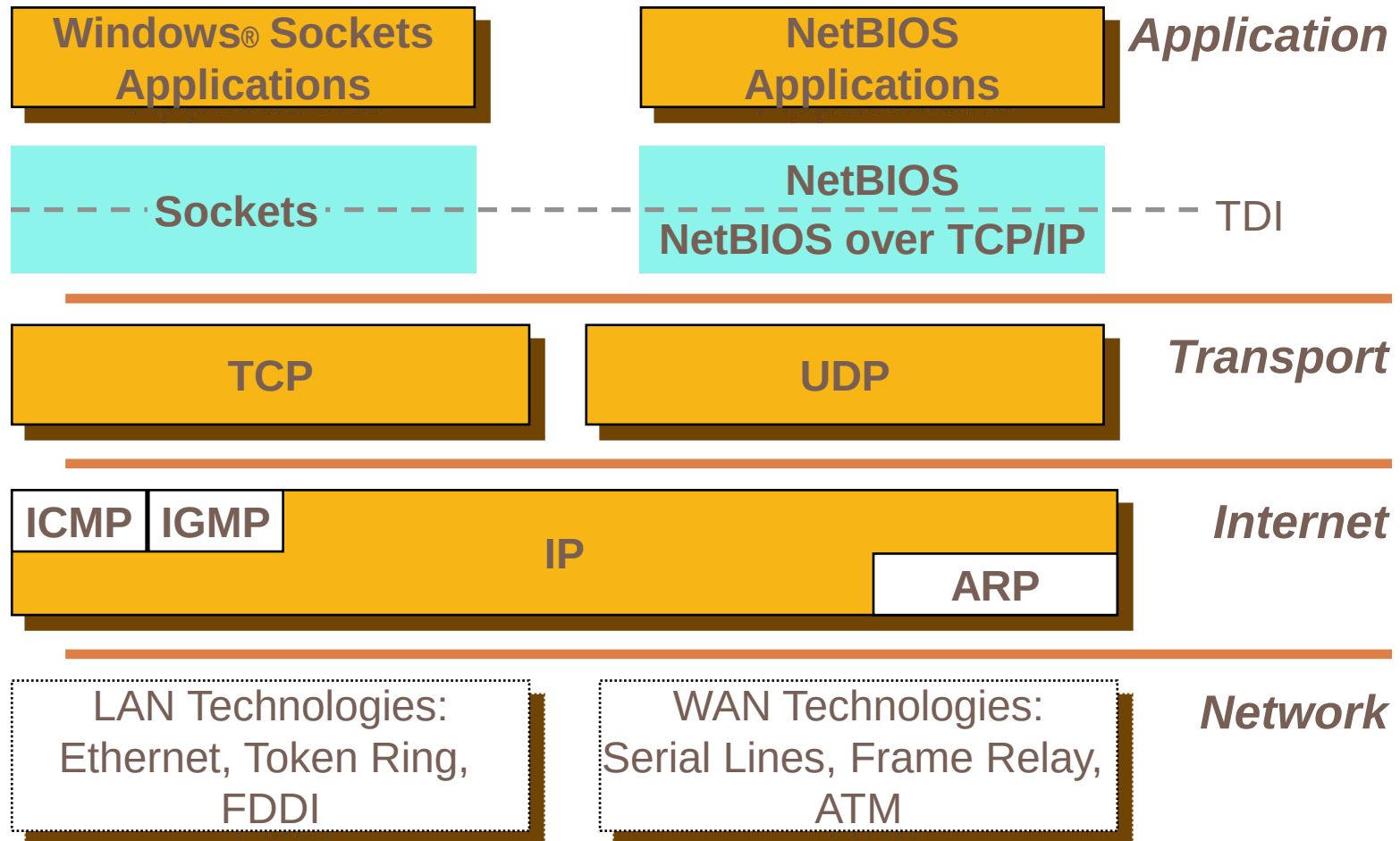
Protocol Graph

30



TCP/IP protocol suite

31



Peer To Peer and Client Server Architecture

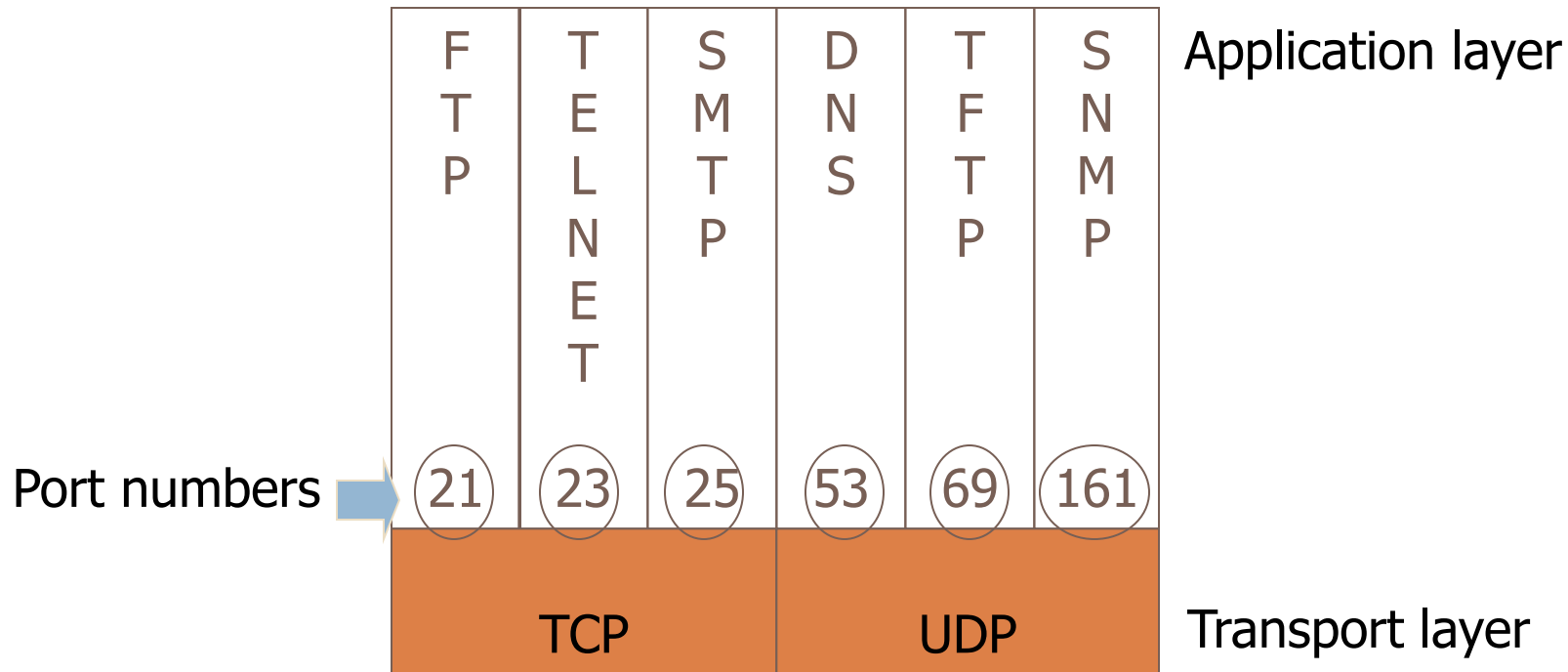
TCP Segment Format

32

Source Port					Destination Port				
Sequence number									
Acknowldgment number									
TCP header length	reserved	URG	ACK	PUSH	RESET	SYN	FIN	Window Size	
Checksum					Urgent Pointer				
Options									
Data									

