RW354
Principles of Computer Networking

A.E. Krzesinski and B.A. Bagula
Department of Computer Science
University of Stellenbosch

Last updated: 16 July 2004

The material presented in these slides is used with permission from

- *Larry L. Peterson and Bruce S. Davie. Computer Networks: A Systems Approach (Third Edition). Morgan Kaufmann Publishers. ISBN 1-55860-577-0.*
- *William Stallings. Data and Computer Communications (Sixth Edition). Prentice-Hall Inc. ISBN 0-13-571274-2.*
- *Andrew S. Tannenbaum. Computer Networks (Fourth Edition). Prentice Hall Inc. ISBN 0-13-349945-6.*

Expectations

This course IS about

- *principles and concepts*
- *general-purpose computer networks*
- *the Internet perspective*
- *network software*
- *designing and building a system.*

This course IS NOT about

- *a survey of existing protocol standards*
- *specialized networks (CATV, telephone, . . .)*
- *the OSI perspective*
- *network hardware (we do survey)*
- *network performance using queuing theory models.*

Perspective

The expectations that you have of a network depends upon your persepctive

- *Network users: network services that user applications need e.g. a guarantee that each message that the application sends will be delivered without error within a certain amount of time.*
- *Network designers: cost-effective design e.g. network resources are efficiently utilized and fairly allocated to different users.*
- *Network providers: a system that is easy to administer and manage e.g. faults can easily be isolated and it is easy to charge for usage.*

Characterizing Networks

Communication networks are divided into 2 basic types

- *connection oriented: circuit switched*
- *connectionless: packet switched.*

Packet switched networks fall into 3 classes

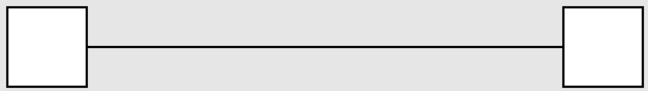
- *WAN: a national/international network*
- *MAN: a network connecting several LANs*
- *LAN: a network connecting computers in a building or a campus.*

Some networks are deliberately kept small in size.

The Internet is designed to grow to an arbitrarily large size (scale).

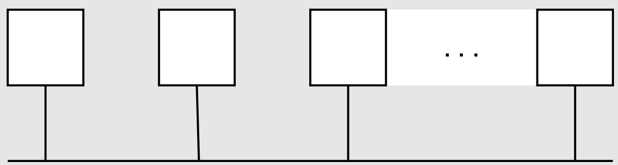
Connectivity

- *network building blocks*
  - *links: coaxial cable, optical fibre, satellite . . .*
  - *nodes: routers*
- *direct links*
  - *point-to-point: one link connects two nodes*



point-to-point network

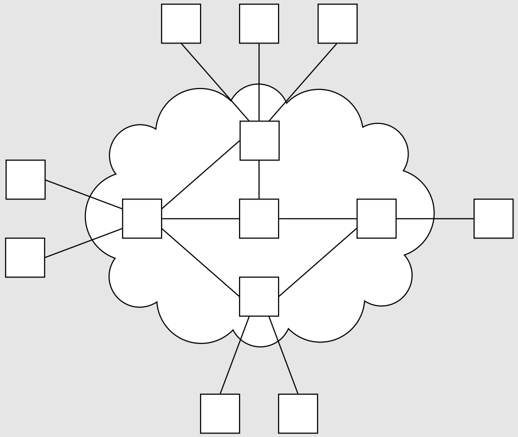
  - *multiple access: many nodes share a link*



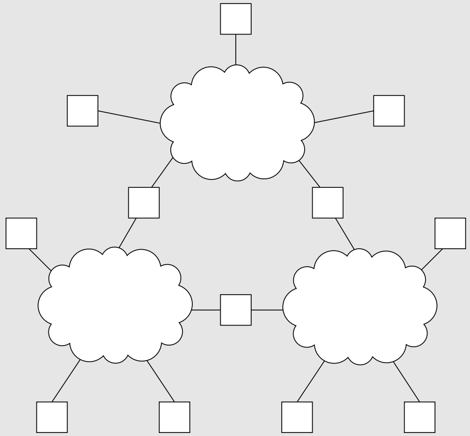
multiple access network

Indirect connectivity

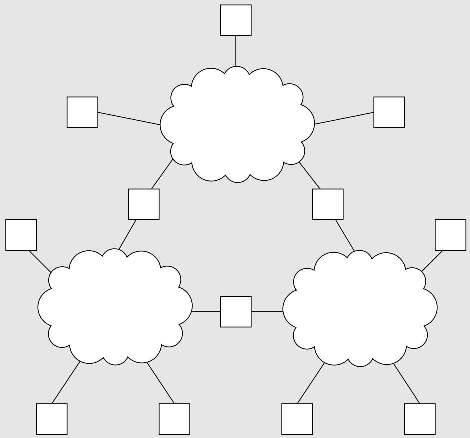
- *switched networks*



- *internetworks*



Indirect connectivity



- *nodes (switches) inside the cloud store & forward packets*
- *nodes (hosts) outside the cloud support users & applications*
- *routers (gateways) are connected to two or more networks.*

Switching strategies

A network can be defined recursively as

- *two or more nodes connected by a physical link, or*
- *two or more networks connected by one or more nodes.*

