de 1

Network Architectures

Joab R. Winkler

Department of Computer Science

The University of Sheffield

COM1100 and COM6720

de 2

Contents

1. Computer networks: Fundamental ideas
2. Network architectures
3. Error detection
4. Algorithms for reliable transmission
5. The Ethernet
6. Wireless networks
7. Routing algorithms

Slide 3

1. Computer networks: Fundamental ideas

Originally, a **network** signified a cable that connects terminals to a main frame computer. Other uses of the term **network** include:

- The voice telephone network
- The cable network for video signals

Characteristics of these networks:

- Each network can handle only one type of data (keystrokes, voice or video)
- They typically consider special purpose devices (terminals, hand receivers and television sets)

Slide 4

1. **What is a computer network ?**

   An interconnection of autonomous computers that

   - allows the interchange of data (connected via copper fibre, fibre optic cable, microwave link, etc.)
   - lacks a centralised control

2. **What distinguishes a computer network from these other networks ?**

   - Computer networks carry a range of different types of data, and they therefore support a wide range of applications
   - They are made from general purpose hardware

Summary: A characteristic of a computer network is its generality

de 5

Types of computer network

- A computer is called a **host**
- The physical medium (eg., coaxial cable, optical fibre) is called a **link**
- It is assumed that the link can carry data in both directions, called the **duplex mode**

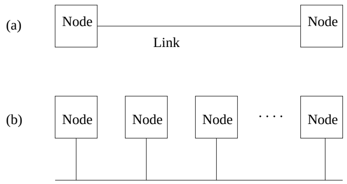


Figure 1: Physical (direct) links can be (a) point–to–point, or (b) multi–access.

de 6

Links, hosts and clouds

- Nodes transmit data over links via **network adaptors**
- A network adaptor encodes bit data (ones and zeros) into a signal
- The sending host may have to convert text, images, etc., into bit data before transmission. A receiving node host must then convert the bit data back again

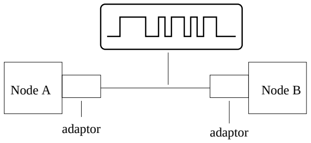


Figure 2: Nodes and adaptors.

Slide 7

If a computer network required that all hosts be connected to each other directly over a common physical medium, then

- the network would be limited in the number of computers that it could support

or

- the number of connections to each computer would be large, and thus the network would be unmanageable

Question : Is there an alternative ?

Answer : Yes. Connectivity between two hosts does not necessarily imply a direct physical link between them. Use either a **switched network** or an **internetwork**, also called an **internet**

Slide 8

A switched network

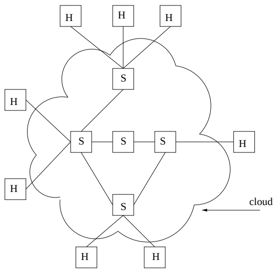


Figure 3: Switched network. S denotes a switch and H denotes a host.

- A switched network contains **switches** that store and forward data.
- A collection of switches is called a **cloud**.

de 9

- Switched networks may be **circuit switched** (used mainly for the telephone system) or **packet switched** (used in the majority of computer networks).
- A **circuit switched network** establishes a route along which all data is communicated.

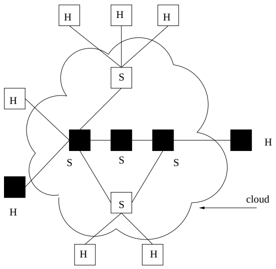


Figure 4: A circuit switched network and a data route.

e 10

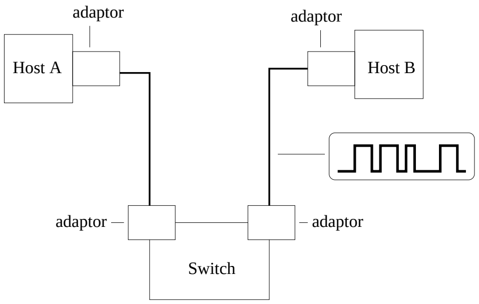


Figure 5: A circuit switched network with adaptors.

Slide 11

What are the differences between a circuit switched and a packet switched network?

- A circuit switched network does not have the facility to store information. The stream of bits passing into a circuit emerges at the other end with the same pattern and delayed by a constant interval
  - Think of the flow of an incompressible fluid through a pipe as an analogy
  - A circuit is set up when a request is made, and it is destroyed when the call is terminated
  - Circuit switching deals with information at a prescribed speed

Slide 12

- In a packet-switched network, the sender divides the message into packets which can take any route through network
- The full message is assembled at the receiver

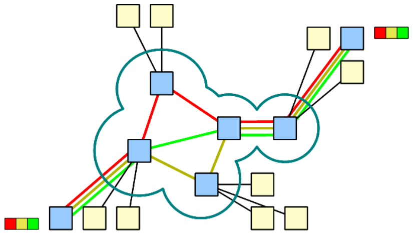


Figure 6: A packet switched network.

e 13

- A packet switched network operates by breaking a stream of data into short messages or **packets** that are handled individually by the network
  - At its simplest, switching consists of directing packets on their way to destinations
  - Packets are stored at each intermediate hosts, where they are switched
  - Each packet has its own routing information
  - Packet switching deals with information as short messages at any prescribed speed
  - This method of switching is therefore also known as **store–and–forward**

e 14

An internetwork (internet)

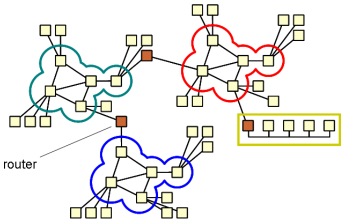


Figure 7: Interconnection of networks

- A generic connection of networks (that is, a collection of clouds joined by nodes called **routers** or **gateways**) is called an *internet*
- The **internet** is a particular type of internet built on the TCP/IP protocol

Slide 15

- Each cloud represents a network (point–to–point, multiple–access, or switched)
- A set of independent networks is connected to form an **internet**
- A node that is connected to two or more networks is called a **router** or **gateway**. It plays the same role as a switch because it forwards messages from one network to another network
- An internet is itself a network, and thus an internet can be built up from a connection of networks

Note:

- Although host–to–host connectivity has been achieved, communication between them must still be addressed

Slide 16

Summary

- A network can be defined recursively as two or more nodes that are connected by a physical link, or as two or more networks that are connected by one or more nodes
- This allows a nested structure of networks, where the lowest level is implemented by a physical link

Challenge

- Define an address for each node on the network, and use this address to route messages to the correct destination node

e 17

Requirements of a computer network

Connectivity

- Every node in a network should be able to communicate with every other node

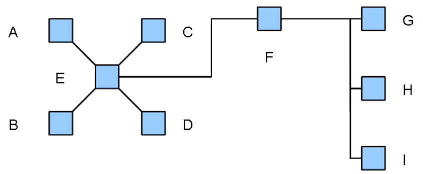


Figure 8: Interconnection of networks

e 18

Scalability

- Is the network flexible enough to add new nodes (or networks)?
- After adding a new node, how are all the other nodes able to find it?
- This leads to the problem of addressing

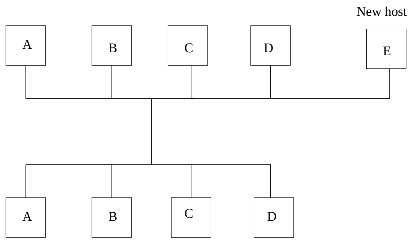


Figure 9: Interconnection of networks

Slide 19

Performance

- How quickly can data be transferred from one node to another node?
  - **Latency:** How long does it take data to travel from one node to another node?
  - **Bandwidth:** At what rate can data be transferred along a link

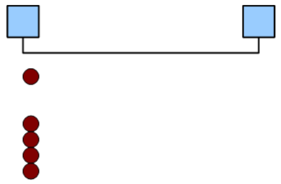


Figure 10: Interconnection of networks

Slide 20

Reliability

- Data might be corrupted by interference (electric storm, microwave)
- How do we detect errors and ensure a message arrives reliably?