Protocols & port numbers

33



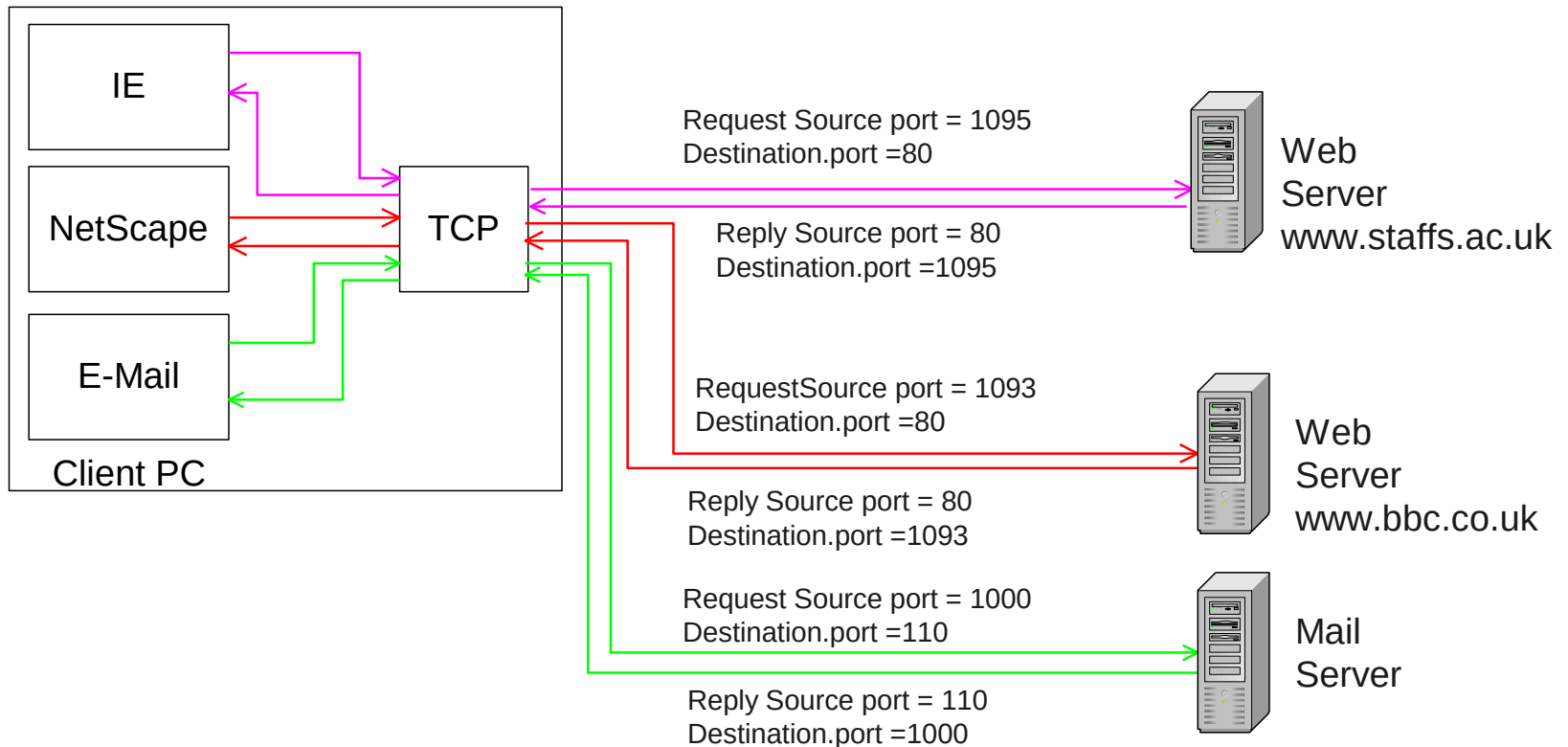
TCP client ports

34

- Q. If you have a computer running an e-mail package, 2 web browsers (Netscape and IE) how does the computer know, when a TCP/IP packet arrives which application should receive the packet ?
- A. Each application sets up its connection using a different port number, when the replies come back from the server the port number is used to send the packet to the correct connection.

TCP ports (cont.)

35



Note : Each application is allocated a different port number by the TCP software

TCP server ports

36

- The server must respond to client requests
- Q. How does the client know which port to send its request to ?
- A. “Well known port numbers” are assigned to particular services.

Port no.	Protocol	Service
80	HTTP	Web Access
110	POP3	retrieving e-mail
25	SMTP	e-mail transfer

TCP Error control

37

- The acknowledgment and sequence number fields are used to guarantee delivery of packets to the destination.
- For each packet sent out an acknowledgment must be sent back.
- If NO acknowledgment is sent back within a certain time the packet is sent again.
- Each new packet to be transmitted is allocated a new sequence no., the returning acknowledgment no. informs the sender of the next expected sequence no.
- The sequence no. is used to keep the packets in order.

