4. Multimedia Communication

4.1 Network technology, as it is today

4.2 Quality of Service in networks

4.3 Multicast

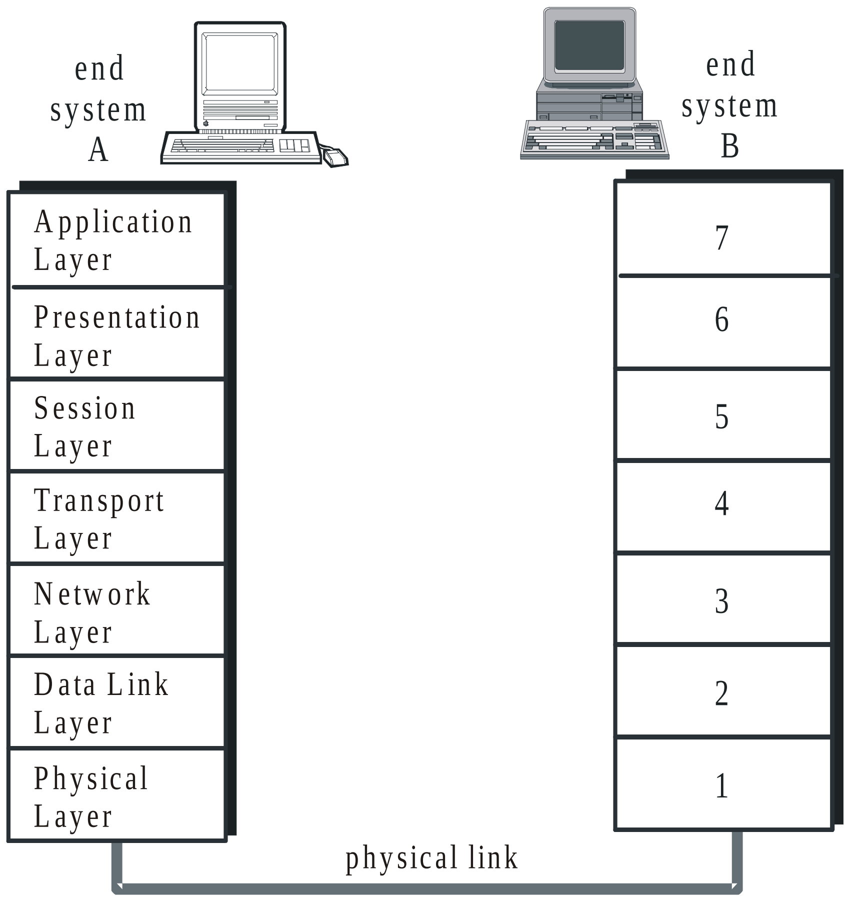
4.4 Media scaling and media filtering

4.1 Network Technology, As It Is Today

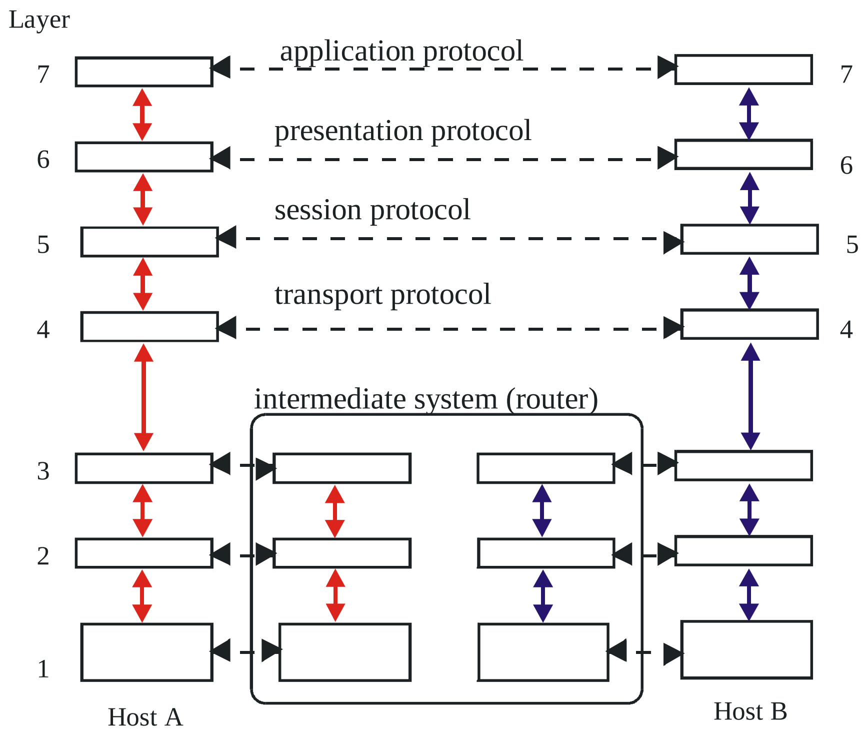
The computer networks we have today were designed with *data* communication in mind, for discrete pieces of data only. We will now look at the mechanisms in existing networks and see why the communication algorithms and protocols that were designed for discrete media are inappropriate for continuous media.

4.1.1. Protocol Architecture in Layers

All network protocol architectures we have today are based on the concept of **layers**.

