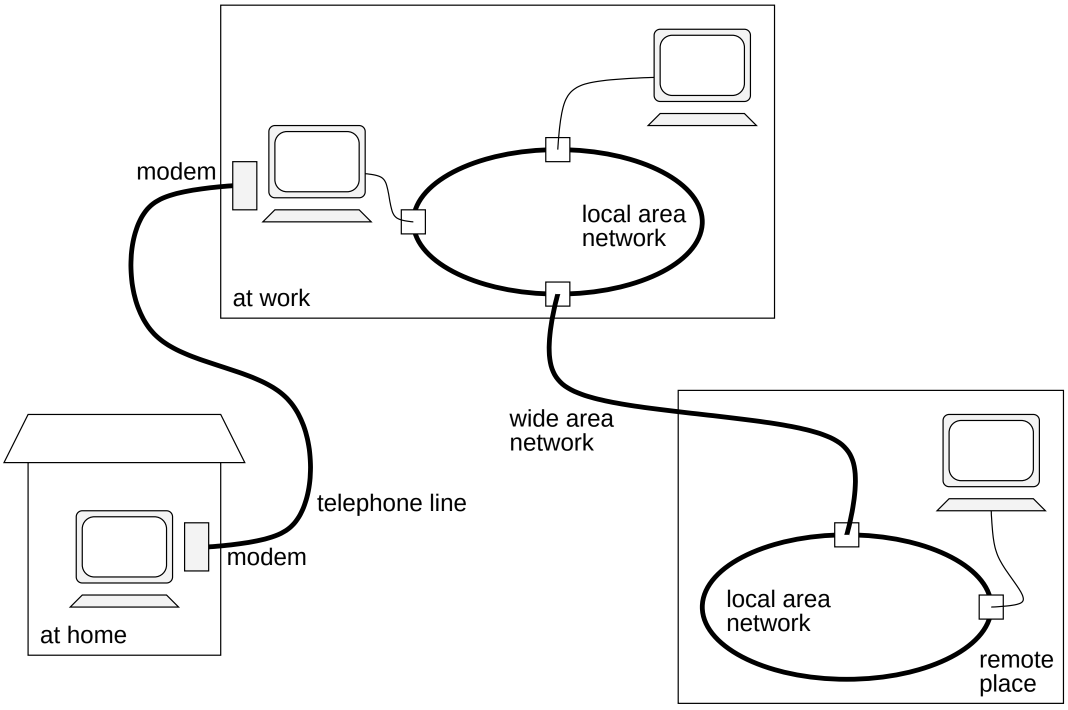
COM 6062

Network and Internetwork Architectures

Part 2: Computer Network

- Network architecture.
  - OSI model.
  - local area network (LAN) protocols.
  - wide area network (WAN).
- Internet.
  - internet protocols.
  - transport protocols.
  - internet applications.

Computer Network



modem

local area network

at work

wide area network

telephone line

modem

at home

local area network

remote place

Computer Network (2)

What is ‘ **computer network** ’ ?

a set of computers and nodes/switches connected by communication lines.

Why ‘ **computer network** ’ ?

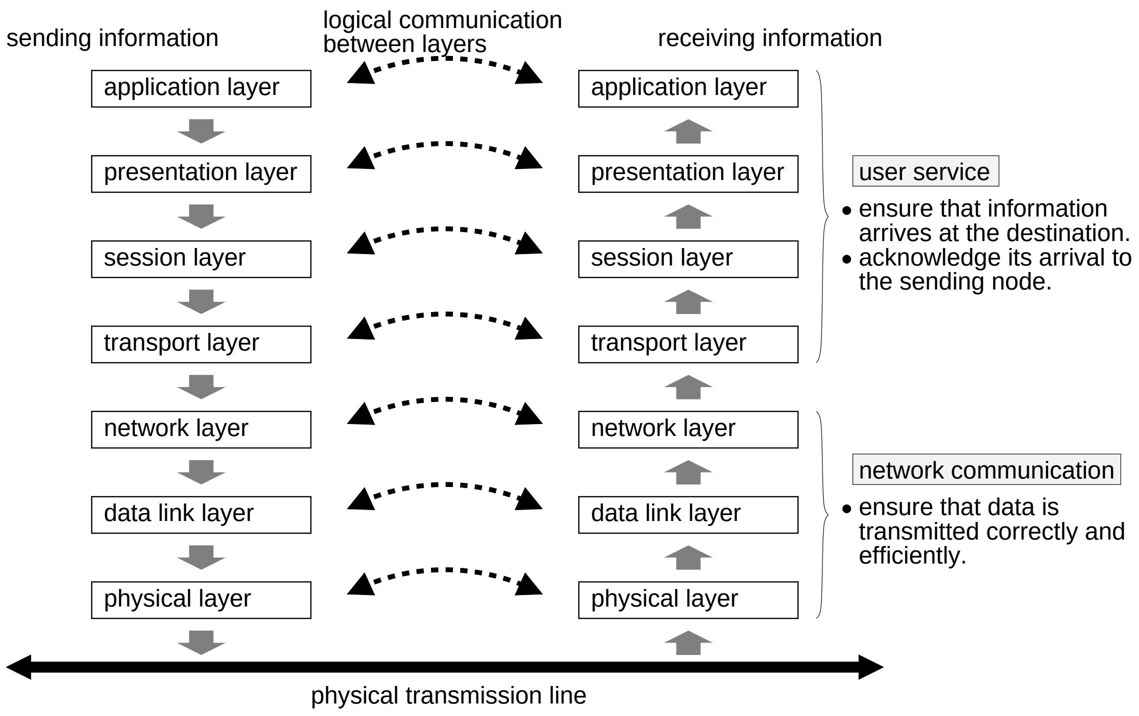
- resource sharing.
- high reliability (immune from a part crashing).
- a medium for personal communication (*e.g.*, e-mail, tele-conferencing).
- *etc.*..

Standard

Why do we need a standard for communication between computers?

- architectures are different between computers.
- a standard defines the rules and the manner in which computers begin and proceed with communication.
- a communication standard is often referred to as a ‘ **protocol** ’.

Open Systems Interconnection (OSI) Model



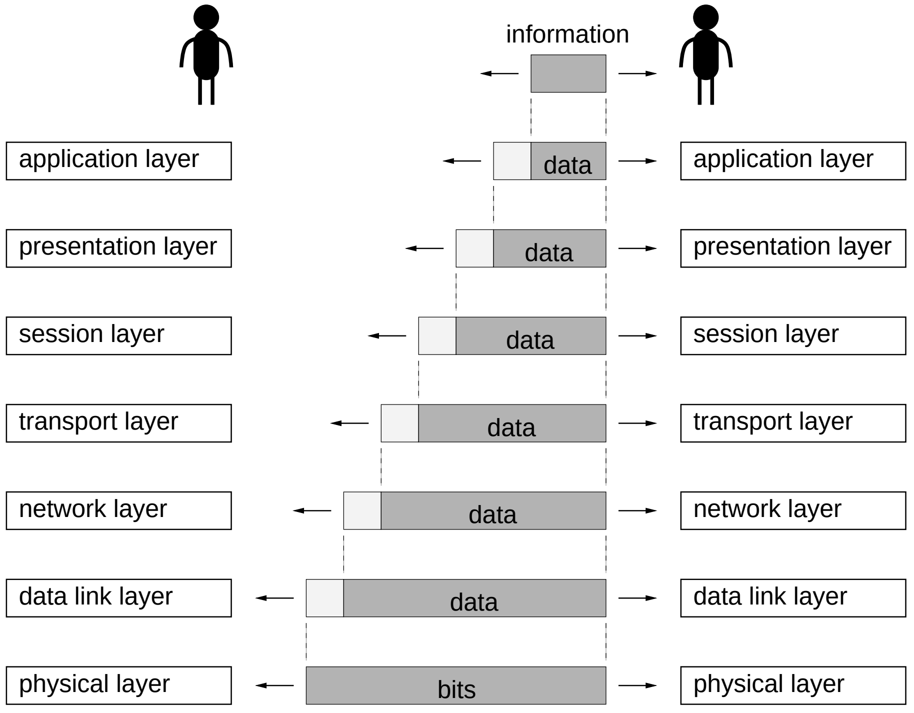
user service

- ensure that information arrives at the destination.
- acknowledge its arrival to the sending node.

network communication

- ensure that data is transmitted correctly and efficiently.

Grossly Simplified Communication by OSI Model



Physical Layer

Transmission media:

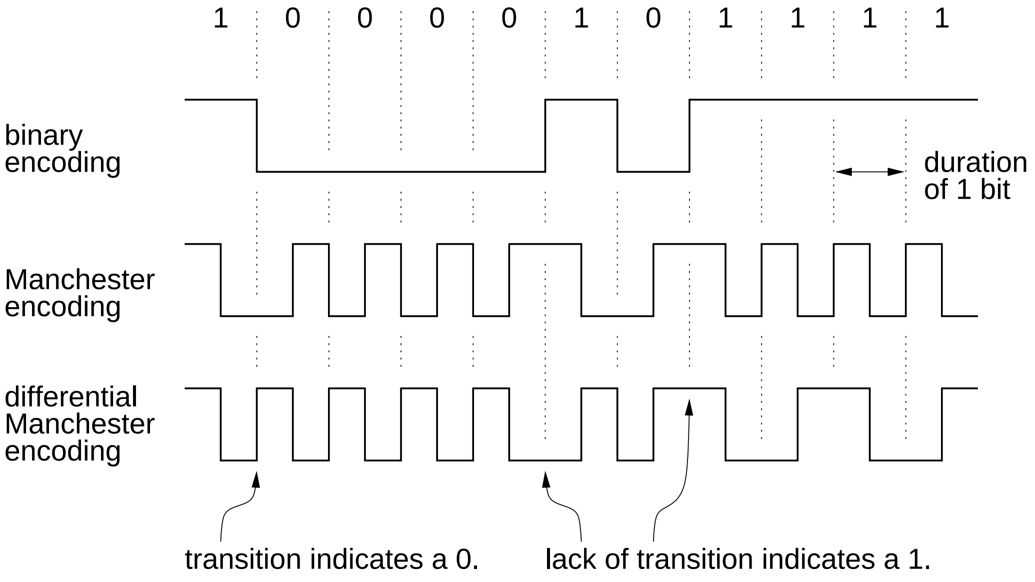
- twisted pair, coaxial cable, optical fibre.
- wireless communication — microwave (3 to 30 GHz) transmission using parabolic dishes and satellites.

Related key words (from signal processing):

- analog and digital signals, encoding schemes.
- Fourier transform, Nyquist theorem.
- noisy channel model by Shannon.
- modulation and demodulation.