Networks use two switching methods

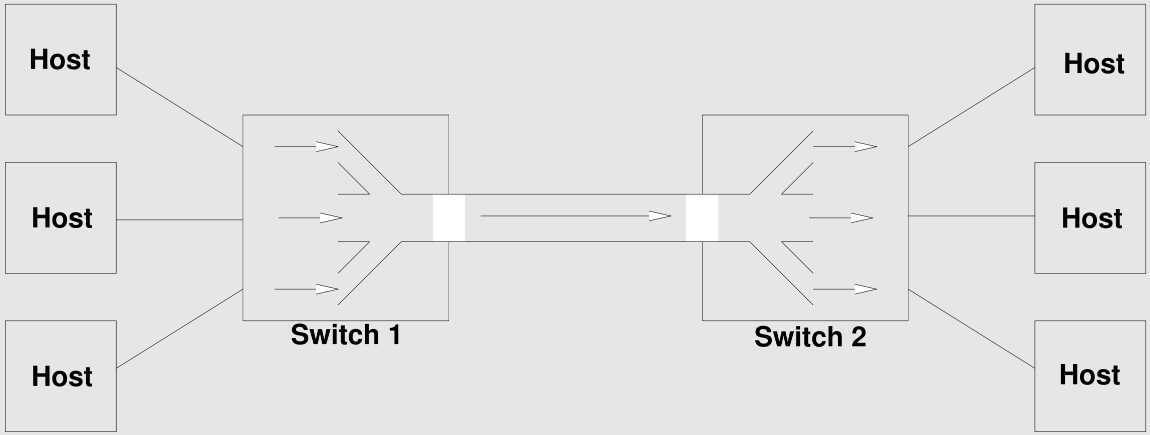
- *circuit switching: dedicated circuits are used to send/receive a bit stream*
- *packet switching: store-and-forward is used to send/receive messages (packets).*

Addressing & routing

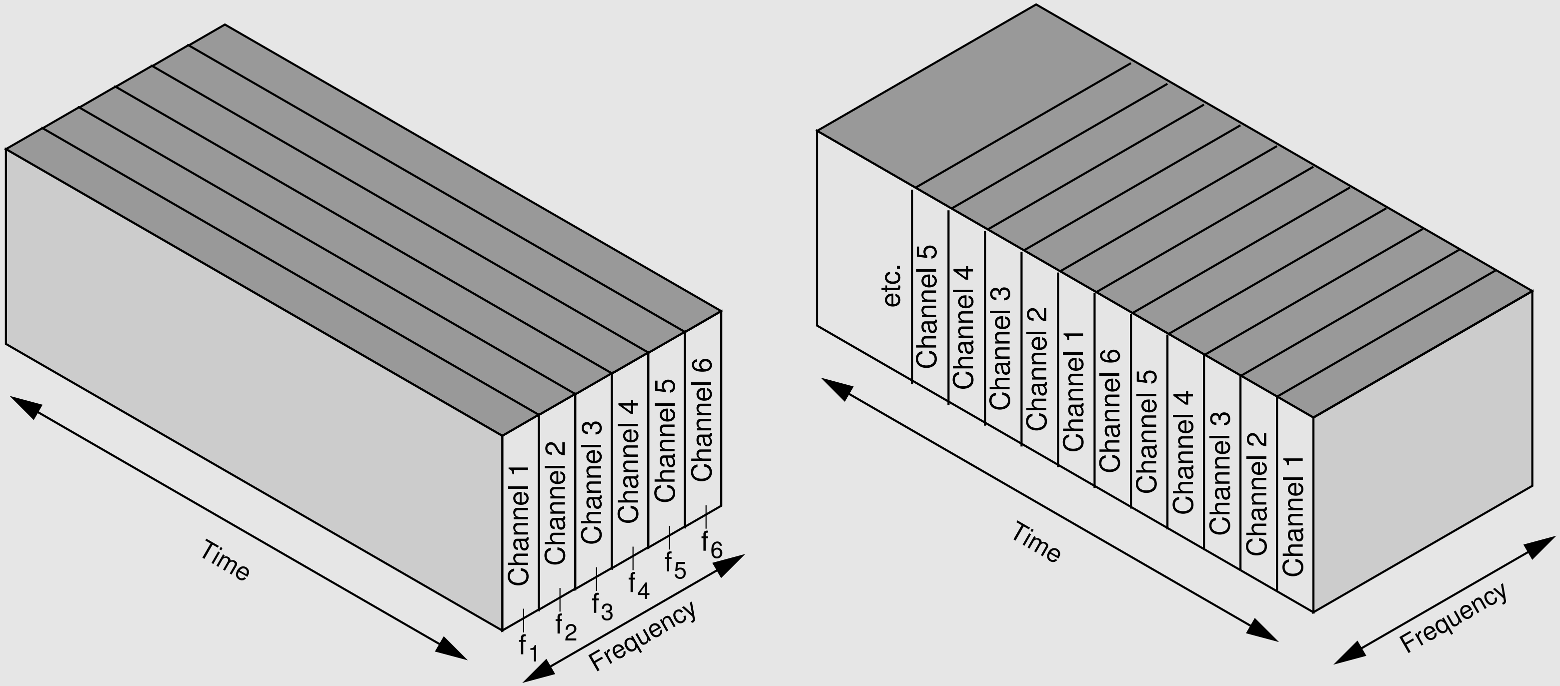
- *an address is a byte string that identifies a node; usually unique*
- *routing is the process of determining how to forward a message towards the destination node based on its address*
- *there are several types of addresses*
  - *unicast: node-specific*
  - *broadcast: all the nodes in the network*
  - *multicast: some subset of the nodes in the network.*

Cost-effective resource sharing

Networks must share (multiplex) network resources (nodes & links) among multiple users.