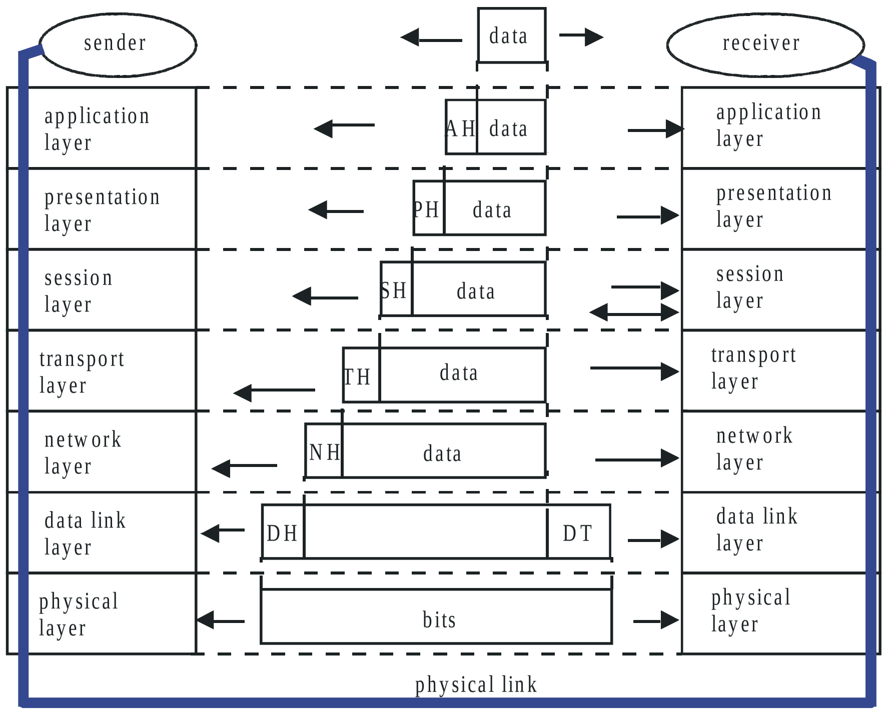


The ISO Reference Model for Open System Interconnection (OSI)



Packet Headers

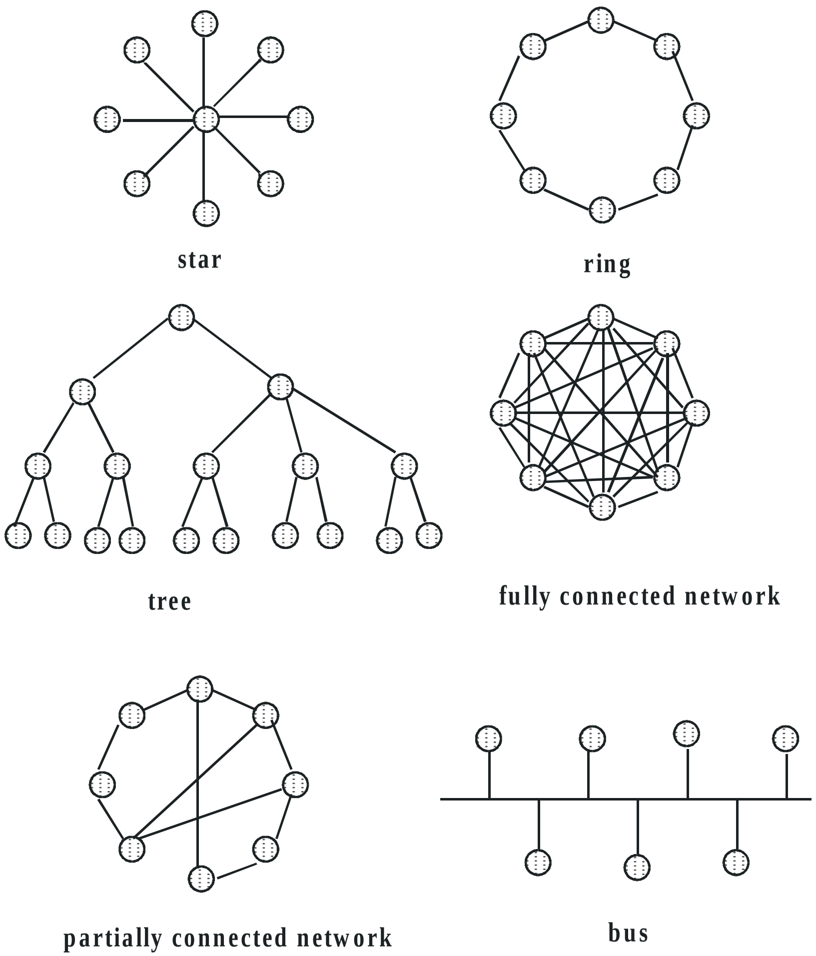
Typically, each layer adds a header, leading to a packet structure as shown below. Layer 2 also adds a trailer.