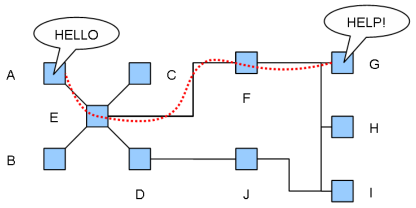


Figure 11: Reliability of networks

e 21

Reliability (cont'd)

- A node (or a group of nodes) in the network might fail, e.g., through a hardware fault or a switch becoming congested.
- How do we reroute the connection if this happens?

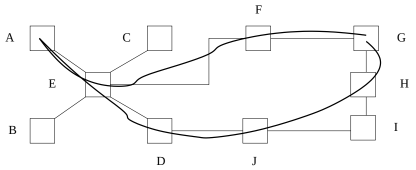


Figure 12: Reliability of networks

e 22

Cost effective resource sharing

- It is not practical to have a point-to-point connection between every pair of nodes in a large network
- How do we share a link (fairly) between multiple flows? This is the problem of **multiplexing**

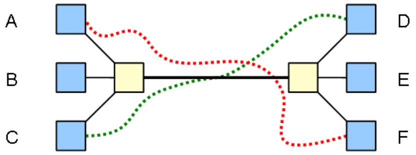


Figure 13: Multipliexing

Slide 23

Support for common services

- A network is responsible for more than simply the exchange of raw data between nodes
- In reality, we require applications that can exchange information, e.g., web browser, e-mail, instant messenger

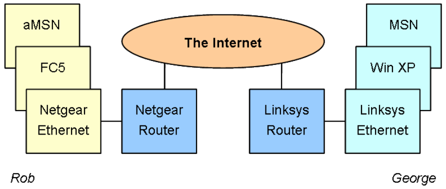


Figure 14: The left and right hand sides use different software and hardware. The left hand side is UNIX and the right hand side is Windows.

Slide 24

Network Size

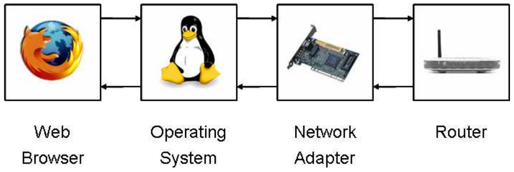
- One of the first things to decide: 'How big is the network?'
- The size of the network influences the choice of technologies that are used
- A smaller network class may constitute part of a larger class

Area Network	Scale	Examples
Personal (**PAN**)	metres	Bluetooth in phone or PDA
Local (**LAN**)	1 km	Home or small office
Metropolitan (**MAN**)	10s km	University Campus
Wide (**WAN**)	Earth	The Internet

e 25

Layering

- Between hosts: With so many users, types of software and hardware, and operating systems, how do hosts on a network communicate?
- Within a host: How can the user, application software, operating system and hardware on a single host/network interact?



e 26

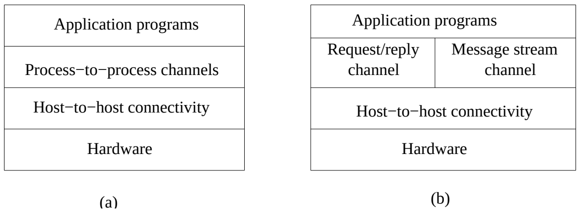
Examples of potential problems

- Two hardware manufacturers each want to design a network card. How can they be sure that the new network cards will be compatible?
- A programmer would like to write an e-mail application but does not want to have to study electrical engineering to understand the network cards and cables.

Slide 27

Solution: Layering

- Divide different types of task into **layers**
- **Abstraction:** Hide the details of a particular layer's implementation from the layers above and below



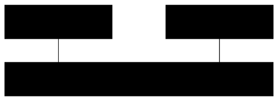
Application programs
Process−to−process channels
Host−to−host connectivity
Hardware
(a)

Application programs
Request/reply channel
Message stream channel
Host−to−host connectivity
Hardware
(b)

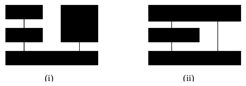
Figure 15: (a) and (b) Examples of a layered network. (b) has 2 abstractions at the same level.

Slide 28

- A particular implementation within a layer is called a **protocol**
- It provides a service interface to other adjacent protocols
- The interface lists what the protocol can do, without specifying how it is to be done
- There may be more than one protocol in a layer:

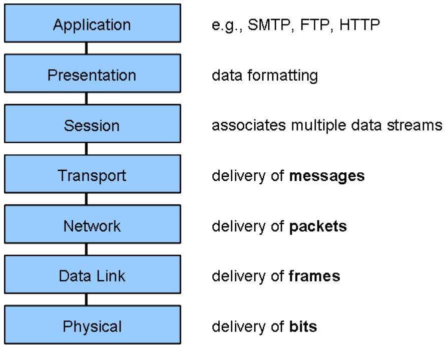


- A protocol may i) span more than one layer or ii) access protocols a few layers below:



e 29

- The OSI architecture defines a seven-layer model

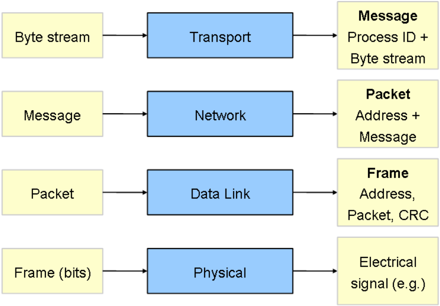


Application	e.g., SMTP, FTP, HTTP
Presentation	data formatting
Session	associates multiple data streams
Transport	delivery of **messages**
Network	delivery of **packets**
Data Link	delivery of **frames**
Physical	delivery of **bits**

e 30

An Example

- View the process like a factory with various specialisations



Byte stream	Transport	**Message** Process ID + Byte stream
Message	Network	**Packet** Address + Message
Packet	Data Link	**Frame** Address, Packet, CRC
Frame (bits)	Physical	Electrical signal (e.g.)

Slide 31

Transport Layer: Responsible for dividing the byte stream into messages

Network Layer: Responsible for adding the final address (the IP address) to the message

Data Link Layer: Responsible for adding local addresses of routers and hosts on the host's network

- After the Network Layer, the message is called a **packet**
- After the Data Link Layer, the message is called a **frame**

The process of adding addresses and other information is called **encapsulation**

Slide 32

Protocol graphs

The protocols that make up a network system are represented by a *protocol graph:*

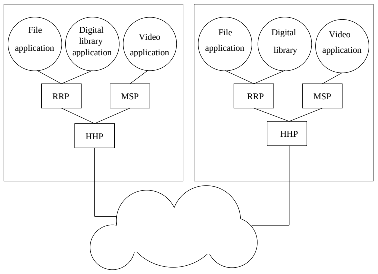


Figure 15: A protocol graph. RRP (request/reply protocol); MSP (message stream protocol); HHP (host–to–host protocol).

e 33

Assume that the file access program on host 1 wants to send a message to its peer on host 2 using the communication service offered by protocol RRP. Then

- The file application asks RRP to send the message
- The RRP invokes the services of the HHP in order to communicate with its peer
- A message is transmitted to its peer on host 2
- On arrival of the message at HHP on host 2, this protocol passes the message to RRP
- The message is forwarded to the file application on host 2

The application uses the *protocol* stack RRP/HHP.

e 34

Advantages of layering:

- The problem of building a network is broken down into a set of smaller and simpler problems
- Allows a modular design of a network:
  - The addition of a new service may only require the modification of the functionality at one level, and functions at other levels can be reused.

The abstract objects that constitute the layers of a network are called *protocols*

- A protocol provides a communication service that higher level objects, for example, application processes, use to exchange messages

Slide 35

The layering of a network has an analogy with many daily activities.