Common multiplexing strategies



Frequency-division multiplexing

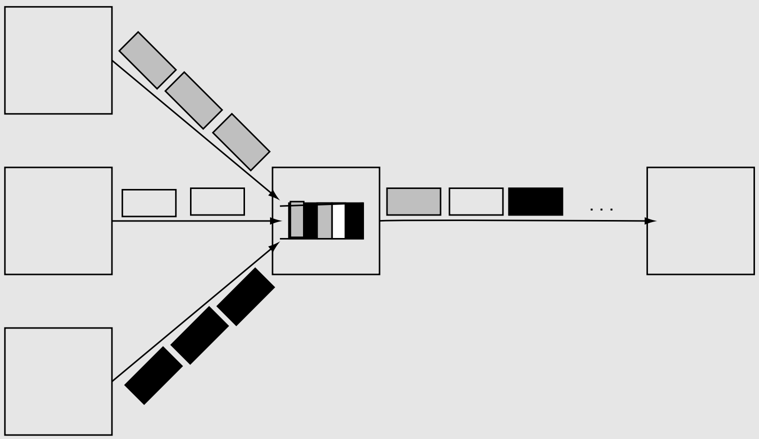
Time-division multiplexing

- *synchronous time-division multiplexing (STDM)*
- *frequency-division multiplexing (FDM)*

Both STDM and FDM are inefficient.

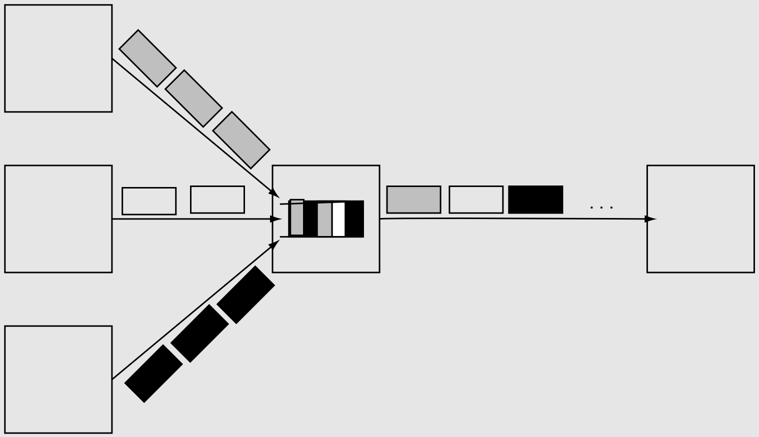
Statistical multiplexing

- TDM on demand rather than during a fixed time slot
- the link is rescheduled on a per-packet basis
- packets from different sources are interleaved on the link
- packets that contend for the link are buffered
- the packet queue is usually processed FIFO
- buffer overflow (dropped packets) is called congestion



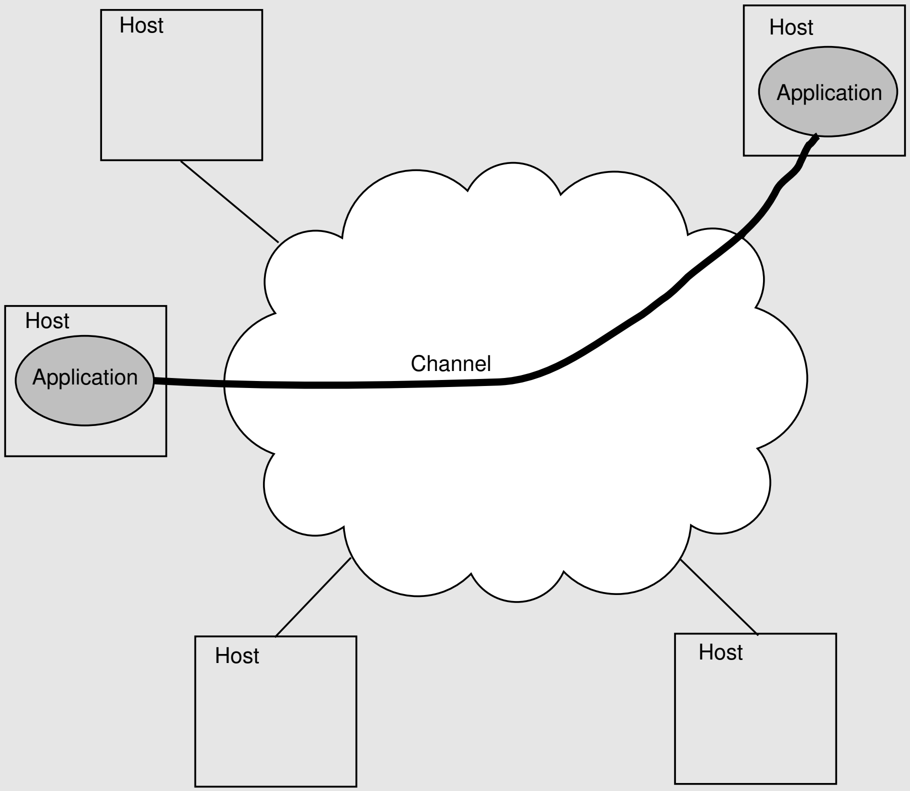
Statistical multiplexing

- *the packet queue is usually processed FIFO, but not necessarily*
  - *packets from different flows are serviced in a round robin fashion*
  - *certain flows receive a certain portion of the link bandwidth*
    - *Quality of Service*



Functionality

Application programs running on hosts connected to the network must be able to communicate in a meaningful way.



The network provides common process-to-process channels. Each channel provides a set of communication services.

Process-to-process channels

What functionality should the channels provide? Guaranteed/best effort delivery? Delivery in/out of order? Privacy? Constant/variable packet delivery rate?

- *request/reply: for file access & digital libraries*
- *message stream: for video applications*
  - *video: sequence of frames*
  - *resolution: 1/4 TV-size image = 352×240 pixels*
  - *24-bit color: frame = $352 \times 240 \times 24 / 8$ = 247.5KB*
  - *frame rate: 30 fps = 7500KBps = 60Mbps*
  - *video on-demand vs video-conferencing*

Network faults

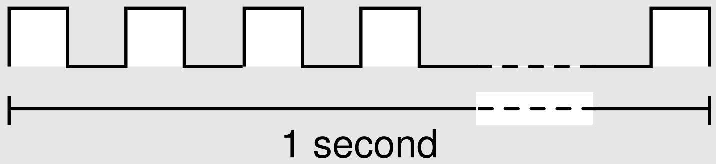
What can go wrong in the network?