Layers of Different Network Architectures

Layer	ISO	Internet	SNA
7	Application	SMTP, FTP, telnet, http	End user
6	Presentation		NAU services
5	Session		Data flow control
4	Transport	TCP	Transmission control
3	Network	IP	Path control
2	Data link control	Data link control	Data link control
1	Physical	Physical	Physical

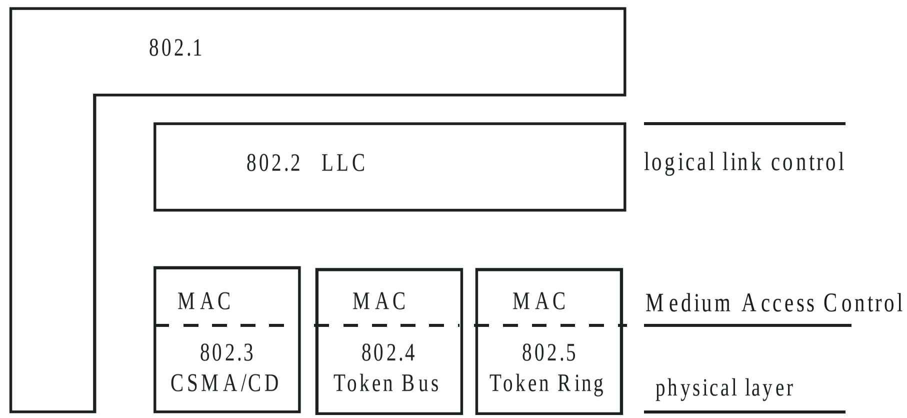
Network Topologies



4.1.2. Medium Access Control and Error Control in Layer 2

LAN characteristics

- Limited geographical area
- High transmission rates
- Low bit error rate and high availability
- Flexible reconfiguration
- Standardized by IEEE and ISO



Point-to-Point vs. Broadcast Networks

Point-to-Point Network

- Pairs of stations are connected by a physical link. The network has the topology of a graph of nodes and edges.
- Each message flows into one direction. Acknowledgements must be sent explicitly.
- Broadcast requires explicit replication of the message.

Broadcast Network

- Several stations share the physical medium.
- All stations hear all messages.
- If two stations send at the same time, the message is destroyed.
- The sender can hear his own message. If the sender hears exactly what he has sent, he can assume that all receivers have also heard the message (implicit acknowledgement).

Medium Access Control (MAC)

Problem

- Broadcast medium
- Independent stations

  => there will be send collisions.

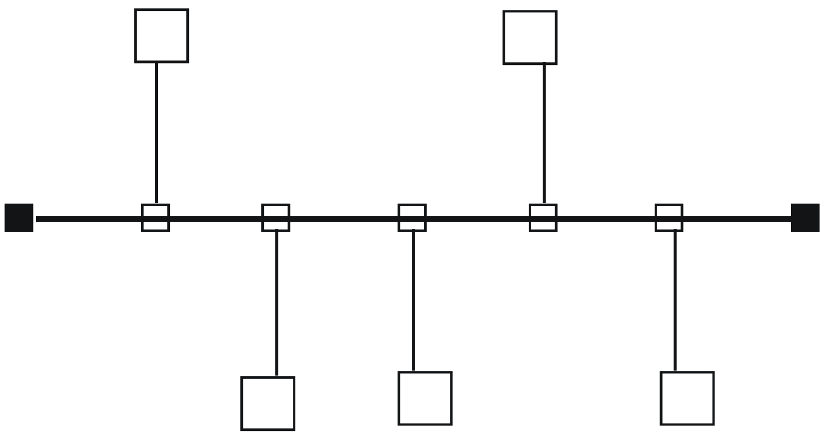
Solution

- Medium access control
- Two principles for MAC:
  1. Collision detection
     Let the collision happen, detect it, repeat the transmission.
  2. Collision avoidance
     Use a circulation token to explicitly control the access right to the medium.

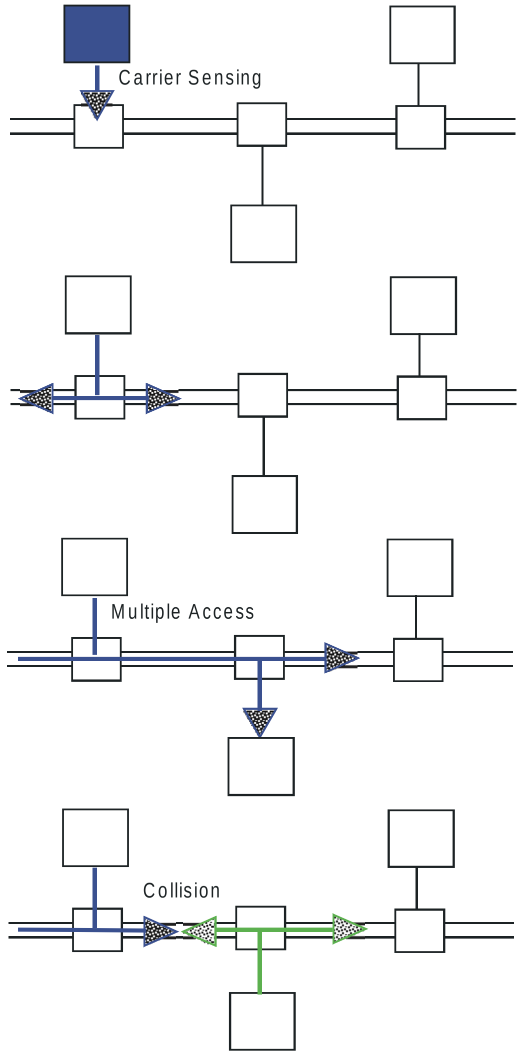
CSMA/CD

- CSMA/CD = **C**arrier **S**ense **M**ultiple **A**ccess with **C**ollision **D**etection
- Standardized as ISO IS 8802/3: MAC and physical layer for CSMA/CD