Physical Layer (2)

Digital encoding schemes:



1 0 0 0 0 1 0 1 1 1 1

binary encoding

duration of 1 bit

Manchester encoding

differential Manchester encoding

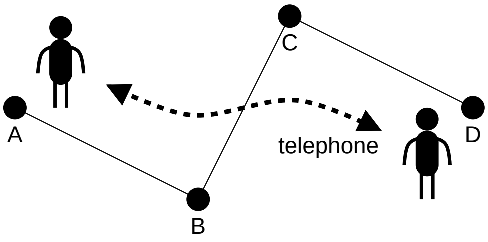
transition indicates a 0.

lack of transition indicates a 1.

Physical Layer (3)

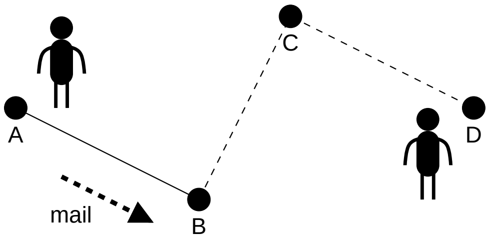
Connection strategies:

if **node A** wants to communicate with **node D**, how do we connect them?



- once a connection is established between two nodes A and D, it is maintained until one of them terminates it.

circuit switching



- a connection is made between adjacent two nodes when a message is sent, but no connection for other moment.
- destination address is attached to the message.

message switching

Physical Layer (4)

Circuit switching

- **advantage**: no transmission delay.
- **disadvantage**: the route is monopolised by one connection, thus all other connections must avoid that route.

Message switching

- the message is stored temporarily at each node, then forwarded according to the attached destination address.
- **advantage**: routes are not monopolised. The recipient needs not to accept the message immediately.
- **disadvantage**: takes longer till messages reach their destination.

Data Link Layer

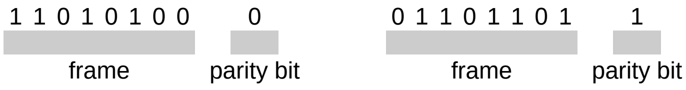
The **data link layer** sits above the **physical layer** and make sure that it (*i.e.*, physical layer) works correctly.

- **contention**: what happens if two or more nodes try to transmit data along the same medium at the same time?
  - carrier sense multiple access with collision detection (**CSMA**/**CD**): **Ethernet** (IEEE 802.3).
  - token passing: **token ring** (IEEE 802.5), **token bus** (IEEE 802.4).
- **error detection** and **correction**: how does a node know the data, it has received, is correct?
  - parity check.
  - many other error detection / correction schemes.

Data Link Layer (2)

Parity check:

- **even parity**: makes the total number of 1's even.
  odd parity: makes the total number of 1's odd.



1 1 0 1 0 1 0 0 0

frame parity bit

0 1 1 0 1 1 0 1 1

frame parity bit

(example) even parity

- the simplest **error detection** scheme, but ...
  - error can go undetected when even number of bits are incorrect.
  - cannot correct error, because it does not identify the exact position of incorrect bit(s).

Network Layer

The **network layer** builds on the **data link layer**, and provides the **transport layer** with the ability to establish end-to-end communication without worrying about the details of lower levels.

- **routing**: search for the best route(s) (*i.e.*, the cheapest and the fastest) between two points. Some considerations are:
  - environment in real network is changing dynamically.
  - a good route may attract lots of traffic, thus resulting in some traffic being eliminated. (If this happens, then the **network layer** protocol must report to the sender.)
- maintain billing information:
  - the cost of the route(s).
  - the amount of data transmitted.

Transport Layer

The **transport layer** allows the three layers above it to perform their tasks independent of a specific network architecture. It relies on the lower three layers to control actual network operations.

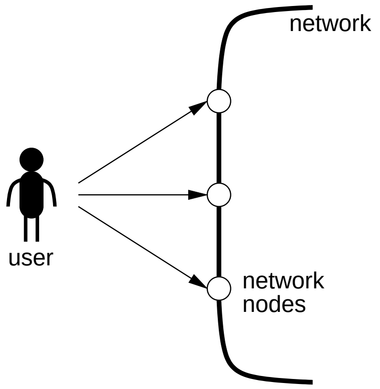
Networks are often unreliable

$\Longrightarrow$ the **transport layer** provides the **session layer** with reliable and efficient communication between two nodes.

- **multiplexing**: establish multiple connections.
- **connection management**:
  establish and release a **transport connection** between two nodes.

Transport Layer (2)

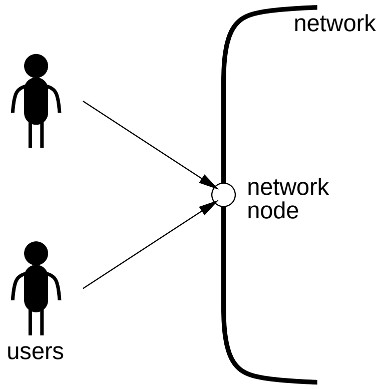
Multiplexing:



network

user

network
nodes



network

network
node

users

downward multiplexing

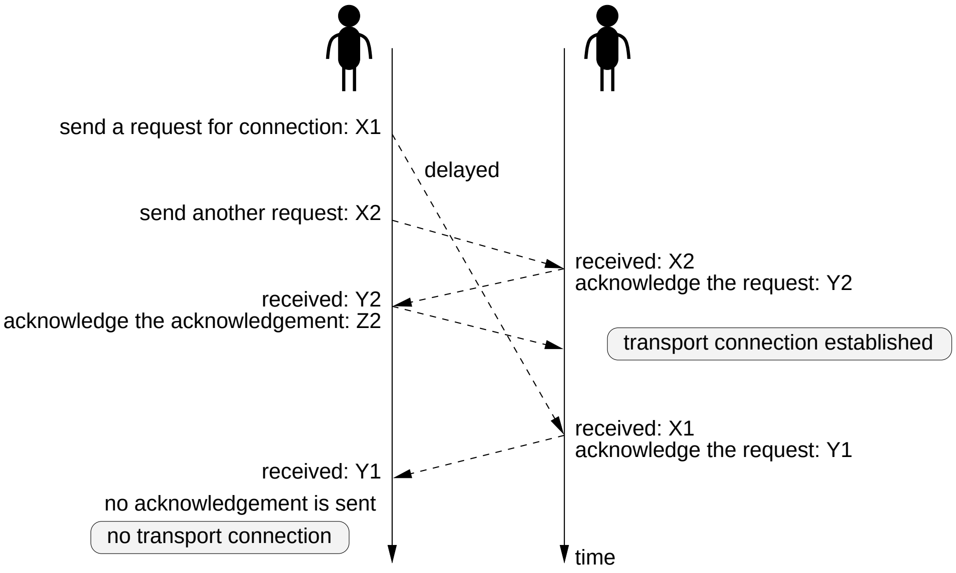
enable a transfer of a large file that exceeds a limit that a single node can handle.

upward multiplexing

enable several users to maintain constant connection to the network.
(sharing causes a slight delay, but cheaper.)

Transport Layer (3)

Connection management: three-way handshake protocol



send a request for connection: X1

delayed

send another request: X2

received: X2
acknowledge the request: Y2

received: Y2
acknowledge the acknowledgement: Z2

transport connection established

received: X1
acknowledge the request: Y1

received: Y1

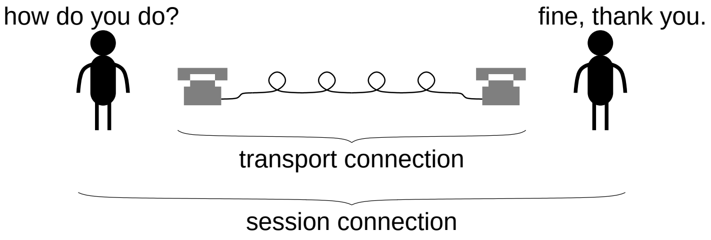
no acknowledgement is sent

no transport connection

time

Session Layer

The **session layer** contains protocols necessary to establish and maintain a connection (or **session**) between two users.

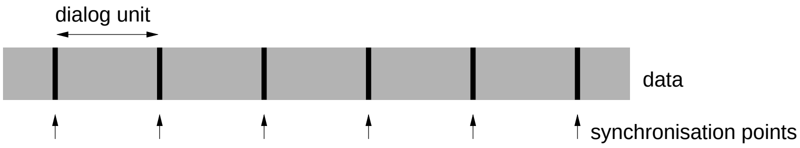


- **dialog management**
  - full duplex: data transmission can go in both directions.
  - half duplex: transmission can go in either direction, but must be alternated each time.
- **synchronisation**: error recovery.

Session Layer (2)

Synchronisation:

- data is divided into distinct **dialog units**.



- at each synchronisation point, the **session layer** (receiving side) acknowledges that a dialog unit has been successfully received.
- if error occurs, then the sender may retransmit the failed dialog unit (thus no need to resend from the beginning of whole data).

Presentation Layer

A computer does not store **information**, it stores **data**. **Information** is a meaning attached to **data**.

- the **presentation layer** defines effective communication of **information**. The following factors may vary between computers, thus require proper treatment in order that any communication takes place.
  - number of bits per word.
  - character codes (*e.g.*, ASCII).
  - expression of numbers (*e.g.*, integers, floating point numbers).
  - other data formats (*e.g.*, arrays).
- **data compression**: a way to reduce the data size while retaining the original meaning.
- data security: **encryption** and **decryption**.

Application Layer

The **application layer** contains network applications.

- **electronic mail**.
- **file transfer protocol** (**ftp**): allows a user to connect to a remote system across the network, examine data, and copy to the user's system.
- **virtual terminal**: allows a user at a terminal to remote login to a computer across the network. Once connected, the user interacts as if the computer were on-site.
- **distributed system**: allows individual computers to access or execute common resources across the network.

LAN and WAN

LAN (**local area network**):

- typically connects computers, printers, disc drives, and *etc.*, located in one building or a cluster of buildings.
- **Ethernet**, **token ring**, **token bus**.

WAN (**wide area network**):

- connects devices located throughout a city, a country, and the world.
- **internet**.

Contention Schemes

Contention: access to the communication medium from many entry points.

Aloha protocol

- anarchistic approach:
  - any node may transmit a signal whenever it wants.
  - in case of collision, a node waits a random time and tries again.

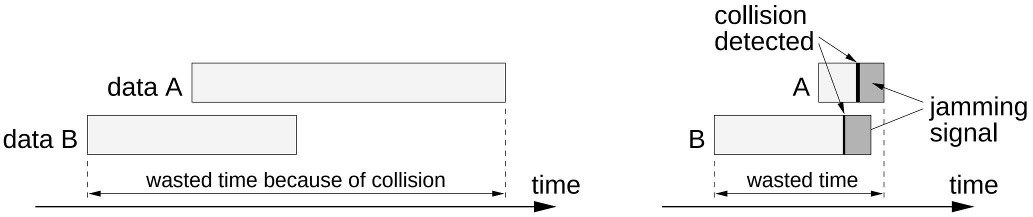
Carrier sense multiple access (**CSMA**)

- protocol:
  - listen to the transmission medium for any activity.
  - transmit if no activity is detected; otherwise wait.
- able to reduce the number of collisions from a simple **Aloha protocol**, but cannot eliminate them.