- *bit errors, burst errors: rare, error correction*
- *packet-level errors: usually caused by congestion*
- *link & node failures*
- *messages are delayed*
- *messages are delivered out-of-order*
- *third parties eavesdrop.*

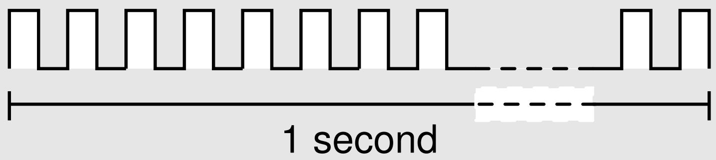
The key problem is to fill in the gap between what applications expect & what the underlying technology provides.

Performance: bandwidth (throughput)

- *the amount of data that can be transmitted per time unit, for example 10Mbps*
- *link versus end-to-end performance*
- *notation: KB = 2^{10} bytes, Mbps = 10^6 bits per second*
- *bandwidth is related to "bit width"*



1Mbps
(each bit 1 microseconds wide)



2 Mbps
(each bit 0.5 microseconds wide)

Performance: latency (delay)

- *latency: the time it takes to send a message from point A to point B*
- *the round-trip time (RTT): from A to B & back*
- *the components of latency*
  - *latency = propagation + transmission + queue*
  - *propagation = distance /* C
  - *transmit = size / bandwidth*
- *the speed of light* C
  - 3.0×10^8 *meters/second in a vacuum*
  - 2.3×10^8 *meters/second in a cable*
  - 2.0×10^8 *meters/second in a fiber*

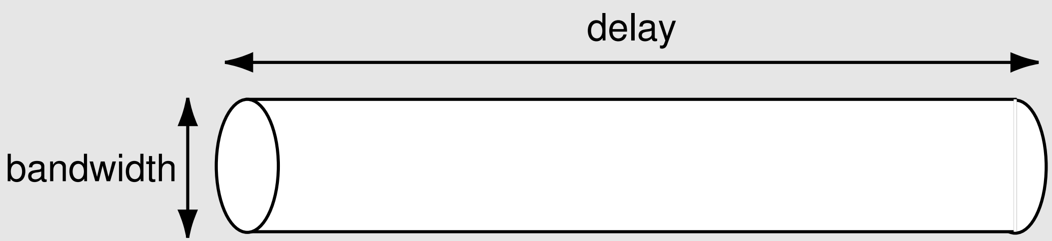
Performance: latency (delay)

Notes

- *no queuing delays in direct link*
- *bandwidth is not relevant for the performance of small transfers*
- *bandwidth is relevant for the performance of large transfers*
- *process-to-process latency includes software overhead*
- *software overhead can dominate when distance is small*

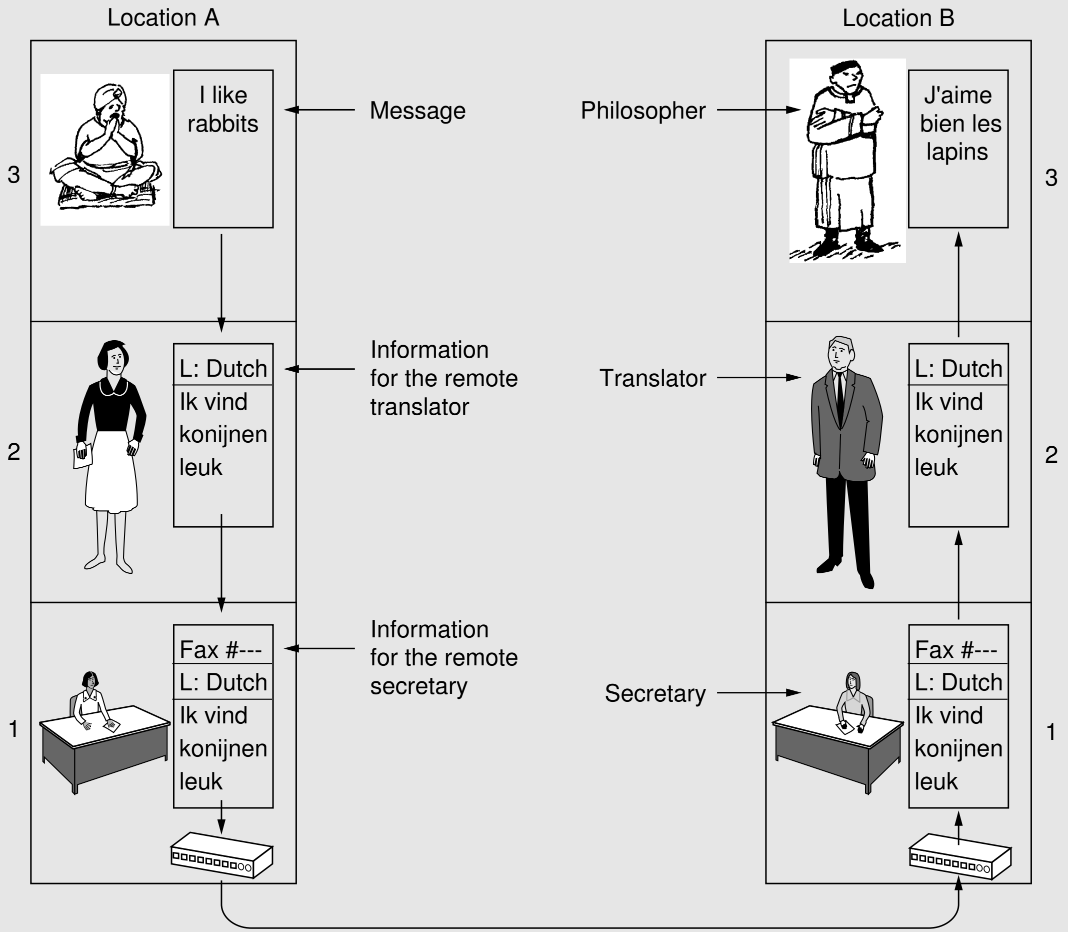
Delay-bandwidth product

- *the relative importance of bandwidth & latency*
  - *small message (e.g. 1 byte): 1ms vs 100ms RTT dominates 1Mbps vs 100Mbps bandwidth*
  - *large message (e.g. 25 MB): 1Mbps vs 100Mbps bandwidth dominates 1ms vs 100ms RTT*
- *delay-bandwidth product: 100ms delay & 45Mbps bandwidth = 560 KB of data in the pipe*