Example Consider sending a parcel from Europe to the US. The steps are:

- Write the address on the label
- Take the parcel to the post office
- The parcel would be placed, with many others, in a container which would be delivered to the airline (transport company)
- The parcel would then be placed in the airplane and flown to its destination, possibly with stops en route
- At the final destination, the airline would transfer the parcel to the local post office, who would deliver it to the local sorting office
- The parcel would then be delivered to the addressee

Slide 36

This order of events generates a hierarchy:

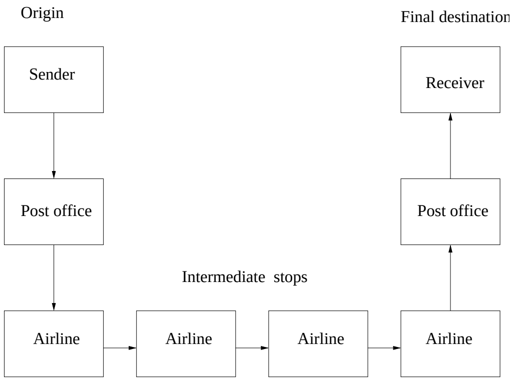
1. The user level (the sender and receiver)
2. The post office
3. The airline

- Each level uses the level below it as an abstract tool:
  - The sender and receiver are not concerned with the details of the post office and the details of the airline
  - The post office is not concerned with the operational details of the airline
- Each level has representatives at the levels above and below it for liaison

□

A computer network provides a similar abstraction of a process into several layers

e 37



Hierarchy of stages of sending a parcel by plane.

e 38

Encapsulation

Questions

1. How does the RRP know where to direct a message ?
2. How can the RRP communicate control information to its peer ?

Answer

- The RRP attaches a header to each message:
  - The header is short, usually a few dozen bytes long
  - This header is usually at the front of each message, but it may be at the back of the message, in which case it is called a trailer
- The rest of the message is called the body or payload

The application's data is *encapsulated* in the new message created by the protocol RRP.

Slide 39

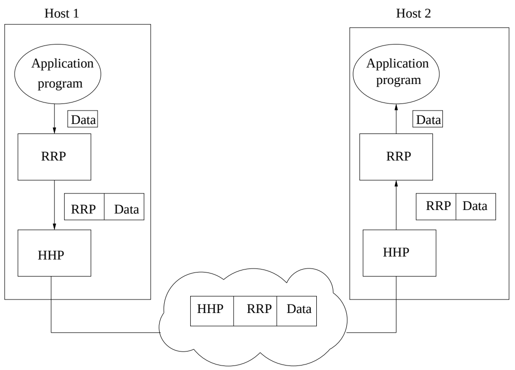


Figure 16: High–level messages are encapsulated inside low–level messages.

Slide 40

The sequence of events:

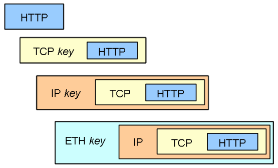
- The RRP encapsulates the data
- The HHP encapsulates RRP's message by attaching its own header
- The HHP sends the message to its peer over the network
- The HHP strips off its header from the front of the message and interprets it
- The HHP passes the body of the message to the RRP
- The RRP strips off its header from the front of the message and interprets it
- The RRP passes the body of the message to the application program in host 2

Note:

- The order of events in host 2 is the reverse of the order of events in host 1
- The message received by the application program in host 2 is exactly the same as the message sent by the application program in host 1

e 41

- The protocol at each stage treats encapsulated packets as raw data. Consider a HTTP (web-page) request over an Ethernet:



HTTP = **Hypertext Transfer Protocol**

TCP = **Transmission control protocol** Delivery of packets reliably and in the right order

IP = **Internet protocol** Delivery of packets over routers between different network technologies

ETH = Ethernet protocol

e 42

Addressing, routing and casting

- Every node in a network must have an *address* in order that other nodes can communicate with it
- Switches and routers are used to forward (or "route") a message towards its destination
- Unicast, broadcast and multicast:
  - A message may be sent to only one node on the network – *unicast*
  - A message may be sent to all nodes on the network – *broadcast*
  - A message may be sent to only some nodes on the network – *multicast*

Summary:

- Each node in a network is defined by its address
- A network must support unicast, broadcast and multicast addresses

Slide 43

Multiplexing

Networks must be efficient – and thus packet–switched networks are used.

Problems:

- How can all the hosts on a network communicate with each other ?
- How can several hosts share the *same link* in the network at the same time ?

Answer: Use *multiplexing* – a system resource is shared among multiple users.

Compare with timesharing in a computer operating system, in which a CPU is shared (multiplexed) between several jobs.

Slide 44

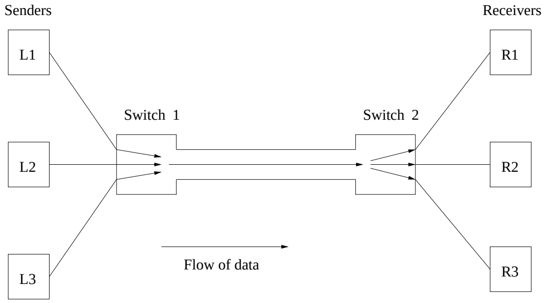


Figure 17: Multiplexing multiple flows of data over a physical link.

e 45

Figure 5 shows three senders (L1, L2, L3) and three receivers (R1, R2, R3) that share a switched network and are connected by one physical link.

- Assume that L1 is communicating with R1, L2 is communicating R2, and L3 is communicating with R3
- There are three flows of data – corresponding to the three pairs of hosts
- Each flow of data is *multiplexed* onto the physical link by switch 1, and then *demultiplexed* back into separate flows by switch 2

e 46

Methods of multiplexing

- Synchronous time–division multiplexing (STDM):
  - Divide time into quanta of equal size
  - The data is sent over the link in a round–robin manner
- Frequency division multiplexing (FDM):
  - Transmit each flow of data over the physical link at a different frequency, in the same way that different TV stations are transmitted at different frequencies over the same physical link at the same time

Slide 47

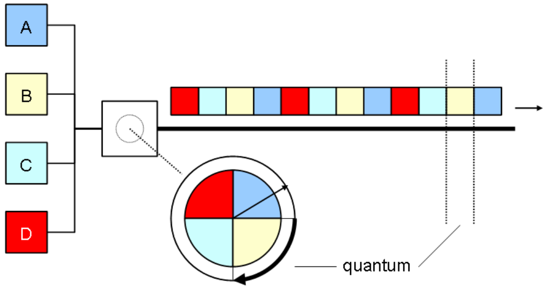


Figure 18: Time division multiplexing.

Slide 48

Example Frequency division multiplexing

- The traditional analog telephone system uses 12 channels, each covering 4 kHz. The total bandwidth is 48 kHz.
- These are mixed together (multiplexed) into the 60–108 kHz band. Note that the bandwidth is still 48 kHz.
- In each 4 kHz channel, only 3 kHz is used for the signal. The bands of 500 Hz on each side guard against interference from adjacent channels.
- Filters at the receiver separate the useful signal from the sidebands, and separate the signals for each channel

□

e 49

- Multiplexing is used in both circuit–switched and packet–switched networks
- Multiplexing in packet–switched networks is achieved by interleaving packets. Recall that each packet has a lot of information (source, destination, channel number, etc.), and thus a device can handle more than one packet stream at a time

Limitations

- If data is not being transmitted between a pair of hosts, then its share (in time or frequency) of the physical link is idle, which leads to inefficiency
- For both STDM and FDM, the number of data flows is fixed; it is not practical to resize or add time quanta for STDM, or add new frequencies for FDM

Example The amount of idle time in computer communication can be large. For example, downloading a web page may be quick, but you may spend a long time reading the page, during which time the link is idle. □

e 50

Solution: Statistical multiplexing

The solution is to use *statistical multiplexing*:

- Like STDM, statistical multiplexing operates in the time domain

but

- unlike STDM, data is transmitted from each flow on demand rather than during a predetermined time slot

Example If only one flow of data must be sent, then it is sent without waiting for its time quantum to come round again, and thus quanta are not assigned to flows that do not have data. This results in a much more efficient system. □

But if a flow of data that was previously empty now has data, how can it access the link ?