Contention Schemes (2)

Collision detection (**CD**)

- reduces the time of collision:
  - when a collision is detected, both nodes stop transmitting signals.
  - then, each will send a jamming signal (a type of electronic scream) to ensure that all nodes know a collision has occurred.



collision without detection

collision with detection

Contention Schemes (3)

CSMA/CD

- combination of **CSMA** and **CD** protocols:
  - if a medium is busy, a node waits.
  - if a medium is quiet, a node transmits a signal and continues to listen to the medium activity.
  - once a collision is detected, a node immediately stops transmission and sends a short jamming signal.
  - after the collision, it waits a random amount of time before another trial is made.
- a protocol for the **Ethernet**.

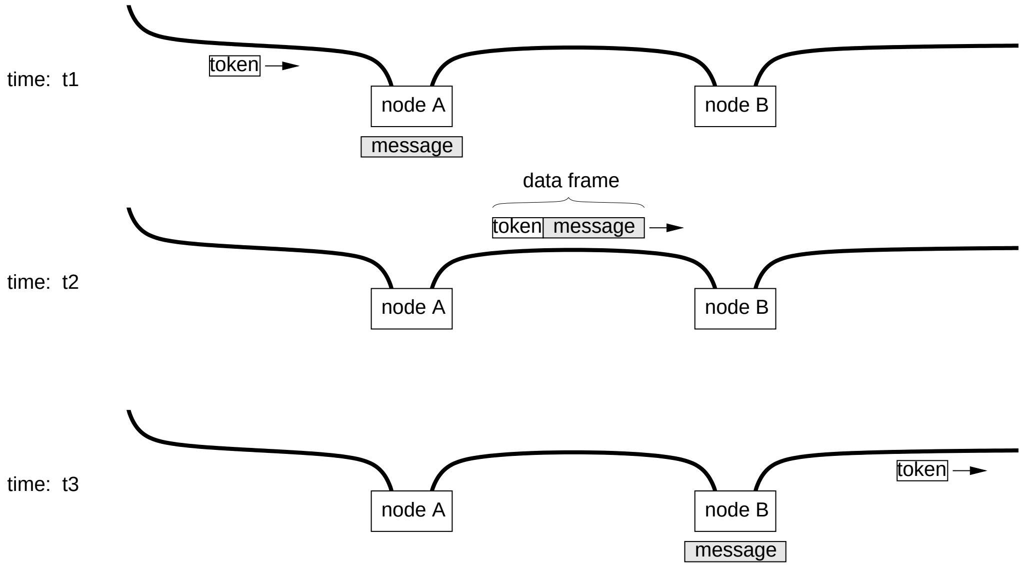
Contention Schemes (4)

Token passing

- protocol:
  - a unique bit stream, called a **token**, is circulating along all network nodes.
  - if a node wants to transmit any message, it must wait until it receives a **token**.
  - once a **token** arrives and it does not carry any data, a node may append the message and the destination address to the **token**.
- a protocol for the **token ring** and the **token bus**.

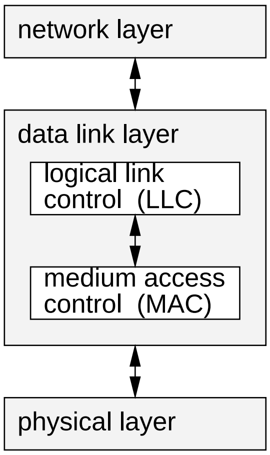
Contention Schemes (5)

Token passing



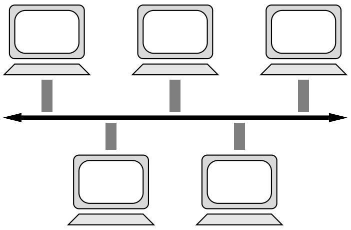
LAN Protocols

Relation to the OSI:

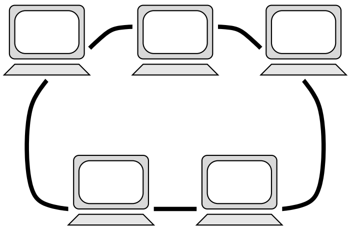


LLC handles logical links between nodes.

MAC handles access to the transmission medium.



bus topology: Ethernet, token bus

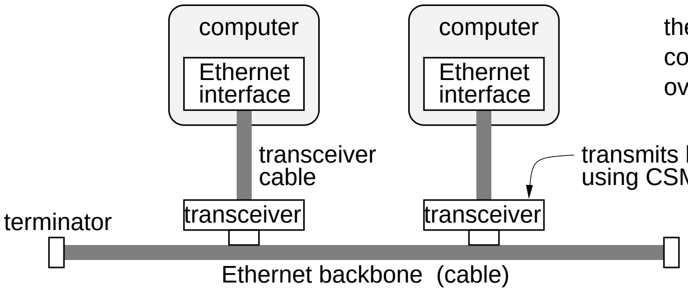


ring topology: token ring

- local area network (LAN) standards are MAC protocols.

Ethernet

Ethernet connection:



the Ethernet interface contains the MAC unit for overall protocol control.

transmits bits onto the cable using CSMA/CD contention.

- advantage:
  - the protocol needs not be changed when new nodes are added.
- disadvantage:
  - collisions do occur and there is no theoretical upper limit.
- works well in general, but occasionally substantial delays may happen.

Ethernet (2)

Frame format:

preamble: is a 7-octet pattern consisting of alternating 0's and 1's, and is used for synchronisation.
(Synchronisation establishes the sampling rate.)

frame delimiter: is a special pattern ' 10101011 ', indicating the start of a frame.

destination address: indicates that the destination is either a specific node (if the first bit is 0), or all nodes (if the first bit is 1).

source address: specifies where the frame comes from.

data field length: contains the number of octets in the combined data / pad fields.

data / pad field: must be between 46 and 1518 octets.
If there is not enough data, extra octets are added (padded) to make up the difference.

frame check sequence: is a error checking code (32-bit CRC).

octet: a sequence of 8 bits.

preamble	7 octets
frame delimiter	1
destination address	2 or 6
source address	2 or 6
data field length	2
data / pad field	46 to 1518
frame check sequence	4

Ethernet (3)

Frame transmission:

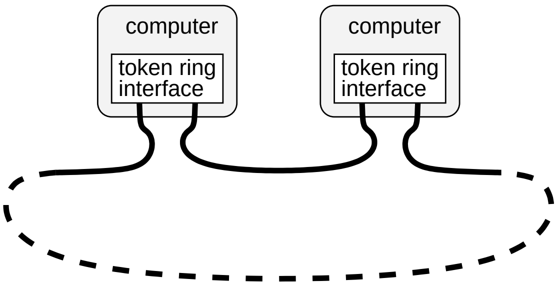
1. Ethernet interface (containing the **MAC unit**) monitors the signal on the bus, then starts transmission of a frame through the **transceiver**.
2. the **transceiver** simultaneously monitors a collision.
3. if a collision has occurred, the **MAC unit** transmits a jamming signal.

Frame reception:

1. once the **MAC unit** detects the presence of an incoming signal from the **transceiver**, synchronisation is achieved using **preamble**.
2. the **MAC unit** tests **delimiter**, then, from **destination address**, determines if this frame should be received.
3. if so, the rest of the frame is processed and passed to the higher layer.

Token Ring

Token ring connection:



the token ring interface contains the MAC unit for overall protocol control.

- advantage:
  - collisions cannot occur.
- disadvantage:
  - a malfunction at one node may destroy/duplicate a token, thus affecting the entire network.

Token Ring (2)

Frame format:

using the differential Manchester encoding scheme.
signal J: starts like a 0, but no transition in the middle.
signal K: starts like a 1, but no transition in the middle.

starting delimiter: a pattern ‘ JK0JK000 ’.

access control: ring maintenance.

frame control: ring maintenance.

destination address: same as the Ethernet.

source address: same as the Ethernet.

data field: 0 to 5000 (typical) octets.

frame check sequence: same as Ethernet.

ending delimiter: a pattern ‘ JK1JK1IE ’.
‘ I ’ bit: 0 for the last frame, 1 otherwise.
‘ E ’ bit: 1 whenever an error is detected.

frame status: ring maintenance.

token

starting delimiter	1 octet
access control	1
ending delimiter	1

data frame

starting delimiter	1 octet
access control	1
frame control	1
destination address	2 or 6
source address	2 or 6
data field	0 to 5000
frame check sequence	4
ending delimiter	1
frame status	1

Token Ring (3)

Frame transmission:

1. a service request (from the upper layer) to transmit a data message is first encapsulated by the **MAC unit** of the token ring interface.
2. the **MAC unit** awaits a **token**.
3. once a **token** is received, the **MAC unit** transmits a **data frame**.

Frame reception:

1. when the **MAC unit** detects reception of a **data frame** by recognising the special bit sequence, it determines, from **destination address**, if this frame should be accepted.
2. if so, the **MAC unit** processes its **data field**, then transmits a **token**.
3. otherwise, it retransmits a received **data frame**.

Token Bus

Bus topology, but operates by the same principle as the **token ring**.

- require logical neighbours: a node receives a **token** (or a **data frame**) from its **predecessor** node and sends to its **successor** node.
- advantage:
  - upper bound on the time each node must wait before getting the token.
- disadvantage:
  - difficult to add/remove a node.

Support from those involving in factory automation and process control.

- bus topology fits better in factories, but for realtime environment, do not want the **Ethernet**, where there is no theoretical limit for delay.