- *application needs*
  - *bandwidth requirements: burst size vs peak rate*
  - *jitter: variance in latency (inter-packet gap)*

Network architecture: Layering



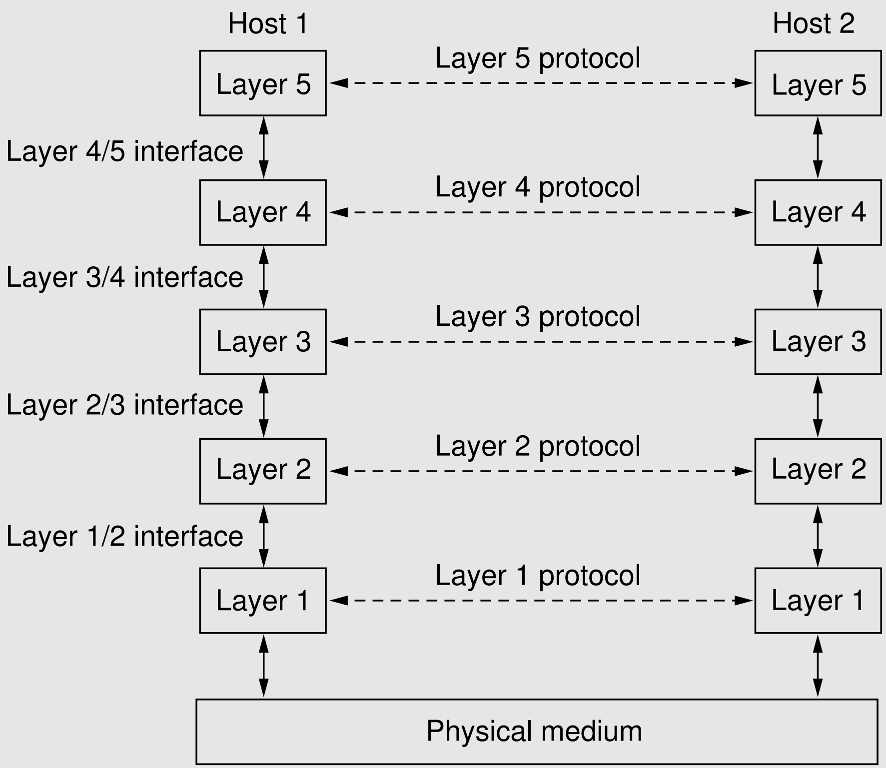
A ***protocol stack*** *is a list of protocols used by a system, one protocol per layer.*

Network architecture: Layering

- *abstractions are used to hide complexity*
- *abstraction naturally leads to layering*

- *alternative abstractions can be present at each layer*

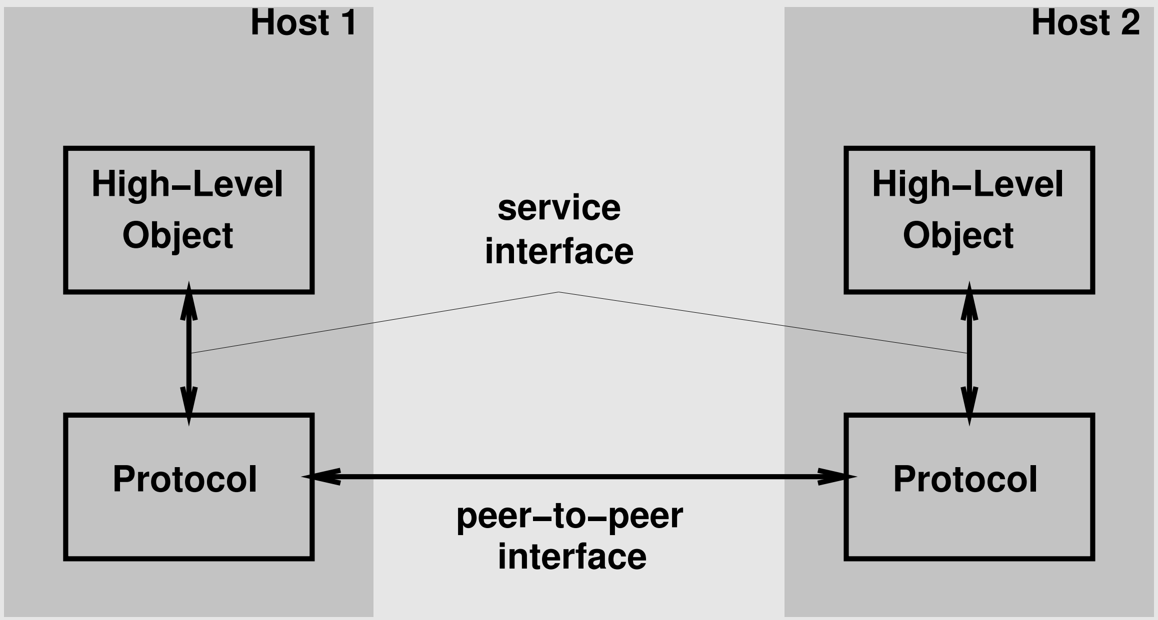
Network architecture: Layering



- *the corresponding layers on different machines are called peers*
- *an interface is present between each pair of adjacent layers.*