Slide 51

Solution

Statistical multiplexing defines an upper bound on the size of the block of data that each flow is permitted to transmit at a given time.

- This block of data is called a *packet* (see earlier notes on packet–switched networks)
- A host cannot send a large message in one packet, and thus the source may need to fragment the message into several packets
- Each flow of data sends a sequence of packets over the network
- A decision is made by the switches as to which packet to send

Note:

- If only one flow of data need be sent, then its packets are sent back–to–back
- If more than one flow has data that must be sent, then the packets are interleaved in the link

Slide 52

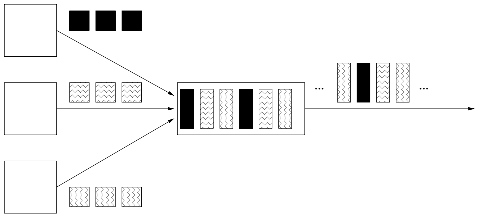


Figure 19: A switch multiplexing packets from multiple sources onto one shared link.

e 53

Question What happens if the switch receives packets at a higher rate than can be accommodated by the shared link ?

Answer Congestion:

- Some packets are stored in memory
- When the available memory runs out, packets must be deleted

e 54

Reliability: How can networks fail ?

- Computer crashes, cut fibres, electrical interference (lightning strike, power surge, etc.)
- Switches run out of memory (congestion)

Errors at bit level due to lightening strikes, power surges, etc.:

- Since a packet is transmitted over a physical link, *bit errors* may be introduced into the data:
  - A 1 may be switched to a 0, and vice–versa
  - More often, a *burst error* occurs – several consecutive bits are corrupted
  - Usually about 1 bit in 10^7 bits for copper cables
  - Usually about 1 bit in 10^{13} for optical fibres
  - Can sometimes be detected and corrected

Slide 55

Errors at packet level – an entire packet (not just a bit) is lost. Reasons include:

- A packet contains a bit error that cannot be corrected, and must therefore be discarded
- A switch is so overloaded that it cannot store the packet, and it is therefore deleted (congestion)

Note:

Packet–switched networks have the ability to round a failed link or node

Slide 56

Common units in networking

bit = **bi**nary digi**t** $\in \{0, 1\}$, 1 byte = 8 bits

GB: gigabyte	=	2^{30} bytes	=	2^{33} bits
MB: megabyte	=	2^{20} bytes	=	2^{23} bits
KB: kilobyte	=	2^{10} bytes	=	2^{13} bits

1 KB = 1024 bytes (NOT 1000 bytes) = 8×1024 bits

1 MB = 1024×1024 bytes = $8 \times 1024 \times 1024$ bits

Gbps: gigabit per second	=	10^9 bits per second
Mbps: megabit per second	=	10^6 bits per second
Kbps: kilobit per second	=	10^3 bits per second

e 57

Bandwidth

Definition:

The bandwidth of a link/channel is the number of bits per unit time that can be transmitted over the link/channel

Measured in bps (bits per second)

- The *bandwidth* is equal to the number of bits that can be transmitted over the network in a given period of time

Example Consider a server transmitting a 25 megabyte (MB) image file over a channel of bandwidth 10 Mbps. How long does it take to transmit the file (i.e., to upload the file on to the link)?

Convert the data volume into bits: 25 MB = $25 \times 1024 \times 1024 \times 8$ bits.

Time taken = $\frac{25 \times 1024 \times 1024 \times 8}{10 \times 10^6} = 20.9715$ seconds. □

e 58

The bandwidth is often used as a measure of the performance of a network because it is an upper bound of the value of the throughput.

Example A communication link may have a bandwidth of 10 Mbps, but it may only achieve a throughput of 2 Mbps in practice. The difference arises because of inefficiencies in the implementation of the network. □

Factors that affect bandwidth:

- The physical/electrical characteristics of the links
- The software overhead that is required for handling/transforming each bit of data
- The load on the network

Slide 59

Latency (delay)

Definition:

The latency (delay) of a system is the length of time it takes to send a message from one end of the system to the other end

Three components:

- Speed of light propagation delay
  2.3×10^8 m/s in cable
  2.0×10^8 m/s in fibre
- Length of time taken to transmit a unit of data (usually a packet). This is a function of the network bandwidth and the size of the packet in which the data is carried. For example, it takes more than 100 ms for a bit to travel from the earth to a satellite, and back to earth.
- Queuing delays inside the network:

Slide 60

  - Electronic devices in a network (hubs, bridges, packet switches, etc). introduce a switching delay
  - An electronic device must wait until all the bits of a packet have arrived, and then it takes a small amount of time to choose the next stage in the transmission. Fast CPUs and special purpose hardware have made switching delays relatively insignificant
  - Since most LANs use shared media, computers must wait until the medium is available. This delay, which is usually not large, is called an *access delay*

e 61

Latency = Propagation + Transmit + Queue

Propagation = Distance /Speed of light

Transmit = Size/Bandwidth

where

Distance is the length of the wire over which the data travels

Speed of light is the speed of light in the medium

Bandwidth is the bandwidth at which the packet is transmitted

e 62

Latency is sometimes defined as the time

first bit sent to first bit received

and sometimes as

first bit sent to last bit received

Why is this difference important ?

- The propagation is the time taken for the first bit of the message to arrive at the destination
- The transmit accounts for the length of the message:
  - If we want to calculate the time *first bit sent to first bit received,* we only use the propagation term
  - If we want to calculate the time *first bit sent to last bit received,* we use the propagation and transmit terms

Note: It is assumed that the delay is zero.

Slide 63

Notes:

- If the message consists of only a single bit and transmission occurs over one link, and not the entire network, then the latency is equal to the propagation delay
- The latency is the sum of the three factors upon which the transmission time achieved in practice depends
- The latency that is *achieved in practice* is called the throughput, which is measured end–to–end
- By calculating each of the three terms that make up the latency, we can calculate the relative importance of each factor, and thus determine, for each network and each message, the dominant factor(s) in the latency

Slide 64

Example Consider downloading a 25 megabyte (MB) image. If the channel has a bandwidth of 10 Mbps, then since

$$25\text{MB} = 25 \times 2^{23} \text{ bits}$$

it takes

$$25 \times 2^{23}/10 \times 10^{6} = 20.97 \text{ seconds}$$

to transmit the image.

Clearly, an RTT of 1 ms or 100 ms is not important, and thus the latency is dominated by the bandwidth. □

e 65

Latency vs. bandwidth

Both latency and bandwidth are required to define the performance characteristics of a channel. The relative importance is a function of the application.

Example Consider a client that sends a 1 byte (=8 bits) message, and receives a message of the same size in return. Consider a link across a room with a round–trip time (RTT) of 1 ms, and a transcontinental link with a RTT of 100 ms.

- If the channel is 1 Mbps, then the time to transmit a byte is 8μs
- If the channel is 100 Mbps, then the time to transmit a byte is 0.08μs

In both cases, the latency is dominated by the RTT and thus the bandwidth (1 Mbps or 100 Mbps) is not important. □

e 66

Delay × bandwidth

Question How many bits can be sent before the first one arrives at the receiver ?

Answer Suppose a link has a bandwidth of F Mbps and a latency of L seconds. Then the bandwidth is equal to $F \times 10^6$ bits per second, and thus the number of bits transmitted in L seconds is $F \times L \times 10^6$ bits, or $F \times L \times 10^6 \times \frac{1}{8}$ bytes of data can be stored.