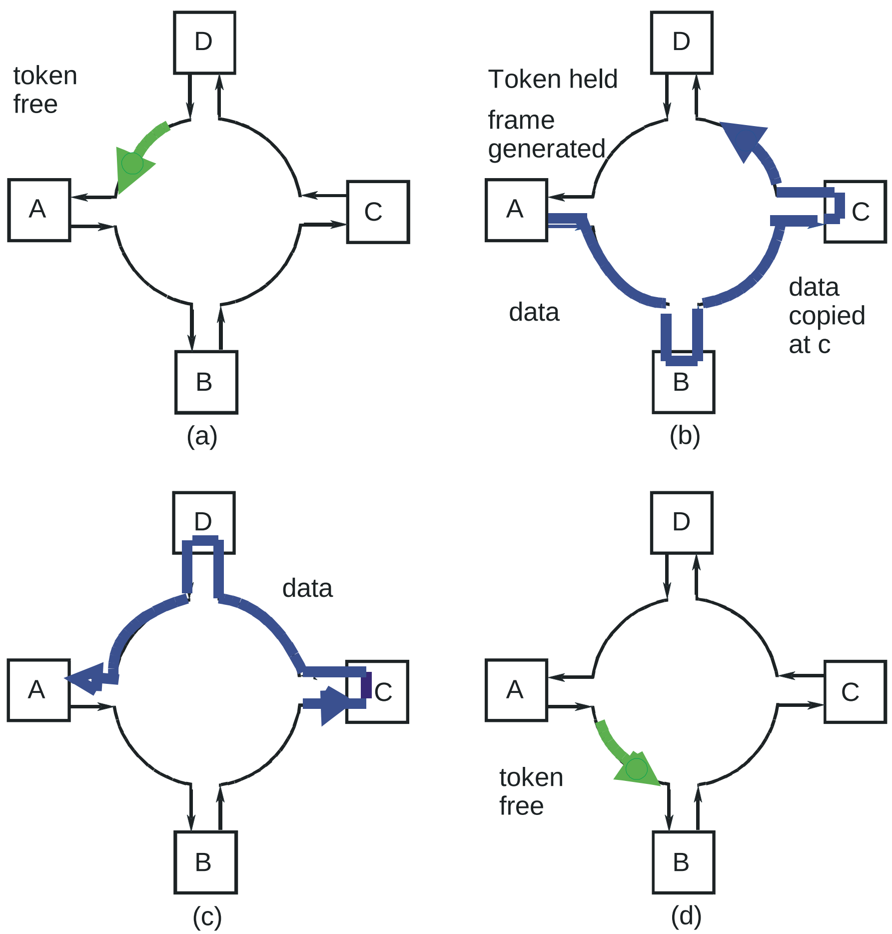
Topology: bus



CSMA/CD – the Protocol



Token-Ring Protocol



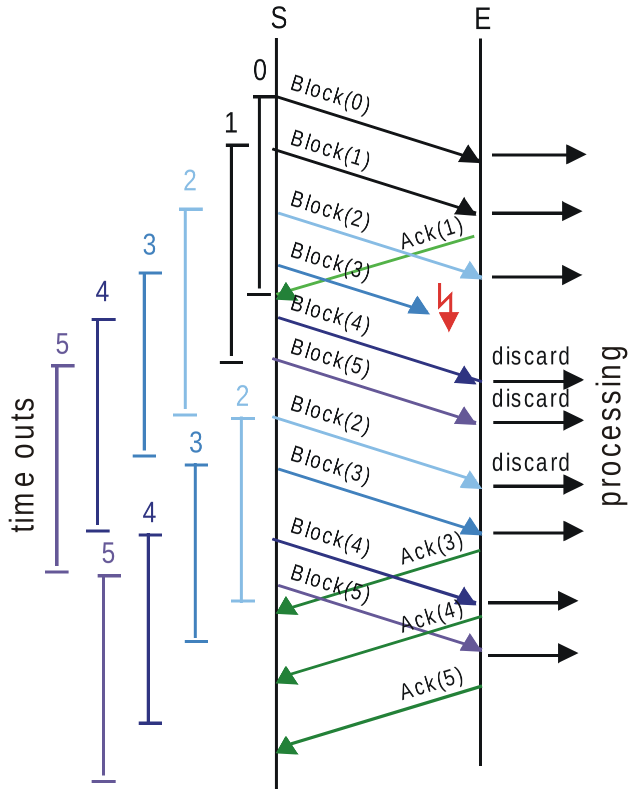
The Data Link Layer (Layer 2)

Tasks of the Data Link Layer (Layer 2)

- Conceal bit errors (errors on the transmission line) between neighboring nodes
- Provide flow control between neighboring nodes
- In LANs also: Implement the MAC protocol