Network architecture: Protocols

- *building blocks of a network architecture*
- *each protocol object has two different interfaces*
  - *the service interface: defines operations on this protocol*
  - *the peer-to-peer interface: defines messages exchanged with peer.*



Network architecture: Protocols

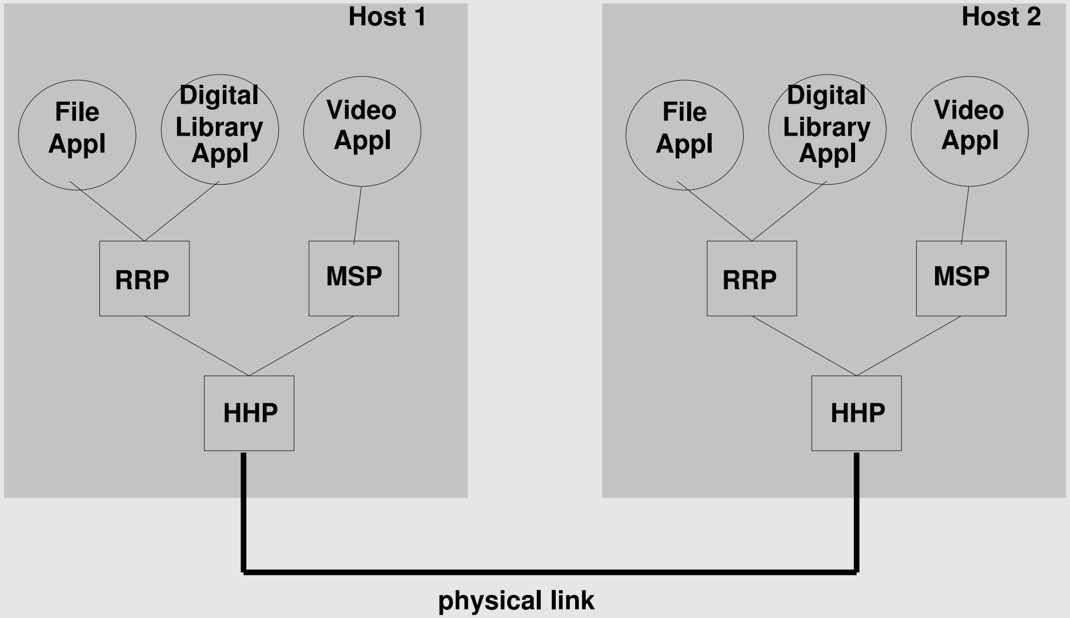
The term “protocol” is overloaded. It implies both

- *the specification of the peer-to-peer interface*
  - *textual, psuedo-code, state transition diagrams, pictures of packet formats*
- *the module that implements this interface.*

Network architecture: Protocols

A protocol graph denotes a collection of protocols & their dependencies

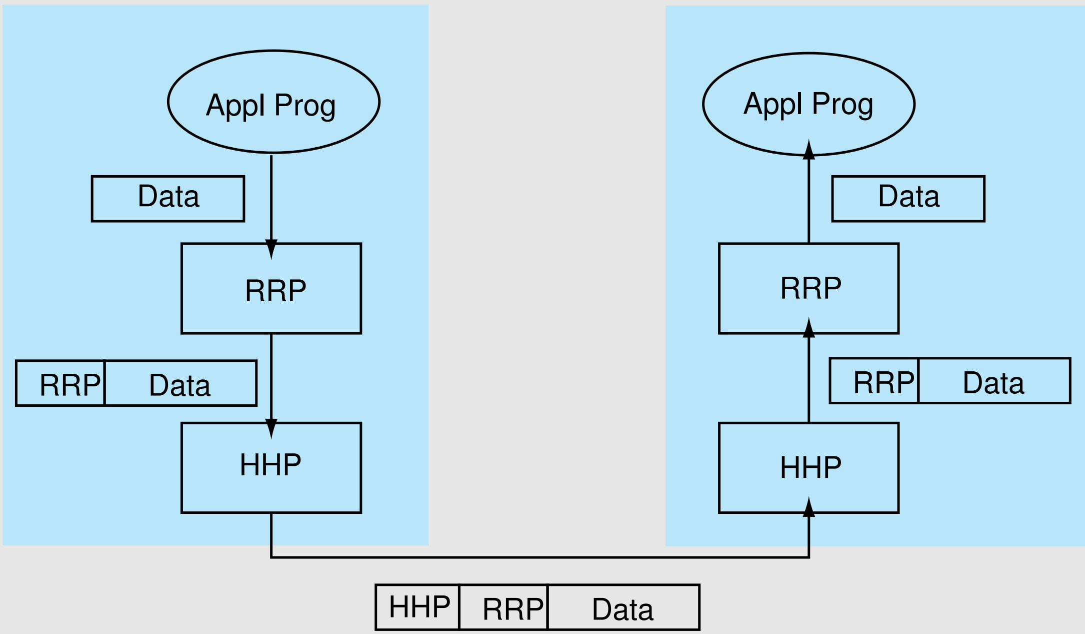
- *nodes correspond to protocols*
- *edges correspond to dependencies*
- *most peer-to-peer communication is indirect*
- *peer-to-peer communication is direct only at hardware level.*



Chapter 1.2

Network architecture: Protocols

- *multiplexing & demultiplexing: the demux key identifies the originating application*
- *encapsulation (header/body)*



The nodes in the network can inspect the HHP header – the payload is not inspected.