Example A transatlantic link with a latency of 50 ms and a bandwidth of 45 Mbps can store

$$45 \times 10^6 \times 50 \times 10^{-3} \times \frac{1}{8} \approx 280 \text{ KB}$$

of data. □

Slide 67

Measuring throughput

Recall that the throughput is a measure of the performance of a channel that is achieved in practice.

Throughput = TransferSize/TransferTime

TransferTime = TransferSize/Bandwidth + RTT

The RTT accounts for the request to transfer the data, and for the data to be sent.

Example Consider fetching a 1 MB file across a 1 Gbps network that has a 100 ms RTT.

1 MB = $8 \times 1024 \times 1024$ bits and thus fetching this file takes 8.39 ms. The transfer time is therefore 100 ms + 8.39 ms = 108.39 ms, and the throughput is

$$1\text{MB}/108.39\text{ms} = 8 \times 1024 \times 1024/108.39 \times 10^{-3} = 77.39 \text{ Mbps}$$

□

Slide 68

Application performance needs

Assumption so far: Application programs require as much bandwidth as the network can support.

Is this always correct ? Not always; the needs of the application must also be considered.

Example Consider the transmission of a 128 KB video frame 30 times per second. This requires

$$128 \times 1024 \times 8 \times 30 = 31.46 \text{ Mbps}$$

and thus a network that can provide a bandwidth greater than about 32 Mbps does not offer any advantage. □

e 69

Example Calculate the latency (first bit sent to last bit received) for the following cases:

1. A 10 Mbps Ethernet with a single store–and–forward switch in the path, and a packet size of 5000 bits. Assume that each link introduces a propagation delay of 10μs and that the switch begins retransmitting immediately after it has finished receiving the packet.
2. Same as part 1 above but with three switches.
3. Same as part 1 above but assume that the switch implements "cut–through" switching, that is, it is able to begin retransmitting after the first 200 bits have been received.
4. Same as part 3, but with 3 cut–through switches.

e 70

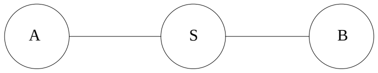
Answer

1. Since a packet consists of 5000 bits, the delay on each side of the link due to the bandwidth is 500μs. Since the propagation delay for each link is 10μs, it follows that the total latency is $(2 \times 500) + (2 \times 10) = 1020\mu$s.
2. With three switches and four links, the latency is equal to $(4 \times 500) + (4 \times 10) = 2040\mu$s
3. The cut–through introduces a delay of 20μs. There is still one delay of 500μs waiting for the last bit to arrive, and a propagation delay of 20μs. Thus the total latency is 540μs. The last bit still arrives 500μs after the first bit: The first bit still has two link delays and one switch delay, but does not have to wait for the last bit before the switch forwards the message.
4. With three cut–through switches, the total delay is $500 + (3 \times 20) + (4 \times 10) = 600\mu$s.

□

Slide 71

Example Hosts A and B are connected to a switch via a 10 Mbps link, as shown below. The propagation delay in each link is 20μs and S is a store–and–forward device that begins transmitting a packet 35μs after it has finished receiving it. Calculate the total time required to transmit 10000 bits from A to B if the message is sent (a) as a single packet, and (b) as two 5000 bit packets that are sent one after the other.



(a) The transmit time on each link due to the bandwidth is 1000μs, and thus the delay on each link is 1020μs. The total delay is therefore $(2 \times 1020) + 35 = 2075\mu$s.

Slide 72

(b)

T $= 0$	Start
T $= 500\mu$s	A finishes sending packet 1. Starts sending packet 2
T $= 520\mu$s	Packet 1 arrives at S
T $= 555\mu$s	Packet 1 departs for B
T $= 1000\mu$s	A finishes sending packet 2
T $= 1055\mu$s	Packet 2 departs for B
T $= 1075\mu$s	First bit of packet 2 arrives at B
T $= 1575\mu$s	Last bit of packet 2 arrives at B

This method is therefore faster. □

e 73

Encoding and Framing

- When counting in binary notation, only the digits 0 and 1 are used
- When a digit increments beyond 1, it returns to 0, and we must add 1 to the next position
- A binary digit (0 or 1) is called a bit

The binary numbers are

$0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100,$

$1101, 1110, 1111, 10000, 10001, \ldots$

e 74

- In base 10:

$$\begin{aligned} 3,256 &= 3 \times 10^3 + 2 \times 10^2 + 5 \times 10^1 + 6 \times 10^0 \\ &= 3 \times 1000 + 2 \times 100 + 5 \times 10 + 6 \times 1 \end{aligned}$$

- In binary (base 2):

$$\begin{aligned} 10100101 &= (1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) \\ &\quad +(1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) \\ &= 128 + 32 + 4 + 1 \\ &= 165 \end{aligned}$$

Where the possibility of confusion exists (e.g., does 1001 mean 'one thousand and one' or 'nine'), mark the base in subscript. e.g., 1001_{10} or 1001_2.

Slide 75

Converting binary to decimal

- The task is to convert a binary number to decimal, e.g., 11010011. First, write out the binary number.

1	1	0	1	0	0	1	1

- Second, starting with one on the right-hand side, write the powers of two above each position, by doubling each time, as shown below:

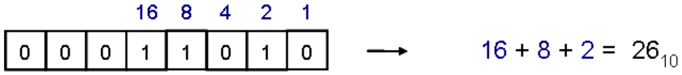
128	64	32	16	8	4	2	1
1	1	0	1	0	0	1	1

- To find the answer as a decimal, add up the powers of two for all the positions containing a 1:

$$128 + 64 + 16 + 2 + 1 = 211_{10}$$

Slide 76

Example Convert 00011010_2 to decimal

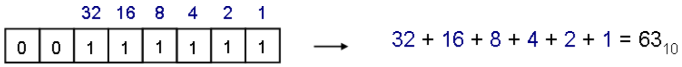


			16	8	4	2	1
0	0	0	1	1	0	1	0

$\longrightarrow$ 16 + 8 + 2 = 26_{10}

□

Example What is the highest number it is possible to express with 6 bits?



		32	16	8	4	2	1
0	0	1	1	1	1	1	1

$\longrightarrow$ 32 + 16 + 8 + 4 + 2 + 1 = 63_{10}

□

e 77

Converting decimal to binary

- Call the number to be converted X. Write out the powers of two above each position, as before, but leave the boxes blank

128	64	32	16	8	4	2	1

- Start from the left
- Is X (strictly) smaller than the power of two above the position?
  - If yes, write 0
  - If no, write 1 and subtract the power of two from X
- Move one box to the right and repeat from the step above, until you have filled in all the boxes

e 78

Example Convert 97_{10} to binary.

128	64	32	16	8	4	2	1

97 is smaller than 128, and thus write 0.

128	64	32	16	8	4	2	1
0	1						

97 is not smaller than 64, and so write 1 and subtract $97 - 64 = 33$.

128	64	32	16	8	4	2	1
0	1	1					

33 is not smaller than 32, and so write 1 and subtract $33 - 32 = 1$.

128	64	32	16	8	4	2	1
0	1	1	0	0	0	0	

1 is smaller than $16, 84$ and 2, so write four 0s.

128	64	32	16	8	4	2	1
0	1	1	0	0	0	0	1

1 is not smaller than 1, and so write 1.