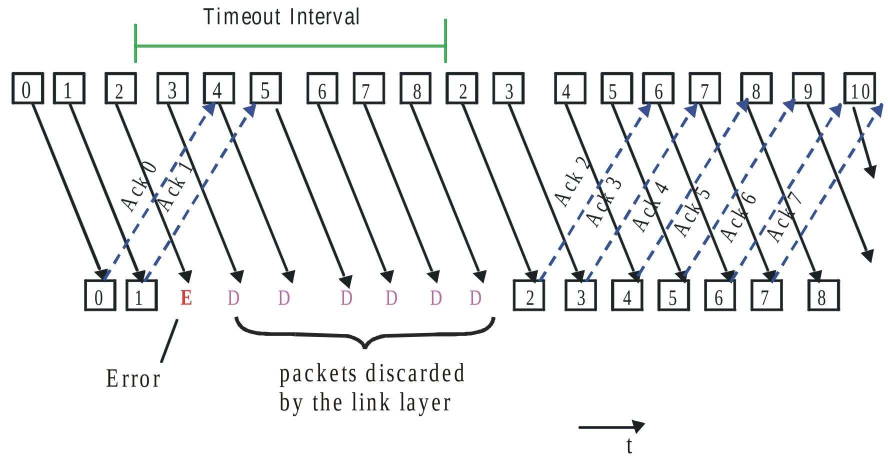
Sequence Numbers

Sequence numbers are used to uniquely identify data packets. The are used by the receiver to acknowledge specific packets. One acknowledgement (ACK) can acknowledge multiple data packets.



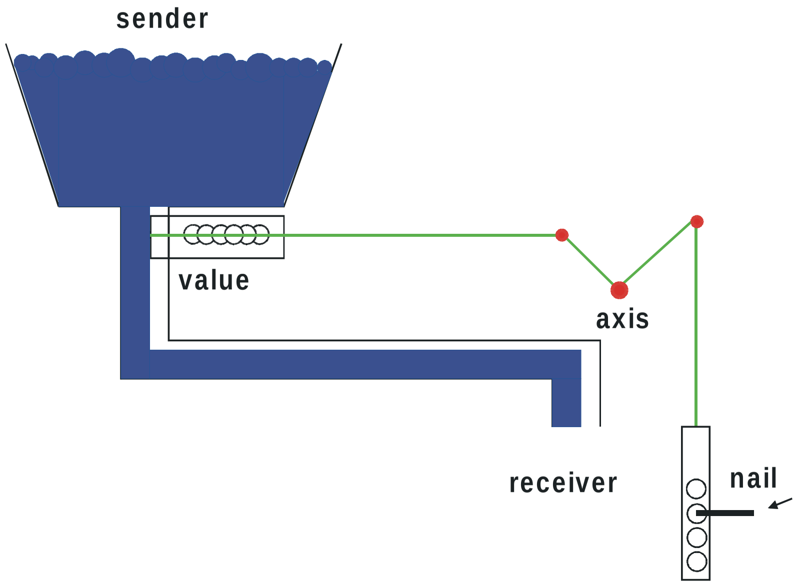
Retransmission in "Go-back-n“ mode

In case of a bit transmission error no ACK will be sent. When the sender‘s timer expires he will retransmit the missing packet and **all** packets sent later.



Principle of Flow Control

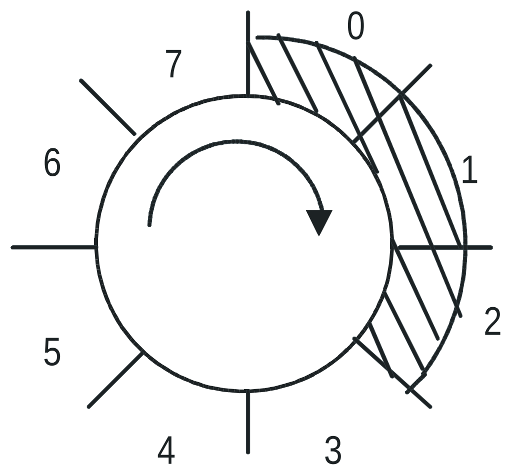
The **flow control mechanism** prevents a sender from overflowing a (slower) receiver.