Interconnecting LANs

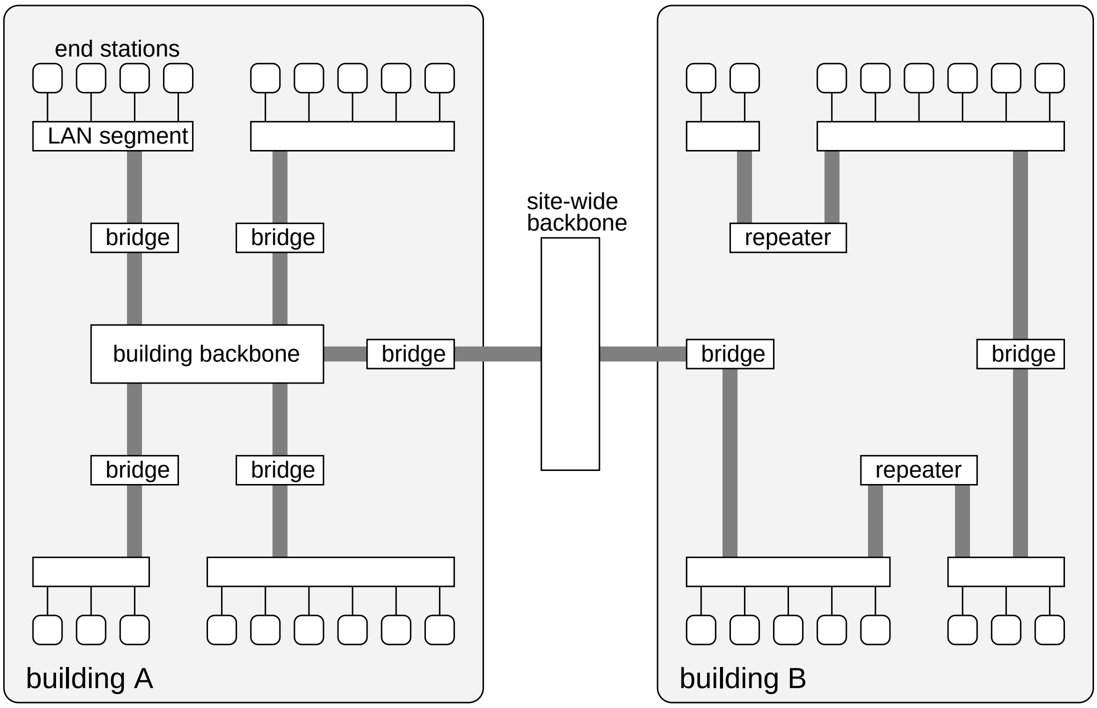
Demands are increasing for communication between separate computer systems, however, physical restrictions exist for **local area network** (**LAN**) protocols:

- number of computers (more computers, more traffic).
- cable length (up to several hundred meters).
- *etc*...

Methods for connecting between LANs within a small geographical area:

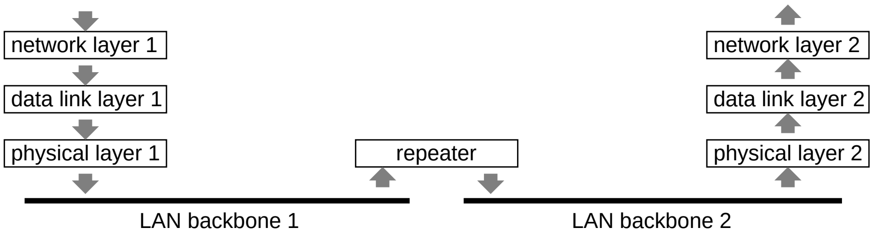
- layer 1 connection: **repeater**.
- layer 2 connection: **bridge**.

Typical Establishment-Wide LAN



Repeater

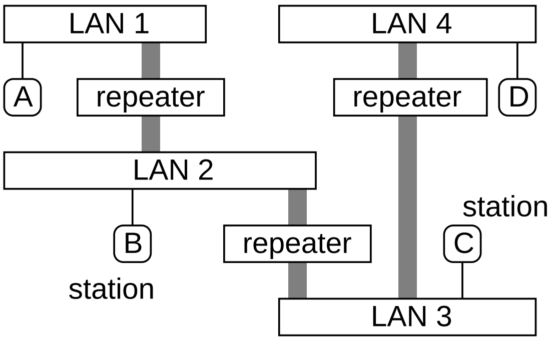
Layer 1 (**physical layer**) connections:



- a **repeater** connects between LANs having the same protocol and the same frame format.
- nodes on interconnected LANs do not recognise the existence of a **repeater**.
  - no need to change the network protocol.
- physically, it is simply a buffer (*i.e.*, an amplifier) of electrical signal.

Repeater (2)

LANs connected with **repeaters**:

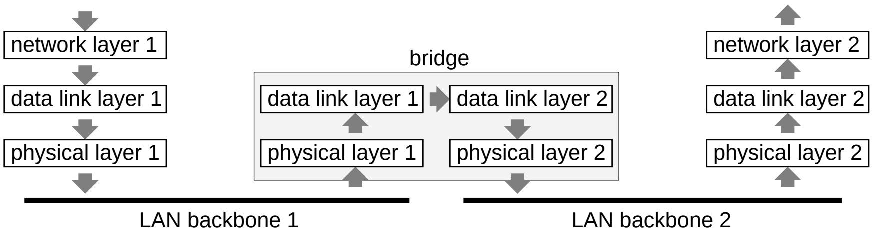


station A sees every frame transmitted by, say, station C, regardless of whether the frame is addressed to station A.

- heavier traffic in the network.
  - difficult to build a very large LAN system using **repeaters** only.

Bridge

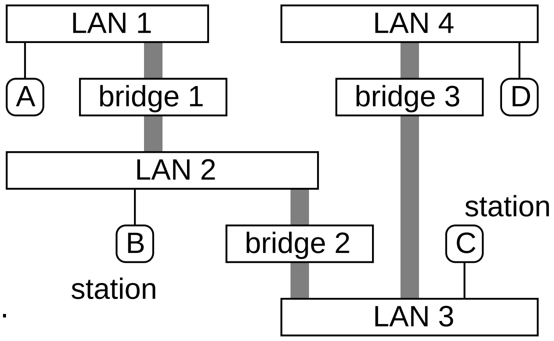
Layer 2 (**data link layer**) connections:



- a **bridge** is a connector with the ability to execute a subset of a protocol.
  - error correction.
  - frame formatting.
- it can connect between LANs having different protocols and different frame formats.

Bridge (2)

LANs connected with **bridges**:



suppose bridge 1 receives any frame from the LAN 2 side, it is transmitted to LAN 1, only if it is destined for any station on LAN 1.

- a **bridge** selectively accepts/rejects frames based on their destination.
  - can reduce unnecessary traffic, thus more efficient.
  - can accommodate some security feature.

Bridge (3)

Bridging different LAN protocols — some considerations:

- different LANs may have different bit rates

  ⟹ requires a sufficient buffer space when transmitting frames to the slower side.
- each LAN has a different frame format

  ⟹ requires reformatting.
- each LAN has a different maximum frame size (*e.g.*, 1518 octets for the **Ethernet**, 5000 octets for the **token ring**)

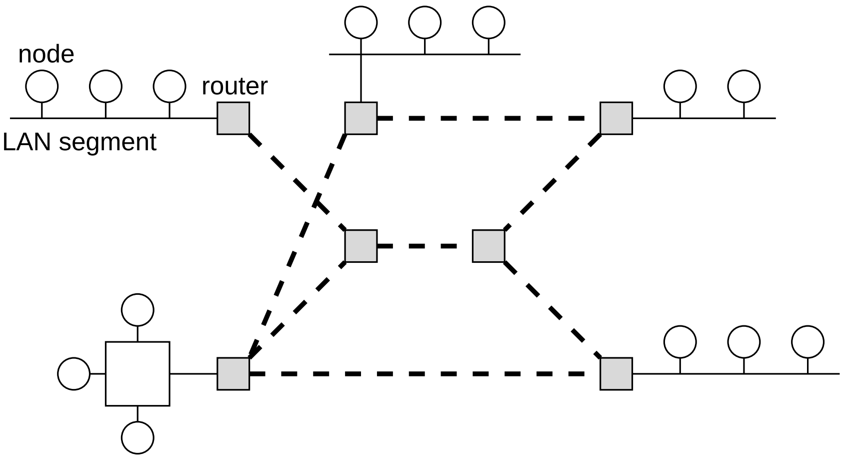
  ⟹ break a large frame into smaller ones.
- *etc.*..

Bridge (4)

Bridge routing — how does a bridge know when to accept a frame?

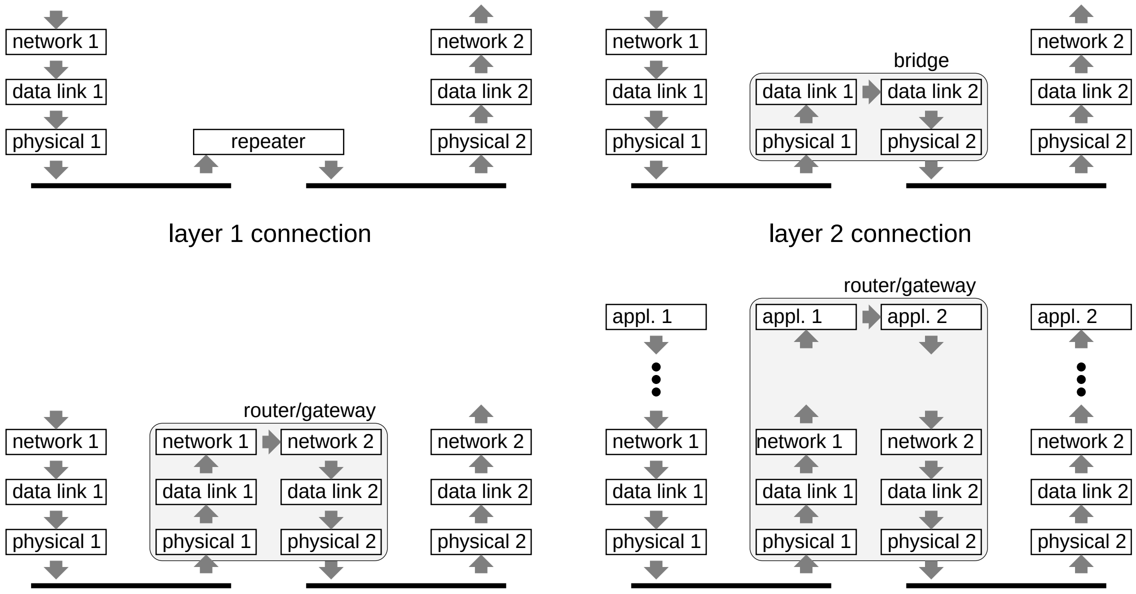
- **fixed routing bridge**:
  - requires a **routing table** that describes existing destinations.
  - drawback: needs to update the table each time configuration changes.
- **transparent bridge**:
  - has the ability to create and update its own **routing table**.
  - route learning: whenever it receives a frame, it examines the source address, thus identifying the existence of a certain station.
- **source routing bridge**:
  - network software at a sending station determines a route to the destination and stores it in the **route designator** of a frame.

Wide Area Network



connecting between LANs

Wide Area Network (2)


network 1
data link 1
physical 1
repeater
network 2
data link 2
physical 2

layer 1 connection

network 1
data link 1
physical 1
bridge
data link 1
physical 1
data link 2
physical 2
network 2
data link 2
physical 2

layer 2 connection

network 1
data link 1
physical 1
router/gateway
network 1
data link 1
physical 1
network 2
data link 2
physical 2
network 2
data link 2
physical 2

layer 3 connection

appl. 1
network 1
data link 1
physical 1
router/gateway
appl. 1
network 1
data link 1
physical 1
appl. 2
network 2
data link 2
physical 2
appl. 2
network 2
data link 2
physical 2

layer 7 connection

two different networks may be connected at any level of OSI layers

Wide Area Network (3)

Network routing:

- one node may send **packets** (*i.e.*, a message that is de-assembled to fixed size units) to another which is not locally connected, but situated in throughout a city, a country, and the world.
- network protocols search for the best route(s) (usually, the cheapest and the fastest) between two points

  ⟹ routing algorithms.
- network routing is not straightforward:
  - environment in real network is changing dynamically.
  - a good route may attract lots of traffic

    ⟹ leading to congestion, or even deadlock.