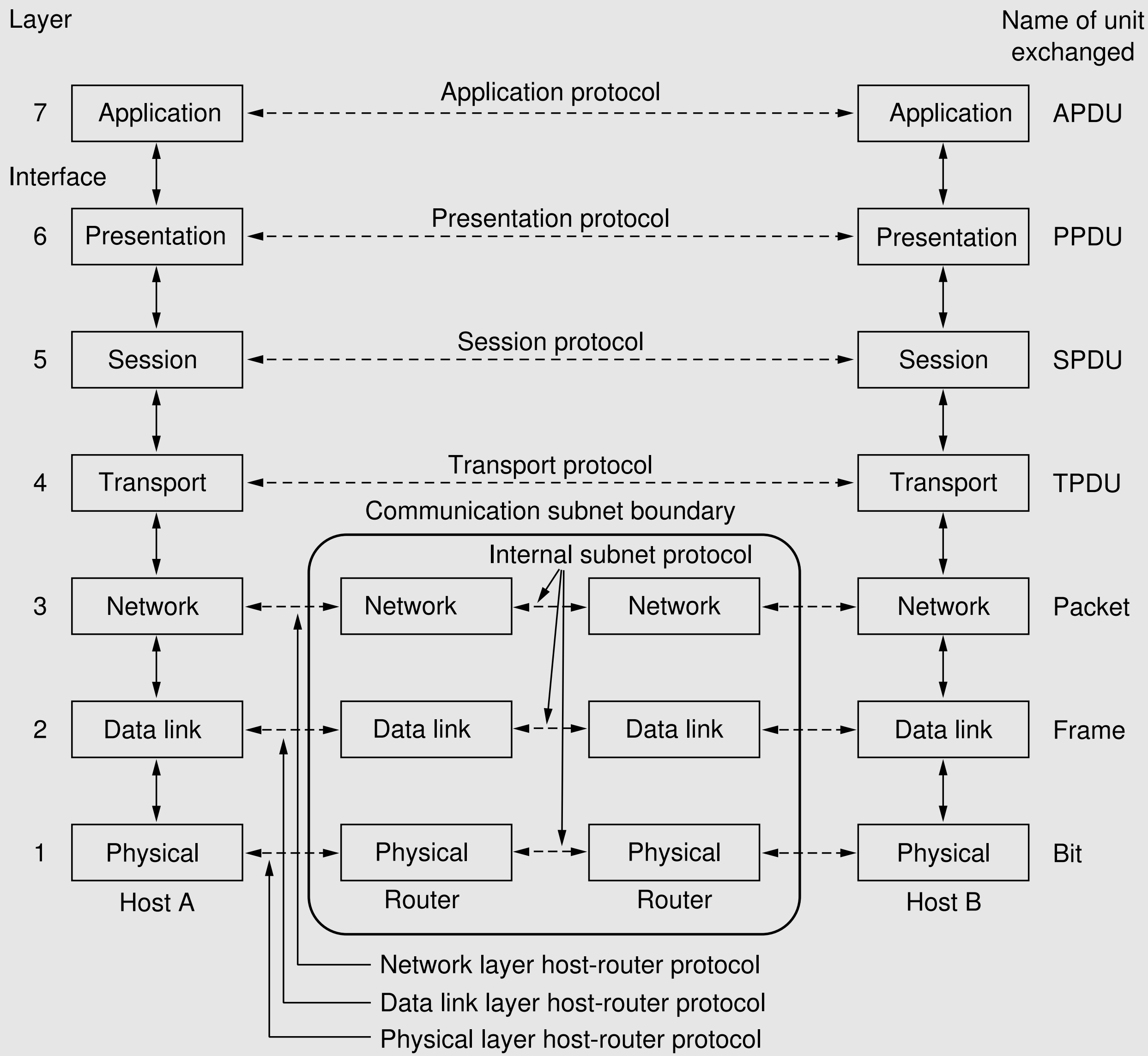
Standard architecture

Not the first network architecture

- *International Standards Organization (ISO)*
- *Open Systems Interconnect (OSI) Architecture*
- *International Telecommunications Union (ITU); formerly CCITT*
- *“X dot” series: X.25, X.400, X.500*