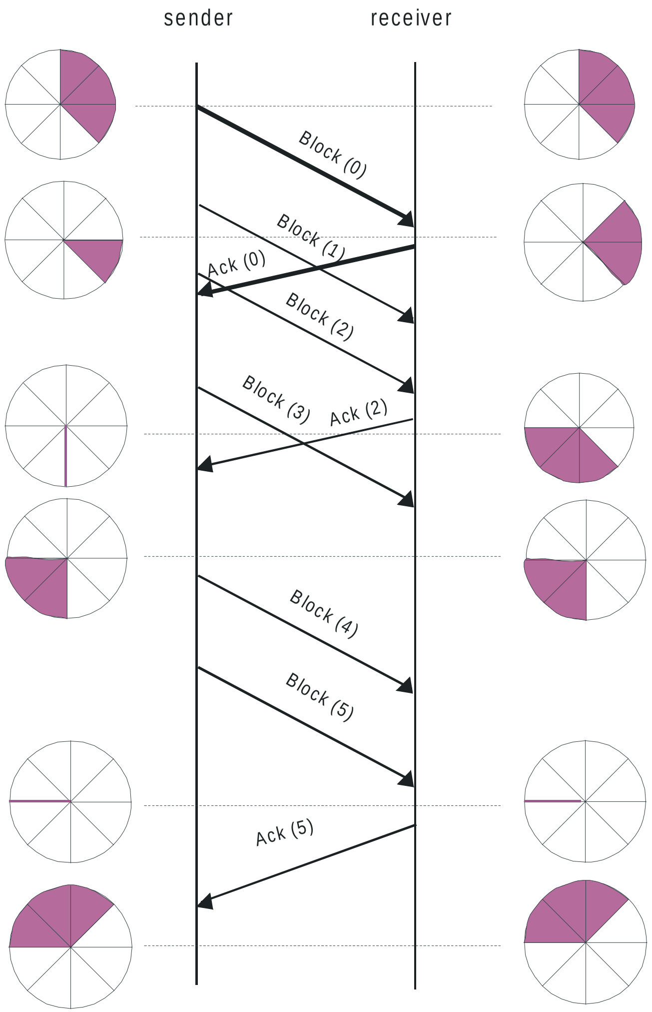
Sliding Window Flow Control

- After connection setup the sender has the right to send as many packets as the window size indicates.
- After that he must wait until he receives an ACK from the receiver. One ACK can confirm several packets.
- The receiver can send ACKs before the window is fully used up. The ACK policy is not standardized.

Example: window size = 3

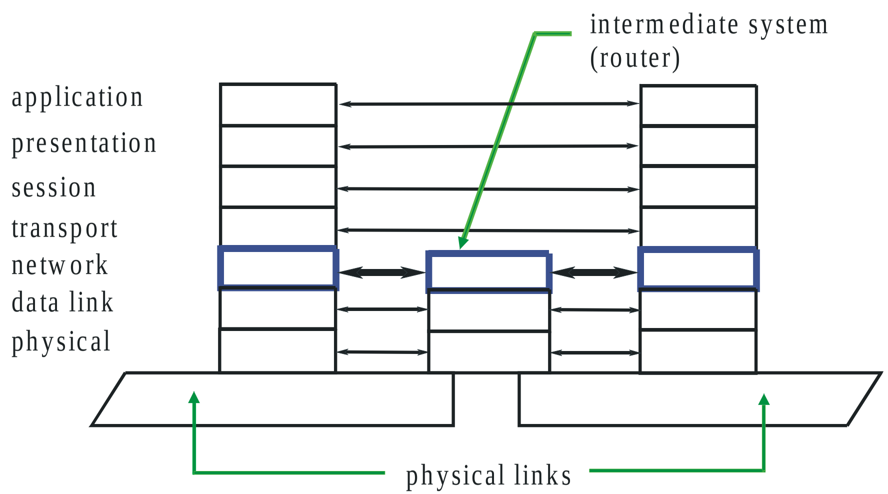


Sliding Window Example



4.1.3 Packet-Switched Networks

Intermediate systems contain only the layers 1-3. An example are Internet routers.



Virtual Circuits vs. Datagrams

Virtual Circuit

The path through the network is established at connection setup time. It remains the same for all packets for the duration of the entire connection. The intermediate nodes store path status information.

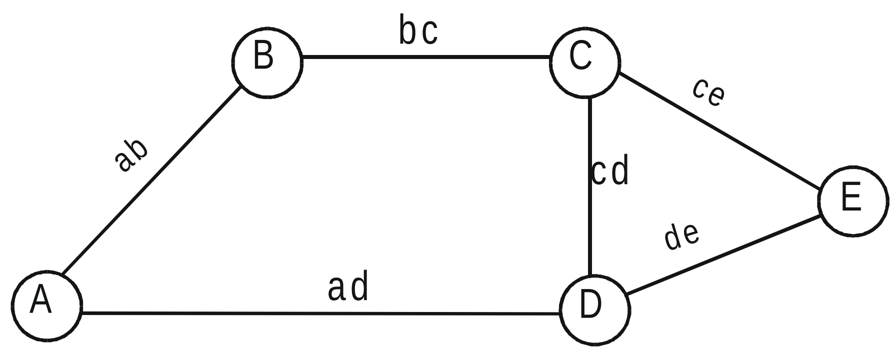
Datagrams

The destination address in each packet determines the next hop on the path. For each datagram the next hop is determined separately in each intermediate node. Different packets can take different paths, e.g., when a link has gone down.

Routing (1)

Each node contains a routing table with next-hop information for each destination address.