Congestion

Congestion:

an excessive buildup of packets at one or more network nodes caused by

- a failure of one or more network nodes/links,
- the number of packets exceeding the network's ability,
- *etc.*..

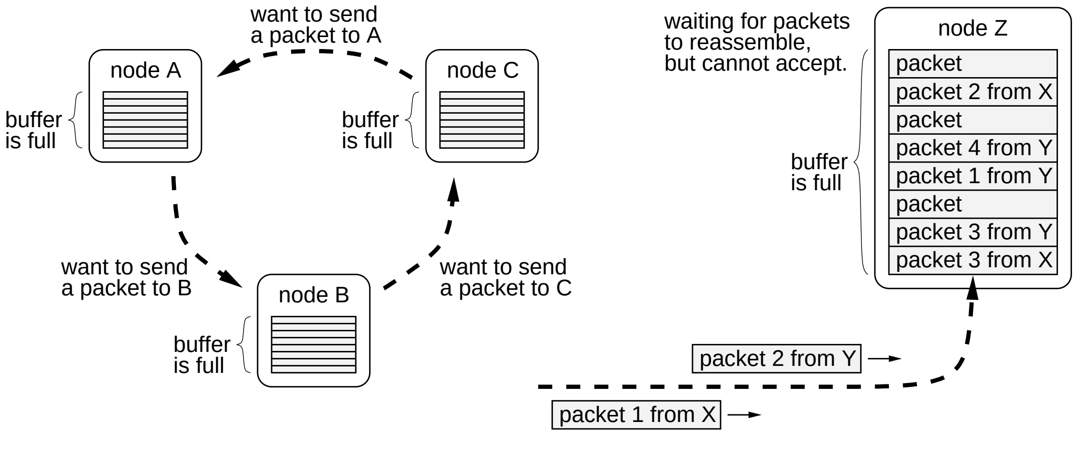
It may cause a chain reaction because the congestion at one node hampers its ability to receive packets from other nodes.

(a good analogy to a motorway/road traffic problem)

Deadlock

Deadlock:

congestion becomes so severe that nothing moves.



store-and-forward deadlock

reassembly deadlock

Congestion Control

packet elimination:

- if an excessive buildup of packets occurs, eliminate some of them.
- unlucky senders' destroyed packets do not reach their destination:
  - senders must determine the fate of their packets.
  - adverse effect is limited (for most cases).

flow control:

- flow control protocols may regulate the number of packets to be transmitted between two nodes.
- congestion can still occur, if too many packets are flowing into a node located between the source and the destination.

Congestion Control (2)

buffer allocation:

- a source node requests nodes on the route for reserving a sufficient buffer space for forthcoming packets.
- if the request is rejected, a source node looks for another route.

choke packet:

- each node monitors the activity on its links, and puts any link(s) to a warning state if its utilisation exceeds some level.
- for any packet ongoing the link under a warning state, its source node is notified by a **choke packet**.
- a node, that has received a **choke packet**, reduces the amount of transmission to that link.

Network Routing

Routing table:

- we wish to find the least costly route.
- given a network, a routing table shows the associated cost for communicating with each destination node.
- it does not specify the entire route for each destination, but it only indicates the next node.
- the cheapest route between nodes A and F is
  A ⟶ B ⟶ E ⟶ F
  and its associated cost is 7.
- but, how can we find such route ?

dest.	next	cost
B	B	2
C	C	1
D	C	5
E	B	5
F	B	7

node A

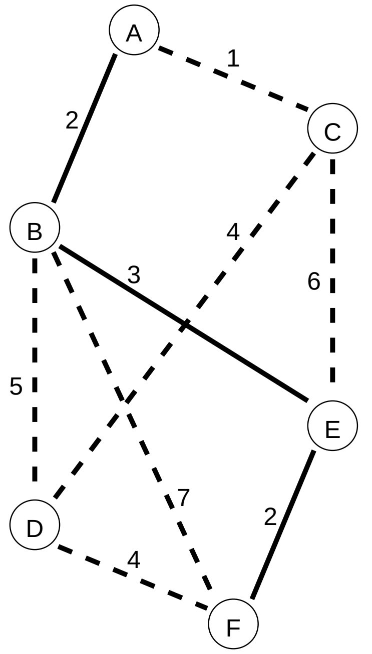
dest.	next	cost
A	A	2
C	A	3
D	D	5
E	E	3
F	E	5

node B

dest.	next	cost
A	B	5
B	B	3
C	C	6
D	F	6
F	F	2

node E

routing tables for nodes A, B, and E



network and associated connection cost

Types of Routing

Centralised routing:

- all interconnection information is maintained at a single node.
- this central node then broadcasts the information to the rest of nodes.
- simple, but a failure of the central node, or any link connected to it, may affect entire network.

Distributed routing:

- no central control, but each node determines and maintains the routing information independently.
  - a node finds its neighbours and calculates the associated costs.
  - each neighbour, in turn, repeats the same procedure.
- complex, but a failure of a node or a link may cause little disruption.

Types of Routing (2)

Static routing:

- once routing information is determined, it will not be modified.
- simple, but insensitive to changing condition.

Adaptive routing:

- each node updates routing information according to dynamically changing network condition.
- pitfall: some packet may shuttle back and forth among several nodes, thus never making any progress towards its eventual destination.
- can provide the recent information regarding link cost, but high overhead for updating routing information frequently.

Hierarchical Routing

It is difficult to determine a route when too many nodes exist in a network

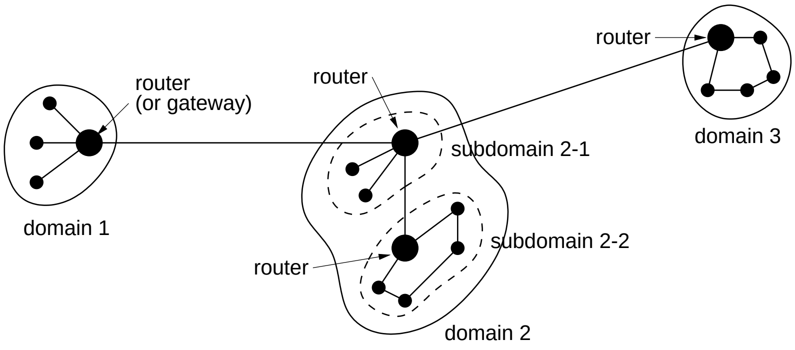
$\Longrightarrow$ structure the network hierarchically.

Some features of **hierarchical routing**:

- all nodes are divided into groups, called **domains**, each of which is a separate and independent network.
- a route between two nodes in a common domain is determined using the domain's protocol.
- each domain has a designated node, referred to as a **router** or a **gateway**, that searches for route(s) between domains.
- if a domain is large, it may consist of multiple **subdomains**.

Hierarchical Routing (2)

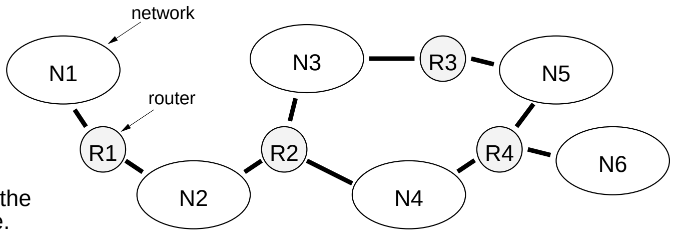
The **Internet** uses a hierarchical structure:



RIP

RIP (**routing information protocol**):

- **routers** in the network let each other know the shortest route to a specified network.
- typically, it uses a **hop count** (the number of intermediate routers) to measure the distance.



N1 to N2: 1 hop.

N1 to N4: 2 hops using the shortest route.

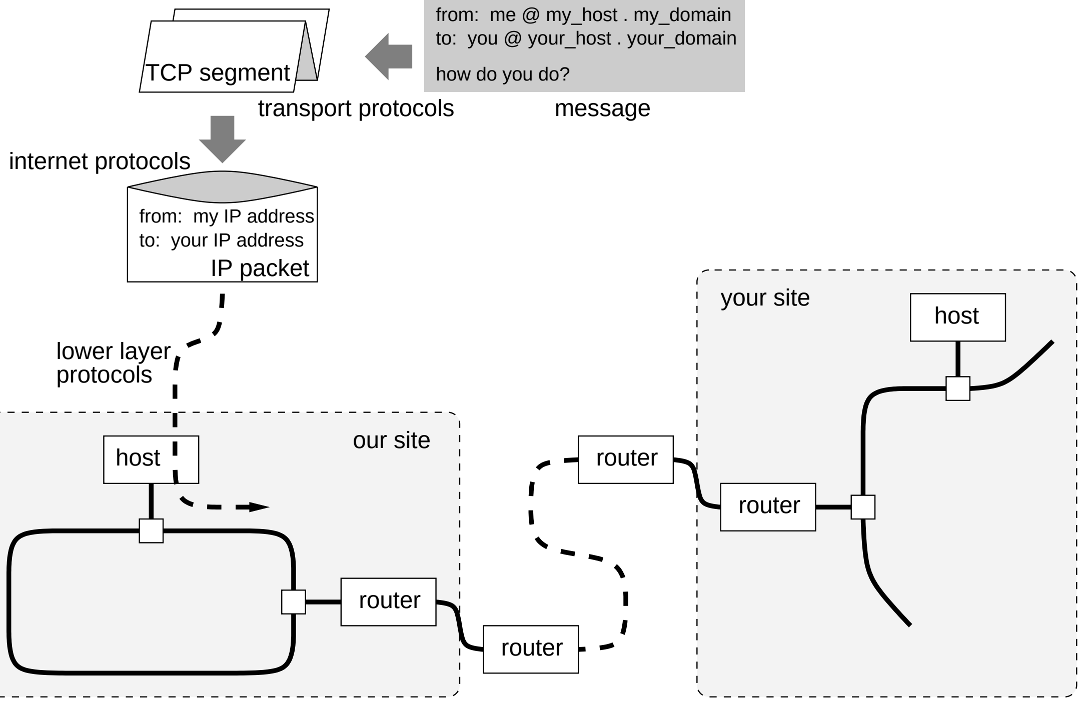
RIP (2)

Procedure:

1. **R1** sends a message to **N2** that it can get to **N1** in 1 hop.
2. because **R2** is connected to **N2**, it now knows that it can get to **N1** in 2 hops, and stores this fact to its routing table.
3. **R2** broadcasts to **N3** that it can get to **N2** and **N4** in 1 hop, and to **N1** in 2 hops.
4. **R3** receives and stores the information it received over **N3**.
5. **R3** broadcasts to **N5** that it can get to **N3** in 1 hop, to **N2** and **N4** in 2 hops, and to **N1** in 3 hops.