□

Slide 79

Binary addition

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

Slide 80

Example Add the binary numbers 10011011_2 and 00111001_2.

$$\begin{array}{ccccccccc} & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ + & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ \hline & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ & & 1 & 1 & 1 & & 1 & 1 & \end{array}$$

□

Try other examples and check the answers by decimal addition

e 81

Bitwise AND

- There are some bitwise operators for binary numbers
- The AND operation is performed on individual bits and is tabulated below

0 AND 0 = 0

0 AND 1 = 0

1 AND 0 = 0

1 AND 1 = 1

Example:

```
        1 0 0 1 1 0 1 1
AND         1 0 1 0 0 1
        ---------------
        0 0 0 0 1 0 0 1
```

e 82

Bitwise OR

- The OR operation is performed on individual bits and is tabulated below

0 OR 0 = 0

0 OR 1 = 1

1 OR 0 = 1

1 OR 1 = 1

Example:

```
        1 0 0 1 1 0 1 1
OR          1 0 1 0 0 1
        ---------------
        1 0 1 1 1 0 1 1
```

Slide 83

Bitwise XOR

- The XOR (exclusive OR) operation is performed on individual bits and is tabulated below
- XOR stands for exclusive or and means ‘one or the other, not both’

0 XOR 0 = 0

0 XOR 1 = 1

1 XOR 0 = 1

1 XOR 1 = 0

Example:

```
        1 0 0 1 1 0 1 1
XOR         1 0 1 0 0 1
        ---------------
        1 0 1 1 0 0 1 0
```

Slide 84

Bitwise NOT

- The NOT operation is performed on individual bits and is tabulated below

NOT 0 = 1

NOT 1 = 0

Example:

```
NOT     0 0 1 0 1 0 0 1
        ---------------
        1 1 0 1 0 1 1 0
```

e 85

Example Evaluate (5 AND 44) OR 9 and give the result as a decimal.

- Convert 5 and 44 to binary:

$$5_{10} = 101_2 \qquad \text{and} \qquad 44_{10} = 101100_2$$

- AND them together:

$$101 \text{ AND } 101100 = 100$$

- Convert 9 to binary: $9_{10} = 1001_2$
- Perform a bitwise OR:

$$100 \text{ OR } 1001 = 1101$$

- Convert the result back to decimal:

$$1101_2 = 13_{10}$$

□

e 86

ASCII

- ASCII is a scheme for encoding text characters as bytes
- To convert a text character to an ASCII code, you can look it up in an ASCII table. Alternatively, to convert:
  - An upper-case letter to an ASCII code, find its position in the alphabet (A = 1, B = 2, etc.) and add 64
  - A lower-case letter to an ASCII code, find its position in the alphabet (a = 1, b = 2, etc.) and add 96
  - A digit to an ASCII code, add 48 to its value
  - Space is 32
- For example, 'Abc 123' has the encoding $65\ 98\ 99\ 32\ 49\ 50\ 51$

Slide 87

Bit encoding

Problem How can the binary data of the source be encoded into signals that the links are able to carry, and then decode the signal back into the binary data at the receiving host ?

A *network adaptor* has several functions:

- It is a piece of hardware that connects a node to a link
- A signalling component in the network adaptor encodes the bits into signals at the sending node, and decodes signals into bits at the receiving node

Assume that we are working with two discrete signals: High and low. These may correspond, for example, to two different voltages on a copper wire.

Slide 88

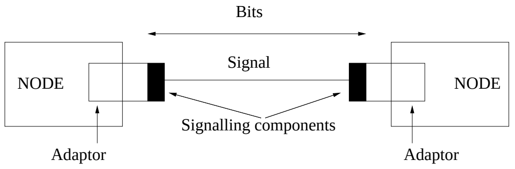


Figure 20: Signals travel between signalling components; bits flow between adaptors.

e 89

Non-return to zero encoding

- Map data value 0 to low signal
- Map data value 1 to high signal

This is called *non–return to zero* (NRZ)

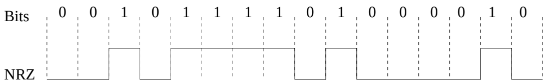


Figure 21: NRZ coding of a bit stream.

- The rate at which an encoder can make transitions is called the **baud rate**. This is not always the same as the bit rate.

e 90

Problems with NRZ

- Assume that a sequence of several consecutive 1s or 0s has been sent, so that the signal stays on high or low for an extended period:
  - The receiver keeps an average of the signal that it has seen, and uses this value to discriminate between low and high signals
  - It is encoded as 0 if the signal is less than this average, and it is encoded as 1 if the signal is greater than this average
  - Too many consecutive 1s or 0s cause this average to change, making it difficult to detect a significant change in the signal
  - This is called *baseline wander*
- Each network adaptor has a clock, and a signal is transmitted upon each clock tick. If the clocks of the sender and receiver are not synchronised, then the receiver may not be able to correctly recover the signals from the sender. This is called **clock drift**.

Slide 91

Example Clock drift.

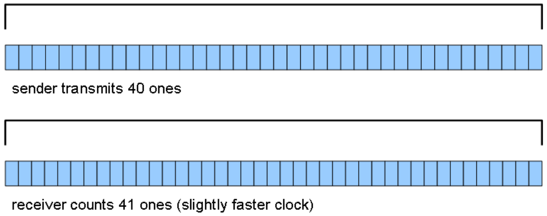


Figure 22: Clock drift

□

Slide 92

A solution: Manchester encoding

Transmit the exclusive–OR of the NRZ–encoded data and the clock.

Note: The exclusive–OR of two logical variables p and q is

- *True* if p and q have different logical values
- *False* if p and q have the same logical values

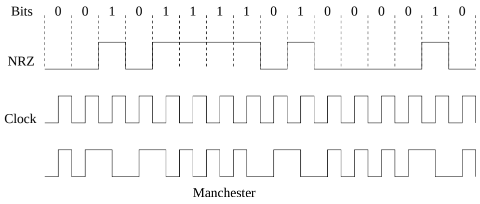


Figure 23: Manchester coding of a bit stream.

e 93

Disadvantage of Manchester encoding

The rate at which the signal changes, called the *baud rate*, is twice the rate of the transmitted bits:

- The receiver has only half the time to detect each pulse of the signal
- The bit rate is half the baud rate, and so Manchester coding is only 50% efficient

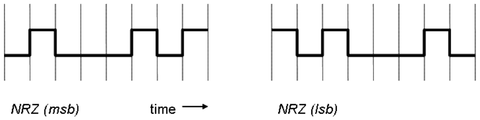
Note: If the receiver is able to maintain the faster baud rate required by Manchester coding, then NRZ coding would be able to transmit twice as many bits per unit time.

e 94

Bit order

- The manner in which different technologies interpret an 8-bit pattern as a byte depends on bit order:
  - **most significant bit** (msb) encoding transmits the highest significant bit first
  - **least significant bit** (lsb) encoding transmits the least significant bit first
- For example, the character ‘E’ transmitted in NRZ as

$$\mathsf{E} = 64 + 5 = 69 = 01000101_2$$



Slide 95

Framing

- Blocks of data (called frames, not packets, at this level) are exchanged between nodes
- The network adaptor enables the nodes to exchange frames, which results in a sequence of bits being sent over the network

Problem: How can a receiver detect where a frame starts and ends ?

This is called the **framing problem**

There are several solutions to the framing problem, each one of which is defined by a different protocol.

Slide 96

Byte-oriented protocols

A frame is viewed as a collection of bytes.

Examples BISYNC (Binary Synchronous Communication), DDCMP (Digital Data Communication Message Protocol) and PPP (Point–to–Point Protocol)

A frame is displayed as a sequence of labelled fields, from left (the beginning of the frame) to right (the end of the frame).

8	8	8		8		8	16
SYN	SYN	SOH	Header	STX	Body	ETX	CRC

Figure 24: BISYNC frame format. The number above each field is its length in bits.

e 97

- The beginning of the frame is denoted by a special SYN (for synchronisation) character
- The data portion of the frame is contained between special *sentinel characters*: STX (start of text) and ETX (end of text)
- The SOH (start of header) field serves the same purpose as the STX field
- The CRC (cyclic redundancy block) is used to detect transmission errors

Question What happens if the ETX or DLE characters appear in the data portion of the frame ?

Answer Use *character stuffing*.