Network topology for our examples

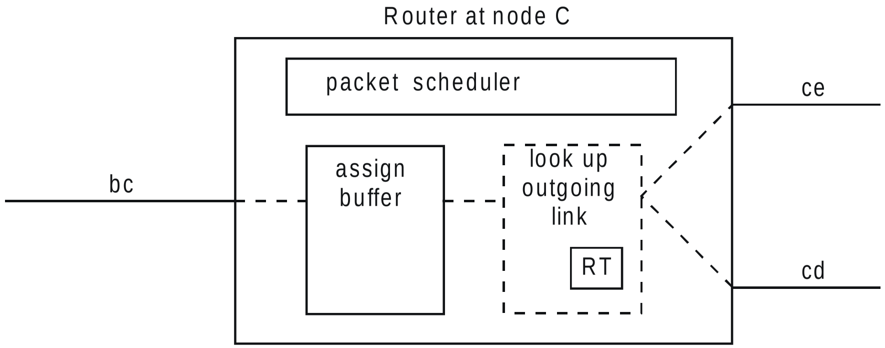


Routing (2)

Packet forwarding based on the routing table

RT routing table at C

From C to	link	cost
A	**bc**	**2**
B	**bc**	**1**
D	**cd**	**1**
E	**ce**	**1**



RT = Routing Table

Routing Algorithms

Task of the routing algorithm

- Determine the best path for packets from a given source node to a given destination node
- Load the routing tables in all nodes such that all best paths are known

Idea 1: Every Node Knows the Full Topology

Use Dijkstra's algorithm for SHORTEST PATHS

Build a tree of shortest paths from the sender (root) to all receivers as follows:

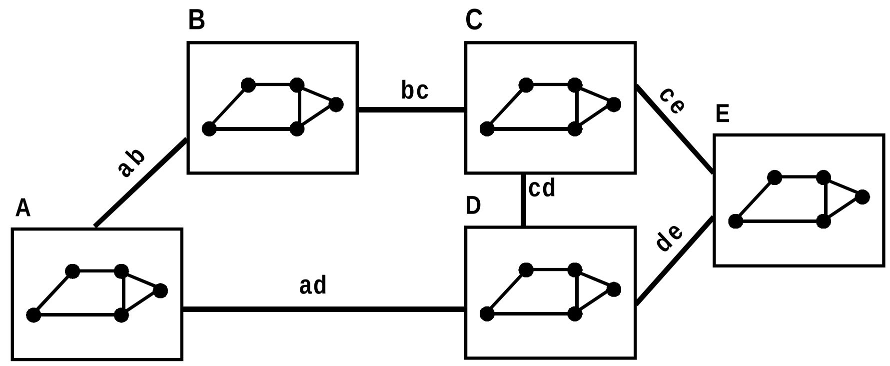
- Begin with the root
- Repeat until all nodes have been reached:
  - Of all those nodes not yet in the tree, add the one which is a neighbor of a tree node and has the shortest path from the root.

A more detailed description of this algorithm can be found in any book on algorithms, e.g.: R. Rivest, Ch. Leiserson, Th. Cormen: *Introduction To Algorithms*, McGraw Hill, 1990

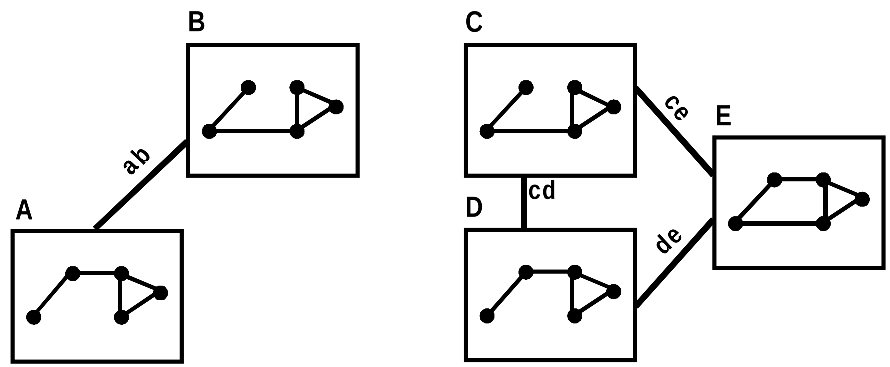
Problems

How do all nodes learn the current topology of the entire network? How can inconsistencies in a transition period be avoided when some nodes have already received the now topology, some still the old one (routing loops)?

"Full Topology" Routing (1)

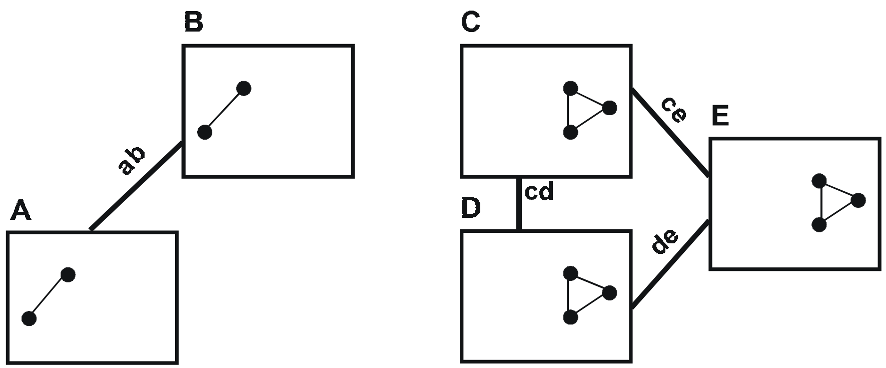


(a) Network in stable state



(b) Links *bc* and *ad* are cut

“Full Topology“ Routing (2)