. . . by repeating this procedure, **N6** eventually knows it can get to **N1** in 3 hops via **N4** and in 4 hops via **N5**.

Internet



from: me @ my_host . my_domain
to: you @ your_host . your_domain

how do you do?

TCP segment

transport protocols

message

internet protocols

from: my IP address
to: your IP address
IP packet

lower layer protocols

our site

host

router

router

router

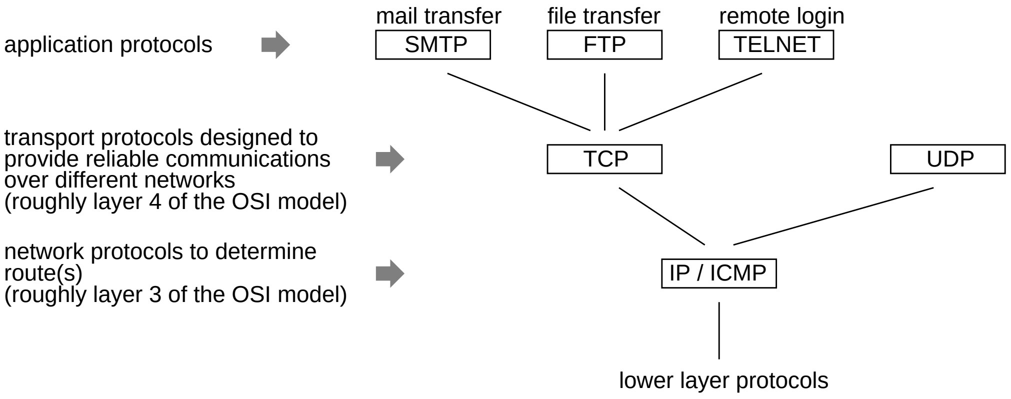
your site

host

router

TCP / IP Suite

Internet connects networks, each of which runs a **TCP/IP**:



application protocols

mail transfer
SMTP

file transfer
FTP

remote login
TELNET

transport protocols designed to provide reliable communications over different networks
(roughly layer 4 of the OSI model)

TCP

UDP

network protocols to determine route(s)
(roughly layer 3 of the OSI model)

IP / ICMP

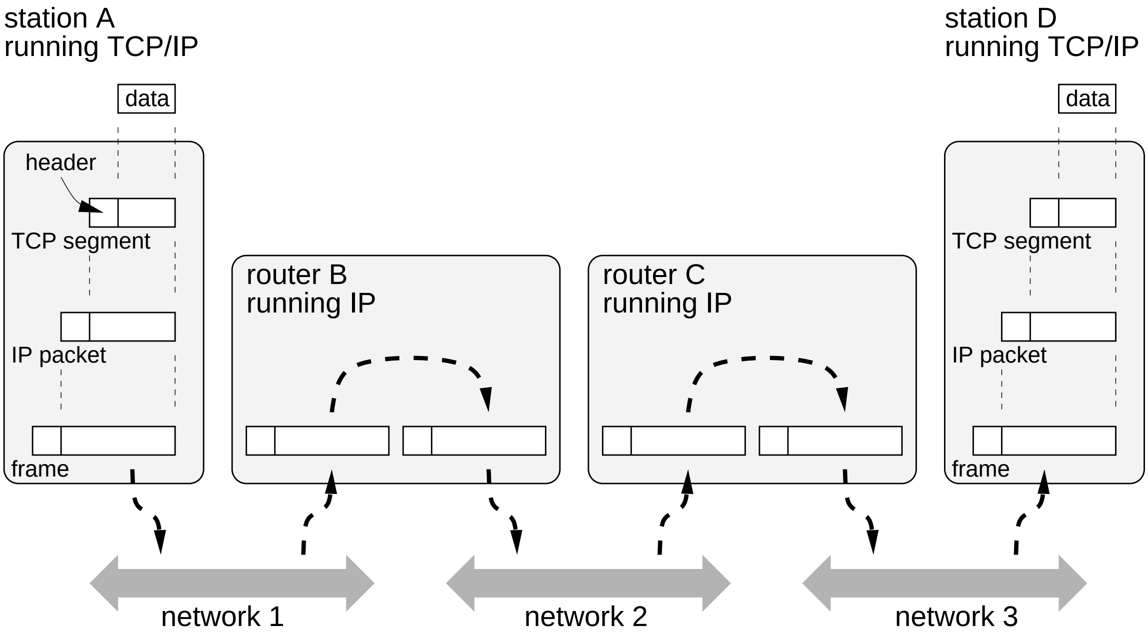
lower layer protocols

- **IP** (**Internet protocol**): roughly the **network layer**,
- **TCP** (**transmission control protocol**): roughly the **transport layer**,

although they are not part of the OSI model.

TCP / IP Suite (2)

Transmitting packets over different networks:



TCP / IP Suite (3)

1. At station A —
   - **TCP**:
     – creates a **TCP segment** containing **data**, then ‘sends’ to station D.
   - **IP**:
     – intercepts the TCP segment, and creates an **IP packet** containing the TCP segment. It also determines route(s).
   - **data link layer**:
     – create a **frame** and send it to router B via network 1.
2. Router B’s **IP** examines the address in the packet and determines that it should go to router C via network 2.
3. This procedure repeats till it reaches the final destination.

Internet Protocols

Internet protocol (**IP**) provides a **connectionless** best effort service to the transport layer protocols.

(**connectionless** service ⟺ **connection-oriented** service)

IP forms **IP packet** (**IP datagram**), given a **TCP**/**UDP** segment, then transmits through a router to its destination.

Internet address (**IP Address**):

- unique internet wide identification assigned to each host.
- consisting of a **network number identifier** (**netid**) and a **host number identifier** (**hostid**).
- centrally managed by the **Internet Corporation for Assigned Names and Numbers** (**ICANN**).

IP routing algorithms to decide the route to the final destination.

IP Packet

IP packet:

header length: specifies the number of 32−bit words in the packet header.

datagram length: specifies the number of octets in the entire packet, providing a maximum length of 65536 octets.

protocol: indicates the higher layer protocol using this packet (e.g., 6 for a TCP segment, 17 for a UDP segment).

header checksum: is used for error detection of packet header only (data, corresponds to TCP or other, has its own error detection).

source / destination IP address: contains IP addresses for the sending and receiving stations.

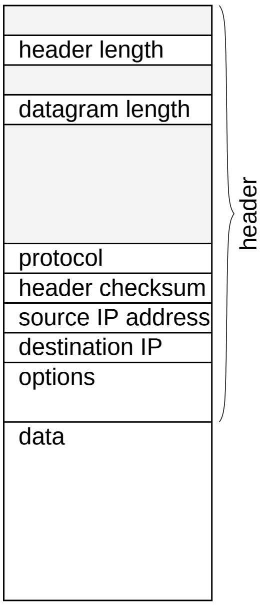
options: is not always required, but it can be used for special treatment.

record route: traces the route a packet takes.

timestamp: recoreds the time at which each router handles the packet.

source route: allows the sender to specify the route to be taken.

data: contains the data provided by the higher layer.



header length

datagram length

protocol

header checksum

source IP address

destination IP

options

data

header

IP Addresses

Domain name

- typical form: (e.g.)

  host . subdomain(s) . domain — hazel . dcs . shef . ac . uk
- e−mail is delivered by a host to a user specified by ‘ @ ’.

IP address

- translated from domain name using a protocol called the DNS (domain name server).
- form of four 8−bit binary numbers: (e.g.)

  □ – □ – □ – □ — 10001111−10100111−00001000−01010100

  this 32−bit IP address is unique to each host. (equivalently, ‘ 143 . 167 . 8 . 84 ’ by a dotted decimal notation)

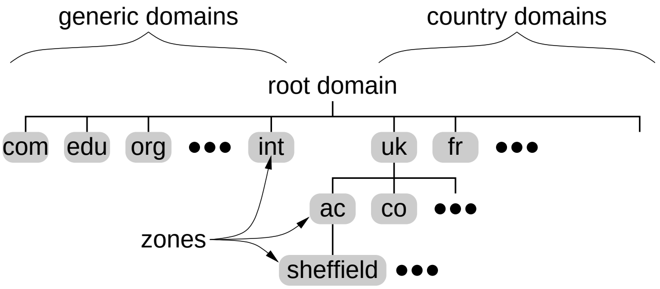
Physical address

- used by the underlying physical network (different from IP addresses).
- only local significance, but none on a grobal IP scale.
  (Recall that IP packets are stored in frames when they travel the networks, and frames contain addresses depending on the data link control protocols.)

IP Addresses (2)

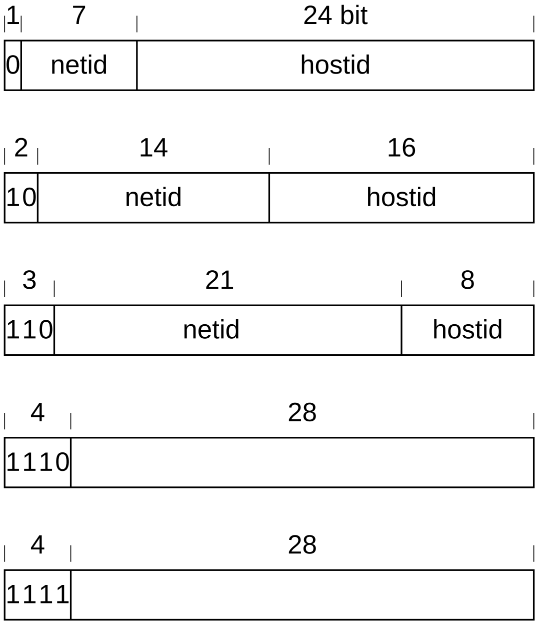
Name server:

- provides scalability to the entire name space.
- administrates names and IP addresses of hosts within the zone.
- knows **domain names** and **IP addresses** of lower/higher level **name servers** in the hierarchy.
- searches a resource record of a given destination host name, located in a different zone.