Precede each character with a DLE character

e 98

Example Suppose we wish to transmit the message ‘Good Afternoon’

- The data transmitted will have the following form (from left to right):

SYN	SYN	STX	G	o	o	d		A	f	t	e	r	n	o	o	n	ETX	BCC

- Header information has not been included in this example and BCC denotes the block check character
- The total frame size is 18 bytes (excluding SYNs)

□

Slide 99

Bit–oriented protocols

- Bit-oriented protocols do not involve byte boundaries, only the delivery of a stream of bits (which may be interpreted as bytes by a higher-level protocol)

10010010	01101111	11011011	00101101	00101101

byte-oriented

. . . 1001001001101111110110110010110100101101 . . .

bit-oriented

- Two examples of bit-oriented protocols are High-Level Data Link Control (HDLC) and Synchronous Data Link Control (SDLC)

Slide 100

Example HDLC (High–level Data Link Control) Protocol

- The beginning and end of a frame are defined by the sequence 01111110
- This sequence is also transmitted when the link is idle so that the sender and receiver clocks remain synchronised
- Use *bit stuffing* if the string 01111110 appears in the data (including across byte boundaries)

8	16		16	8
Beginning sequence	Header	Body	CRC	Ending sequence

Figure 25: HDLC frame format. The number above each field is its length in bits.

101

Bit–stuffing in HDLC Protocol

- The sender inserts a 0 whenever five consecutive 1s have been transmitted
- If five consecutive 1s are sent, then the receiver acts as follows:
  - If the next bit is 0, then it must have been stuffed, and it is therefore removed
  - If the next bits are 10, then the end of frame marker has been received
  - If the next bits are 11, then an error has been introduced and the whole frame is deleted because the frame cannot be the data (it has not been stuffed) and it cannot be the end of a frame (it has seven 1s).

102

Example

- It is required to transmit the following:

1	0	0	1	1	1	1	1	1	0	0	1	1	1

- We transmit as far as 10011111. Five 1s have been transmitted so we stuff a 0:

1	0	0	1	1	1	1	1	0	1	0	0	1	1	1

- There are no groups of five or more 1s, so we transmit as the remainder directly

□

Slide 103

Error detection

- Errors are sometimes introduced into frames by electrical interference, thermal noise, etc.
- How can they be detected?
  - A common technique is cyclic redundancy check (CRC)
  - Used in the HDLC protocol
- Can an error that has been *detected* be *corrected*?
  - Yes, but only under certain circumstances
  - Use an error correcting code

Slide 104

Although errors are rare (1 in 10^7 for copper cable, 1 in $10^{12} - 10^{14}$ for optical fibre), they must be detected. How can this be done ?

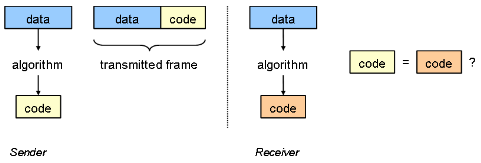
- Naive method:
  - Always transmit two copies of the frame
  - Wastes bandwidth (50% overhead)
- Better method:
  - Add redundant bits to the frame
  - Overhead is small
  - An ethernet frame with a 12000–bit body contains only 32 extra bits. This allows reliable error detection

Objective: Find an error detection principle that introduces the fewest redundant bits, and detects the largest number of errors

105

Error detection principle

- The sender transmits a redundant error-checking code which is computed from the frame data
- The receiver also computes an error-checking code from the frame data using the same procedure
- If the received and computed codes differ, then either the frame or the error-checking bits were corrupted



106

Error detection procedure

By definition, redundant bits do not add extra information. They are only used to check that the transmission is free of errors.

Algorithm

- The redundant bits are derived from an algorithm that is applied to the message. This algorithm is also known to the receiver.
- The sender transmits the message and the redundant bits.
- The receiver applies the same algorithm to the message and checks the redundant bits.
- If the transmitted and received redundant bits are the same, then it is assumed that the message was sent correctly. Otherwise, an error occurred.

Slide 107

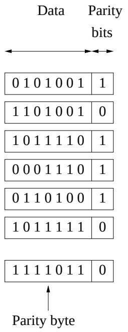
Two– dimensional parity: A simple error detection scheme

Example Even parity, odd parity and two–dimensional parity.

Consider a 7–bit code, and a redundancy bit:

- **Even parity** If the number of 1s in the code is odd, the redundant bit is set to 1 in order to make the total number of 1s in the byte an even number
- **Odd parity** If the number of 1s in the code is even, the redundant bit is set to 0
- **Two–dimensional parity** Apply these rules to each bit position across each of the bytes in the frame

Slide 108



Data	Parity bits
0 1 0 1 0 0 1	1
1 1 0 1 0 0 1	0
1 0 1 1 1 1 0	1
0 0 0 1 1 1 0	1
0 1 1 0 1 0 0	1
1 0 1 1 1 1 1	0
1 1 1 1 0 1 1	0

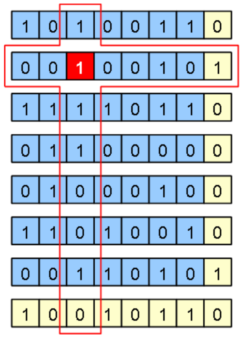
Even, odd and two–dimensional parity.

Note that 14 bits of redundant information have been added to a 42–bit message.

□

109

Error: If a single bit is corrupted, the parity check fails in both the row and the column



1	0	1	0	0	1	1	0
0	0	1	0	0	1	0	1
1	1	1	1	0	1	1	0
0	1	1	0	0	0	0	0
0	1	0	0	0	1	0	0
1	1	0	1	0	0	1	0
0	0	1	1	0	1	0	1
1	0	0	1	0	1	1	0

110

Checksum algorithms

Redundant bits are called checksum if the error detection algorithm is based on addition.

- Add all the transmitted data, where each word (=16 bits) is interpreted as an integer. The result is called the checksum.
- The receiver performs the same calculation on the received data and compares the result with the received checksum.
- If the two results are the same, it is assumed that the transmission is free of errors.

How is the addition performed ?

Use 16–bit 1's complement arithmetic for checksum calculations.

Slide 111

- A checksum is an error detection code based on adding
- A particular type of checksum is the Internet checksum which is computed as follows:
  - The checksum is made over a sequence of 16-bit integers using 1's complement arithmetic
  - For 1's complement, we use the following rules:

To negate a number, invert every bit (i.e., NOT).

To add two numbers, perform addition as normal,
but if there is a carry out, add one to the result.

Slide 112

1's complement arithmetic

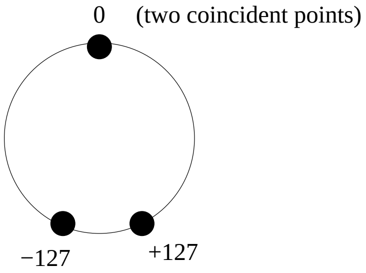
- All calculations are performed using numbers of a fixed length. Lengths of 8, 16 and 32 (powers of 2) bits are frequently used
- Numbers are represented along a circle, not a straight line.

Example Consider calculations performed on 8 bit integers.

Since $2^8 = 256$, and positive and negative integers must be considered, all integers in the range $-127 \ldots 127$, where $2^7 = 128$, can be represented. The first bit represents the sign

- The integer 0 is represented twice.

113



The 1s complement form of arithmetic requires that numbers are represented in a circle.

114

Example Positive numbers have a leading zero, and negative numbers have a leading one.