(c) After an additional round of update messages

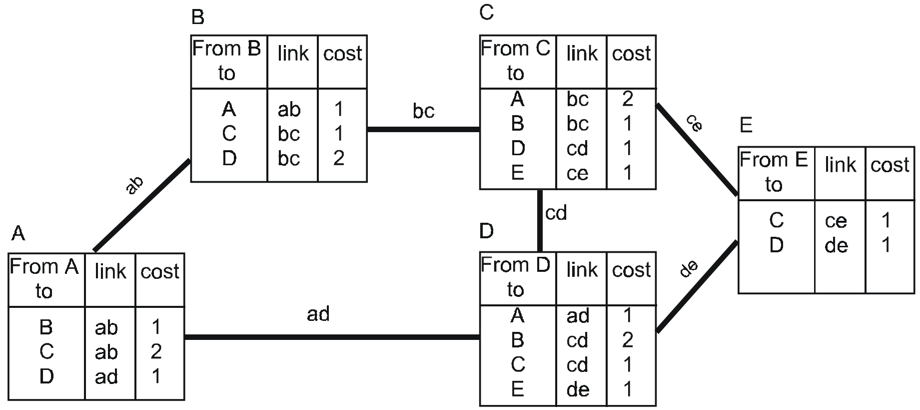
“Full Topology” routing is also called “link state routing“. The nodes maintain a “link state database“ in which they keep up-to-date information on the topology of the **entire** network. This “world view“ allows them to compute the optimal routing trees locally.

An algorithm of this class, called OSPF (Open Shortest Path First) is the one most widely used in the Internet today.

Idea 2: Routing with Distance Vectors

Only the distance to the destination and the next hop to take is known locally.

Distance vectors are exchanged periodically with neighbor nodes in routing table update messages.



From A to	link	cost
B	ab	1
C	ab	2
D	ad	1

From B to	link	cost
A	ab	1
C	bc	1
D	bc	2

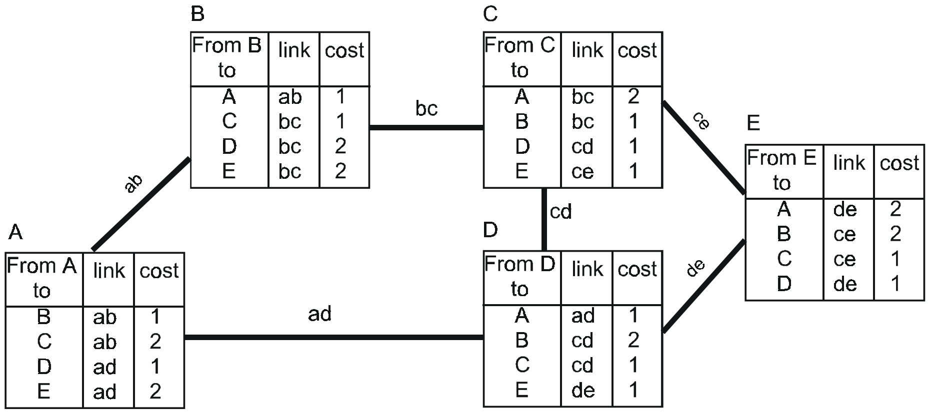
From C to	link	cost
A	bc	2
B	bc	1
D	cd	1
E	ce	1

From D to	link	cost
A	ad	1
B	cd	2
C	cd	1
E	de	1

From E to	link	cost
C	ce	1
D	de	1

(a) E is added as a new node

Distance Vector Routing



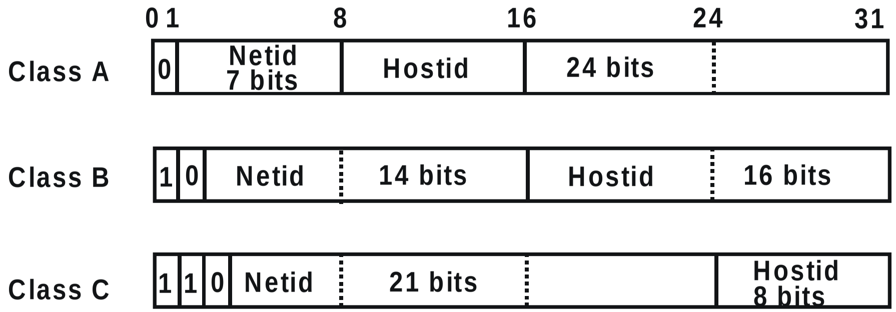
(b) After an additional round of messages

The protocol RIP (Routing Information Protocol) of the Internet is an example of distance vector routing.

Internet Addresses

An IP address is a hierarchical 32-bit address with a **netid** and a **hostid**. This keeps routing tables small and allows decentralized assignment of host addresses.

For point-to-point addressing three classes of addresses are provided:



For reasons hard to understand a notation with four decimal numbers (one for each byte) is popular.

Example

192.5.48.0 for a small LAN (class C)

Observations for Multimedia Streaming

- **The traditional algorithms and protocols destroy the continuous flow of packets!**
  They create considerable jitter (variance in the delay). This is true for
  - **all** MAC protocols in LANs
  - error control by retransmission (e.g., Go-Back-n)
  - flow control by sliding window (typically leads to stop-and-go traffic)
  - and many more algorithms!
- **The traditional algorithms and protocols provide no QoS guarantees!**
  Traditional networks, in particular the Internet, are thus often called **„best effort networks“**.
- **The traditional algorithms and protocols provide no support for multicast!**