IP Addresses (3)

Class based addresses:



class A netid: 2^7 newworks
hostid: 2^{24} hosts
(eg) nation wide network

class B netid: 2^{14} newworks
hostid: 2^{16} hosts
(eg) large organisations

class C netid: 2^{21} newworks
hostid: 2^8 hosts
(eg) small site networks

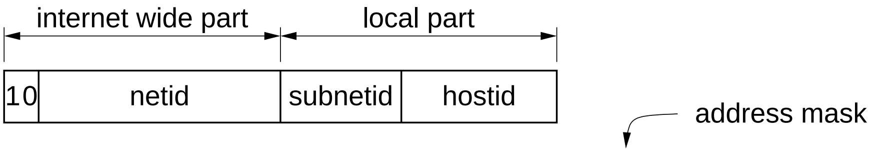
class D (special use)

class E (reserved)

IP Addresses (4)

Subnetting:

(eg) class B



11111111111111111111111000000000 (255.255.254.0)

- a single **netid** is allocated for each site.
- a site may contain multiple LANs with their own routers.
- each LAN (with a router) in a single site is assigned a **subnetid**.
- an **address mask** defines the boundary between a network address (**netid** and **subnetid**) and the **hostid**.

IP Routing

Address resolution protocol (**ARP**):

- to find a **physical address** of a target host.

ARP request

- broadcasting a frame to a local network, containing sender's **physical** and **IP addresses** and recipient's **IP address**.

ARP response

- reply using the same format, but now with the recipient's **physical address** filled in.
- a router replies when the recipient is not in the same LAN.

protocol type
operation
sender physical address
sender IP address
recipient physical address
recipient IP address

operation
1: ARP request
2: ARP response

IP Routing (2)

Router:

when a frame is received,

- extract a packet, and examine the destination's **IP address**.
- determines where to send next, using the **ARP**.
- build a new frame, with the next recipient's **physical address**, and send it.

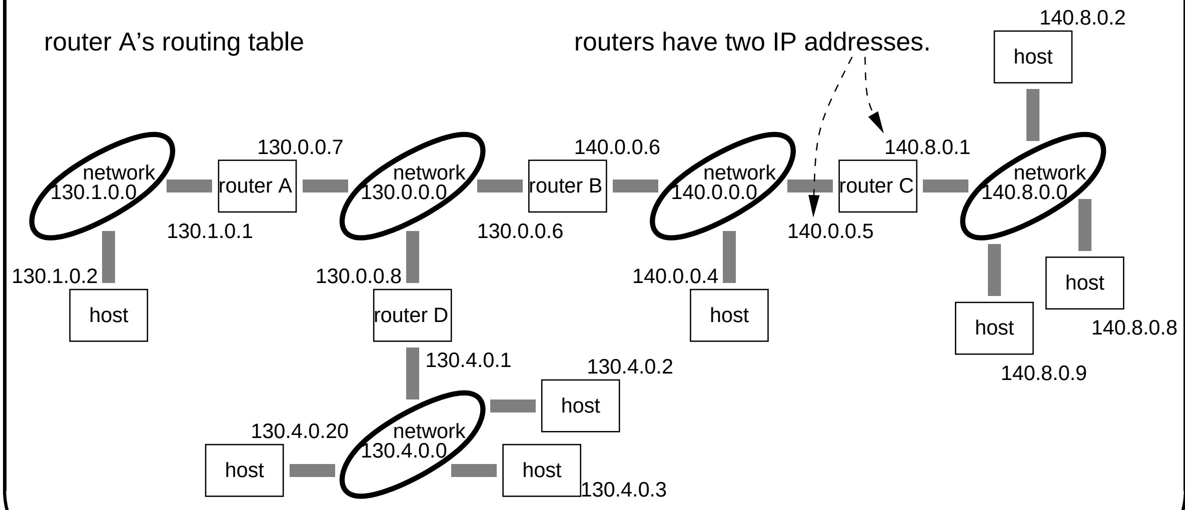
the packet then travels from router to router until it reaches one connected to the destination network.

IP Routing (3)

in order to reach hosts on network:	we take this route:
130.0.0.0	direct
130.1.0.0	direct
130.4.0.0	130.0.0.8
140.0.0.0	130.0.0.6
140.8.0.0	130.0.0.6

router A's routing table

- suppose router A receives a packet containing an IP address of '130.0.x.y', then the packet can be delivered to the destination directly.
- suppuse A receives a packet with a destination address '140.8.p.q', it should create a frame containing an IP address '130.0.0.6', that specifies router B via network '130.0.0.0'.



IP Routing (4)

IP routing algorithm:

1. when a router receives a packet, it determines the IP address of the destination network.
2. if the destination matches any of directly connected networks, the router sends the packet there.
3. else if a route is specified by a sending station, the packet is sent accordingly.
4. else if the destination appears in the router's routing table, the packet is sent according to the routing table data.
5. else if a **default router** exists, the router sends the packet there.
6. else the router declares a **routing error**.

ICMP

ICMP (**Internet control message protocol**) forms an integral part of whole IP implementation, and is used for network management.

Error reporting

- **destination unreachable**:

  the destination may be down, or may not exist).
- **time exceeded**.
- **parameter error**:

  packet's header parameter is not recognised.

Reach-ability testing

- **echo request** / **reply**:

  to find whether a particular destination is reachable.

ICMP (2)

Congestion control

- **source quench**:

  to request a host to reduce the number of packets.

Route exchange

- **redirect**:

  to inform a host to use the alternative route.

Performance measuring

- **time stamp request** / **reply**:

  to determine the delay in transmission between two hosts.

Subnet addressing

- **address mask request** / **reply**:

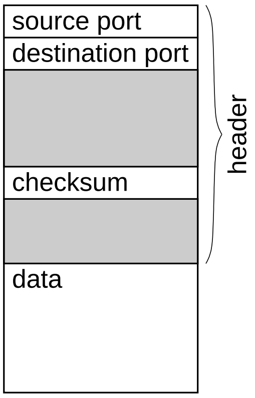
  to find the address mask associated with a subnet.

TCP

TCP (**transmission control protocol**) may provide the perception of a **connection-oriented** (*i.e.*, reliable) service by interfacing between the user and network protocols.

- provides the handshaking by establishing, maintaining, and releasing connections.
- handles requests to deliver information to a destination reliably.
- receives data or requests from its user, stores it in a **TCP segment**, and gives it to the **IP**.

the maximum size of the receive buffer is typically 4098, 8192, or 16 384 bytes, thus larger size data must be fragmented.



TCP segment

TCP (2)

Connection establishment:

- a **three-way handshake** protocol is used.
- a TCP connection is fully duplex.

  (*i.e.*, data can be transmitted in both direction simultaneously.)
- in a client-server applications, a client always initiates a TCP connection.

Data transfer:

- **error control**:
  - successful transmission of each TCP segment is acknowledged, using <`ack`> signal, by the recipient.
  - a failed TCP segment (*e.g.*, corrupted data, substantial delay caused by congestion) is retransmitted.

TCP (3)

Data transfer: (continued)

- **flow control**:

  use of **buffer allocation** scheme to ensure the sufficient buffer space at the recipient side.

- **congestion control**:

  initially TCP does not know the congestion status of the network, thus

  - starts transmission with the small number of **TCP segments**, while monitoring the return of `<ack>` signals.
  - gradually increases the number of **TCP segments** per unit time.

Connection termination:

- one host initiates the closure of connection (**active close**), and the other side follows (**passive close**).

UDP

UDP (**user datagram protocol**) is a **connectionless** (*i.e.*, best effort) service.

- little more than interfacing between the user and the lower layer.
- no handshake is established.
- data or requests is stored in a **UDP segment** — which has limited abilities — then passed to the **IP** for delivery.
- no mechanisms for acknowledging error, nor completion of transmission.
- used for realtime applications (*e.g.*, internet telephony with audio / video data).

Client-Server Model

Client: any program that makes requests to a server.

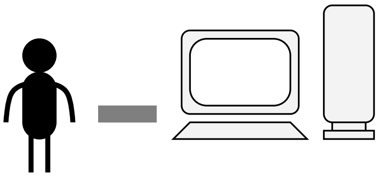
Server: any application that provides a service to network users —

- **file server**:
  - stores, manages, and provides access to files.
  - allows users to share files kept somewhere in a network.
- **communication server**:
  - establishes connection between a user and a network host computer once such request is made.

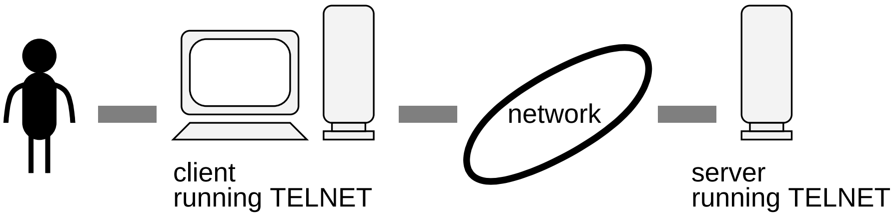
Typically (but not always) a **client** and a **server** run on different machines.

TELNET

Login to a local machine:



Remote login:

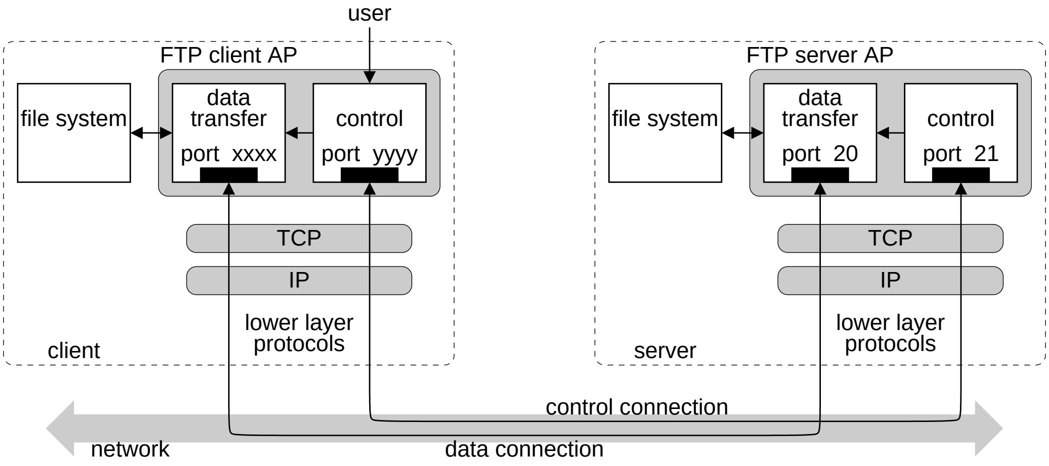


- a local and a remote computer run the TCP/IP to establish a connection.
- once connected, TELNET works in the background, thus transparent to the user and appears much like a local login.