Standard architecture

The ISO OSI reference model

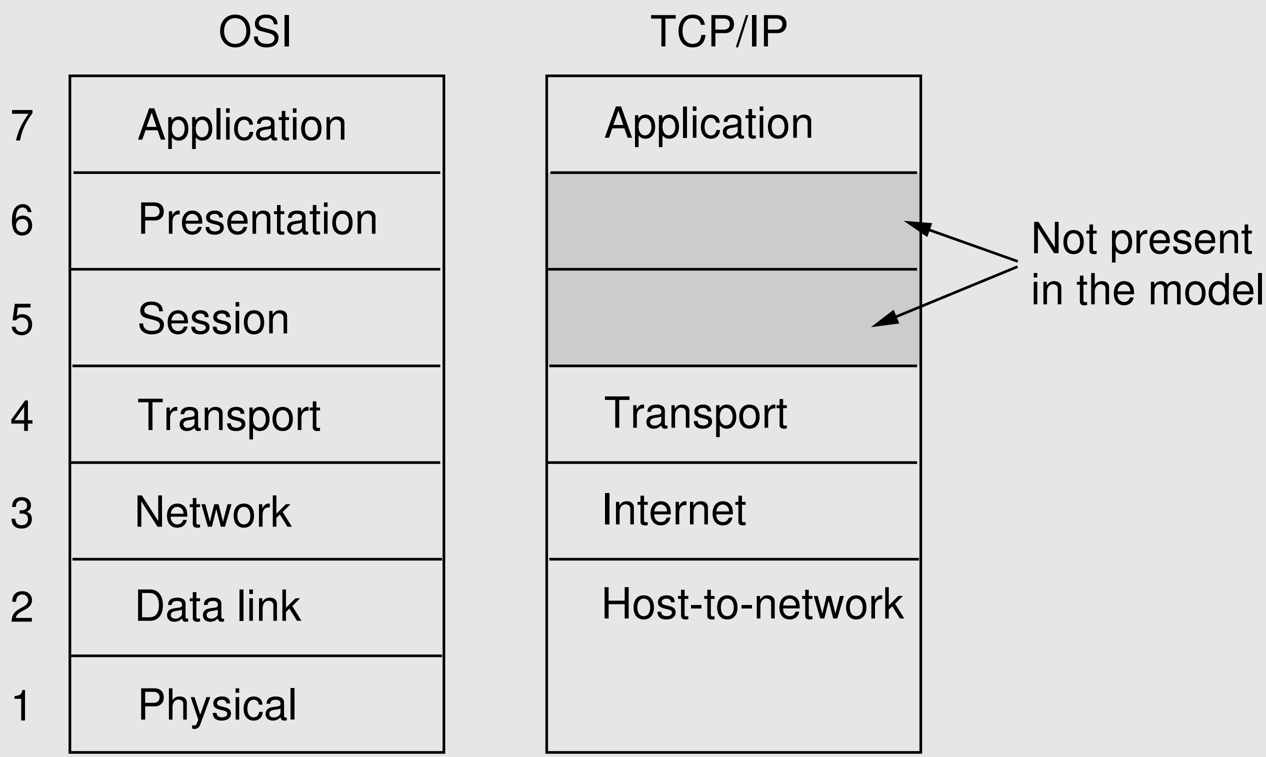


Standard architecture

- *the physical layer: the transmission of bits on the physical link*
- *the data link layer: correct transmission of a frame from one node to the next node*
- *network layer: correct transmission of a packet from source to destination*
- *the transport layer: correct transmission of a message from source to destination*
- *the session layer: manages different transport streams that are part of a single application*
- *the presentation layer: the format of the data exchanged between peers*
- *the application layer: the application*

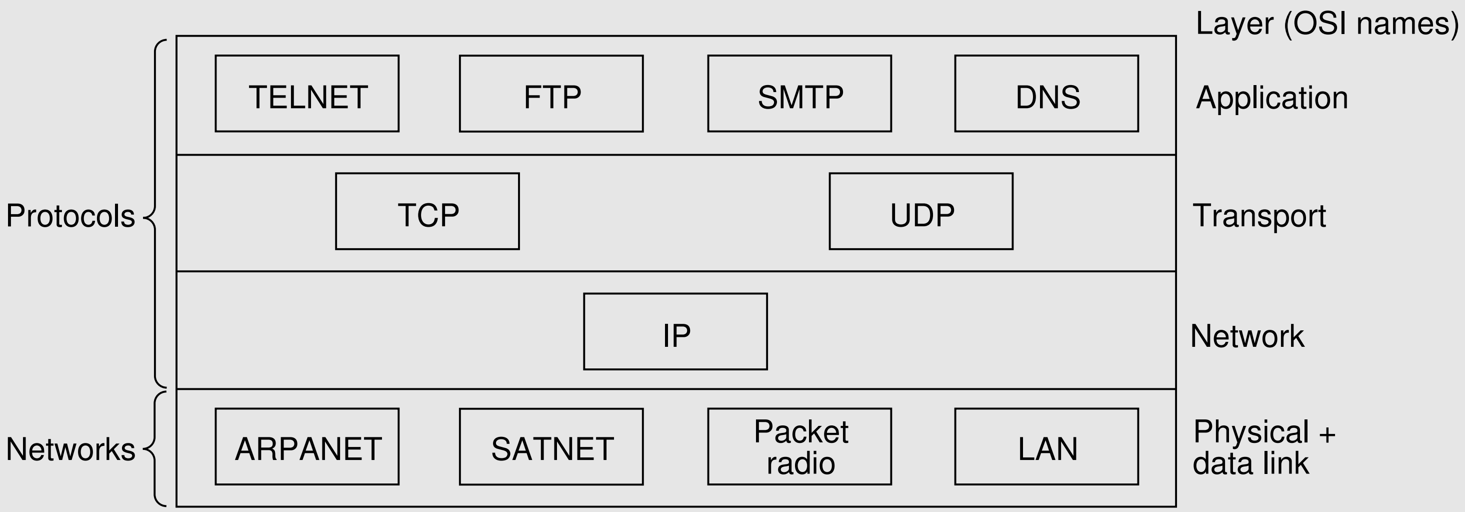
Internet architecture

The Internet has a 4-layer model



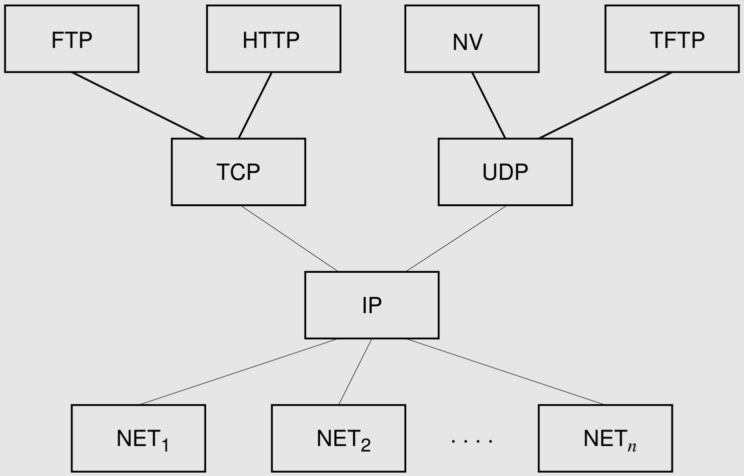
Internet architecture

The Internet has a 4-layer model



Internet architecture

The process of defining the Internet architecture is controlled by the Internet Engineering Task Force (IETF).



- *Application vs Application Protocol (FTP, HTTP)*
- *Features*
  - *does not imply strict layering*
  - *hourglass shape – IP is the focal point*
  - *design & implementation go hand-in-hand.*