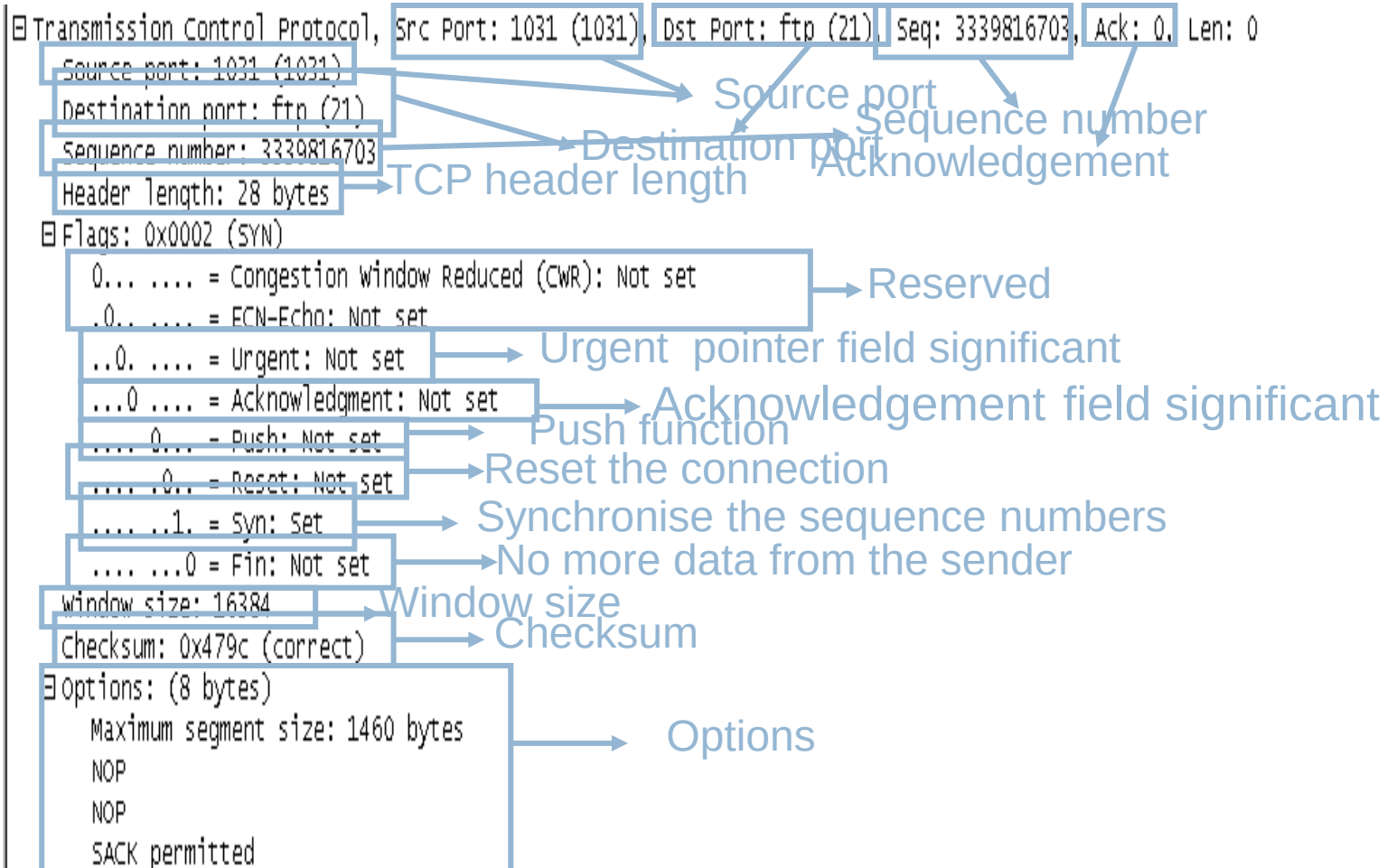
TCP flow control

38

- The window size field is used by the receiver to control the flow of packets from the sender.
- If the receiver sets the window size to 400 the sender is only allowed to send 400 bytes before stopping.
- The receiver can stop the sender by setting the window size to 0.

TCP Header

39



UDP (User Datagram Protocol)

40

□ Service

- connectionless
- provides port allocations the same as TCP
- Unreliable, does NOT guarantee delivery
- does NOT guarantee sequencing
- useful when speed is more important than reliability e.g. Internet telephony
- Transmits messages called datagrams
- Does not reassemble incoming messages
- No flow control
- NO acknowledgments

UDP segment format

41

# of bits	16	16	16	16	n
	Source Port	Destination Port	Length	Check Sum	Data...

UDP protocol example

42

```
⊞ Frame 14 (46 bytes on wire, 46 bytes captured)
⊞ Ethernet II, Src: 00:04:75:9e:1e:45, Dst: 00:08:a3:a8:55:21
⊞ Internet Protocol, Src Addr: 192.6.6.10 (192.6.6.10), Dst Addr: 192.6.6.1 (192.6.6.1)
    Version: 4
    Header length: 20 bytes
    ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 32
    Identification: 0x02f3
    ⊞ Flags: 0x00
        .0.. = Don't fragment: Not set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: UDP (0x11)
    Header checksum: 0x0000 (incorrect, should be 0xabc2)
    Source: 192.6.6.10 (192.6.6.10)
    Destination: 192.6.6.1 (192.6.6.1)
    ⊞ User Datagram Protocol, Src Port: 1074 (1074), Dst Port: 7235 (7235)
        Source port: 1074 (1074)
        Destination port: 7235 (7235)
        Length: 12
        Checksum: 0x5343 (correct)
    ⊞ Trivial File Transfer Protocol
        opcode: Acknowledgement (4)
        Block: 2
```

UDP

IP network and host id.

43

- An example with Class A addressing
 - ▣ First digit is network id , last 3 digits host id
- Note the routers use the network id to route the packets across the internet