FTP

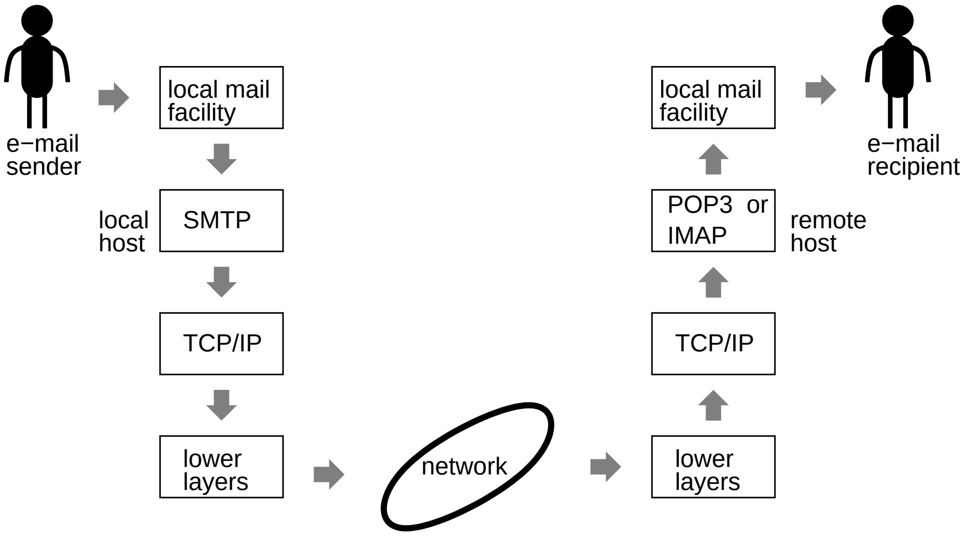
File transfer protocol (**FTP**):

- initiated by a user to transfer various files from one place to another.
- two TCP connections (**control** / **data connections**).
- port numbers — server side: 20/21 (fixed), client side: one-time use.



Electronic Mail

SMTP (**simple mail transfer protocol**) —



- SMTP (sender’s side) first call TCP to establish a connection with the remote site.
- when the connection is made, both sides exchange packets and eventually the mail is delivered.

Electronic Mail (2)

POP3 (**post office protocol**, version 3) —

- provide a simple mailbox manipulation: copy a message from the mailbox (on server) to the host, then delete from the mailbox.

  (*i.e.*, once accessed, a message is deleted permanently from the server.)

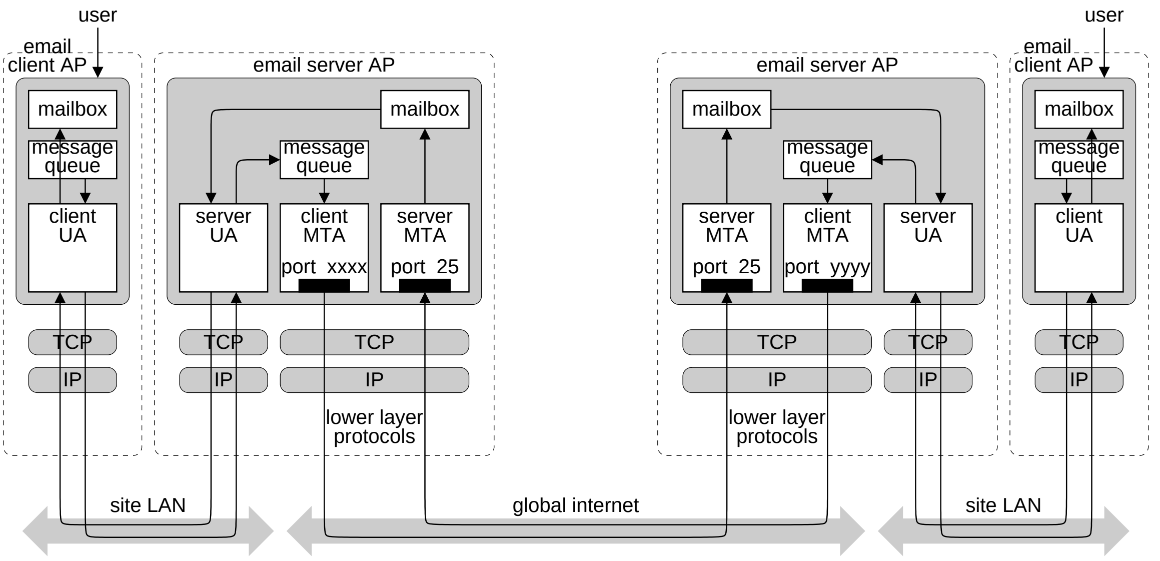
IMAP (**internet message access protocol**) —

- allow accesses to the mailbox from multiple places.
- more advanced, flexible, secure.
- treat a message as a collection of **MIME** body parts.
- **MIME** (multipurpose internet mail extentions):
  - a header plus structured message body;
  - defined encoding rule for non-**ascii** text, audio and visual data, *etc.*

Electronic Mail (3)

Email server / client:

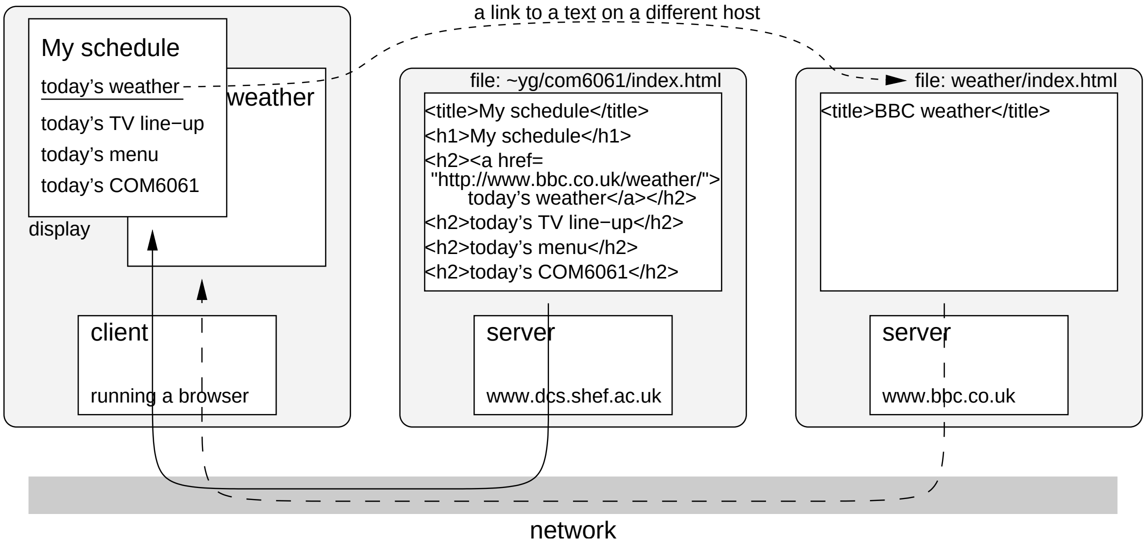
- port numbers — server MTA: 25 (fixed), client MTA: one-time use.



AP: application process UA: user agent MTA: message transfer agent

World Wide Web

Web model:



a link to a text on a different host

My schedule

today's weather

today's TV line−up

today's menu

today's COM6061

weather

display

client

running a browser

file: ~yg/com6061/index.html

```
<title>My schedule</title>
<h1>My schedule</h1>
<h2><a href=
 "http://www.bbc.co.uk/weather/">
       today's weather</a></h2>
<h2>today's TV line−up</h2>
<h2>today's menu</h2>
<h2>today's COM6061</h2>
```

server

www.dcs.shef.ac.uk

file: weather/index.html

```
<title>BBC weather</title>
```

server

www.bbc.co.uk

network

World Wide Web (2)

A view from a web **client** —

- the **web** consists of a vast worldwide collection of documents, usually just called '**pages**' for short.
  - each page may contain links (pointers) to the other related pages, anywhere in the world.
  - users can follow a link (by clicking on it), which then takes them to the page pointed to.
- pages are viewed with a program called a '**browser**' (*e.g.*, **Netscape**).
  - **HTML** (**hypertext markup language**) describes how documents are properly formatted.
  - the browser fetches the page (**HTML** document), interprets text and formatting commands that it contains, and display on the screen.

World Wide Web (3)

URL (**uniform resource locator**): http://www.dcs.shef.ac.uk/teaching/

- assigned to each page, uniquely in worldwide.
- consisting of
  - protocol (*e.g.*, **http**)
  - name of the server where the page is located (*e.g.*, www.dcs.shef.ac.uk)
  - file name (*e.g.*, teaching/index.html).

HTTP (**hypertext transfer protocol**):

- typically used with the **TCP** for transport connection.
- consisting of a set of requests from browsers to servers, and a set of responses going back to the other direction.

World Wide Web (4)

CGI script (**common gateway interface script**):

- filled-in form integrated into a client's web page.
- CGI script at the server processes an input from the client.

Java applets:

- separate programmes called from an HTML page, downloaded from a web server, and run on the client's machine.

Java script:

- a programme code included in an HTML page.

Plug-ins:

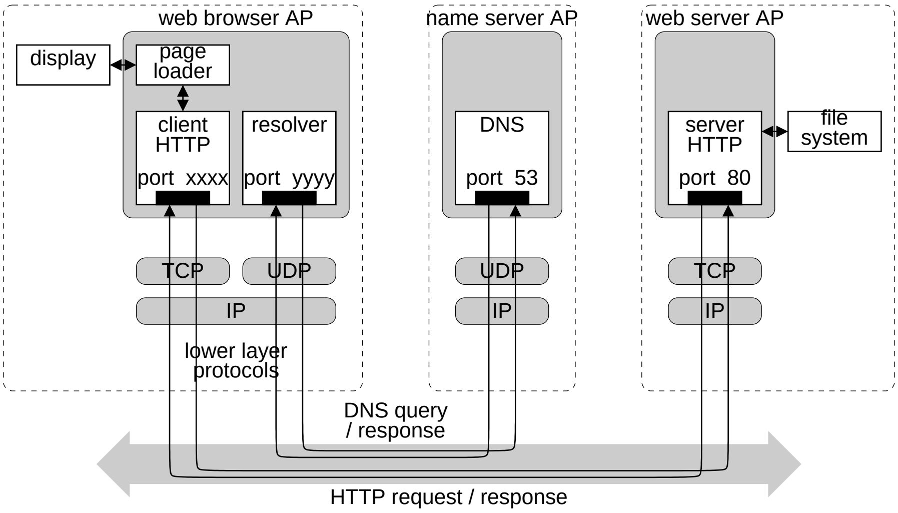
- process multimedia applications such as audio and video.

World Wide Web (5)

Web server / client:

- port numbers — server HTTP: 80 (fixed), client HTTP: one-time use.

(port number for DNS server: 53)



World Wide Web (6)

What may happen if someone clicks a link on the browser —

1. the browser will
   (a) determine the **URL** of the link.
   (b) ask **DNS** for the **IP address**.
   (c) determine the protocol, then make a **TCP connection** to the web server specified by the **IP address**.
   (d) send commands to get the specified page.
2. the server will send the requested page.
3. the browser will
   (a) release the **TCP connection**.
   (b) display the formatted page on the screen.