The binary representation of 3 in 4 bits is 0011

The binary representation of -3 in 4 bits is 1100

The binary representation of 5 in 4 bits is 0101

The binary representation of -5 in 4 bits is 1010

The binary representation of 76 in 8 bits is 01001100

The binary representation of -76 in 8 bits is 10110011

The binary representation of 127 in 8 bits is 01111111

The addition of 1 to 127 yields 10000000, which is -127 □

Slide 115

- Adding one is equivalent to going round the circle
- The system of numbers is cyclic

Example The 8 bit binary representation of -1 is 11111110, and the addition of 1 to this number yields

$11111110 + 00000001 = 11111111.$

Since the leading bit is 1, the number is negative, and so the absolute value of the number is obtained by flipping all the digits ($1 \longrightarrow 0$), which yields 0. □

Slide 116

Example 1's complement addition:

$11101010 + 11100011 = 1 \mid 11001101$

where the 1 on the left (msb) is the carry from the final addition. This 1 is the moved to the right (lsb) and standard addition is performed:

$11001101 + 00000001 = 11001110$ □

- The carry bit is moved to the right because of the cyclic nature of 1's complement addition

Example 1's complement addition:

$11111001 + 11001 = 1 \mid 00010010$

where the 1 on the left (msb) is the carry from the final addition. This 1 is the moved to the right (lsb) and standard addition is performed:

$00010010 + 00000001 = 00010011$ □

117

Example The 8 bit binary representation of 17 is 00010001 and the 8 bit binary representation of 11 is 00001011.

(a) Thus the binary representation of $-17-11$ is

$11101110 + 11110100 = 111100010$

which is a 9–bit integer.

We add the 9th digit (that is, the carry) to the 8–digit integer and obtain 11100011, which is the 1's complement of -17 and -11.

The complement of this number is 00011100, which is the binary representation of 28.

(b) The 1's complement of $17-11$ is

$00010001 + 11110100 = 1000000101$. Using 8–bit integers, the 1's complement is equal to $00000101 + 1 = 00110$, which is the binary representation of 6. □

118

Internet Checksum

- Keep adding 16 bit words together using 1's complement
- Negate the result and transmit as final word

```
1 0 0 0 0 1 1 0 1 0 0 0 0 1 1 0

0 1 0 1 0 1 1 0 1 1 1 1 0 0 1 0

1 0 0 0 1 0 0 0 1 0 0 0 0 1 1 1

1 0 0 1 0 1 1 0 1 0 1 0 0 0 1 0

0 0 0 0 0 0 1 1 0 1 0 1 1 1 0 1
```

- The sum over the entire block should be zero (all 1s)
- This scheme is easily implemented in hardware

Slide 119

Cyclic redundancy check (CRC)

Objective of error detection:

- Maximise the probability of detecting an error using only a small number of redundant bits

CRCs are a powerful method for error detection.

Example A 32–bit CRC gives strong protection against common bit errors in messages that are thousands of bits long. □

Question How do CRCs work ?

Answer Use the theory of polynomials

Slide 120

A message of length $(n+1)$ can be represented as a polynomial of degree n with coefficients of either 0 or 1.

Example The byte 10011010 corresponds to the polynomial

$$\begin{aligned} M(x) &= (1\times x^7)+(0\times x^6)+(0\times x^5)+(1\times x^4) \\ &\quad +(1\times x^3)+(0\times x^2)+(1\times x^1)+(0\times x^0) \\ &= x^7+x^4+x^3+x^1 \end{aligned}$$

□

It is convenient to think of the sender and receiver exchanging polynomials with each other.

121

Computing a CRC

1. The sender and receiver agree on a *divisor polynomial* $C(x)$ of degree k.
2. The sender transmits a message of length $(n+1)$ bits, plus k bits from $C(x)$. Denote this extended message $T(x)$. The divisor polynomial $C(x)$ is chosen so that it is an exact divisor of $T(x)$.
3. The receiver checks to determine if $C(x)$ is a divisor of $T(x)$ after transmission. If it is, it is assumed that an error has not occurred. Otherwise, a transmission error has occurred.

Important : The divisor polynomial has a significant impact on the type of errors that can be detected.

122

The CRC algorithm

Recall the objective of the CRC algorithm:

- Create a polynomial for transmission that is derived from the original message $M(x)$, is k bits longer than $M(x)$, and is exactly divisible by $C(x)$

Algorithm:

1. Multiply $M(x)$ by x^k, that is, add k zeros to the end of the message. This the zero–extended message is $T(x)$
2. Divide $T(x)$ by $C(x)$ and calculate the remainder
3. Subtract the remainder from $T(x)$ by performing an exclusive–OR (XOR) operation between $T(x)$ and the remainder

Slide 123

Rules for division

- Any polynomial $B(x)$ can be divided by a polynomial $C(x)$ if the degree of $B(x)$ is equal to or higher than the degree of $C(x)$
- The remainder from the subtraction of $C(x)$ from $B(x)$ is performed by performing an exclusive–OR operation on matching coefficients

Recall: The exclusive–OR of two logical variables p and q is

- *True* if p and q have different logical values
- *False* if p and q have the same logical values

Slide 124

Example Let

$$M(x) = x^7 + x^4 + x^3 + x \qquad \text{and} \qquad C(x) = x^3 + x^2 + 1$$

Thus

$$k = 3 \qquad \text{and} \qquad T(x) = x^{10} + x^7 + x^6 + x^4$$

The division $10011010000/1101$ yields a quotient of 11111001 and a remainder of 101. In terms of polynomials

$$\frac{x^{10} + x^7 + x^6 + x^4}{x^3 + x^2 + 1} = x^7 + x^6 + x^5 + x^4 + x^3 + 1 + \frac{x^2 + 1}{x^3 + x^2 + 1}$$

The transmitted bit stream is the XOR of

$$x^{10} + x^7 + x^6 + x^4 \qquad \text{and} \qquad x^2 + 1$$

that is,

$$x^{10} + x^7 + x^6 + x^4 + x^2 + 1$$

125

Its bit steam representation is

$$10011010101$$

which is the original message $M(x)$ appended by the remainder.

Note that

$$\frac{x^{10}+x^7+x^6+x^4+x^2+1}{x^3+x^2+1} = x^7+x^6+x^5+x^4+x^3+1$$

and a remainder of zero. □

It is simplest to divide the polynomials rather than the binary integers.

126

Why do CRCs work?

- Cyclic redundancy checks are powerful because it is unlikely that a polynomial will be divided exactly by the divisor $C(x)$
- The polynomial divisor is ‘designed’ to catch particular types of error, e.g.,
  - All 1 or 2 bit errors
  - Odd number bit of errors
  - Bursts (consecutive bits errors)

Slide 127

Divisor polynomials and error detection

If it is desired to transmit a message $P(x)$, then the presence of errors causes the message to be perturbed to $P(x) + E(x)$ where $E(x)$ represents the errors.

Question : How should the divisor polynomial $C(x)$ be chosen ?

Answer : Choose $C(x)$ so that it is not a divisor of $E(x)$ for the common types of error

Let $C(x)$ be of degree $k \geq 1$. Then

- If the coefficients of x^k and x^0 are non–zero, then all single bit errors can be detected
- If $C(x)$ has 3 or more terms, then all double bit errors can be detected
- A sequence of consecutive error bits can be detected if it is less than k bits long

Slide 128

Examples of CRC polynomials

CRC-8	x^8+x^2+x+1
CRC-10	$x^{10}+x^9+x^5+x^4+x+1$
CRC-12	$x^{12}+x^{11}+x^3+x^2+1$
CRC-16	$x^{16}+x^{15}+x^2+1$
CRC-CCITT	$x^{16}+x^{12}+x^5+1$
CRC-32	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}$ $+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

129

Error detection and error correction

It is advantageous to *detect* and *correct* errors. So what are the advantages and disadvantages of error correction against error correction and detection ?

- Error correction is mathematically more involved than error detection
- Error correction requires substantially more redundancy than does error detection
- Error correction does not require that corrupt frames be retransmitted
- If the number of transmission errors is small, then it is cheaper to resend a few messages than transmit the redundancy in error correction codes

130

Reliable transmission

Review

- We can turn a string of characters into a string of bits and imagine that images and sound, etc., can be similarly encoded
- We have looked at some encoding schemes for transmitting bits over a link... ...and some protocols for grouping the bits into frames
- We are now familiar with some algorithms for checking that no errors have been introduced into the frames at the receiver

What if errors have been introduced into a frame?

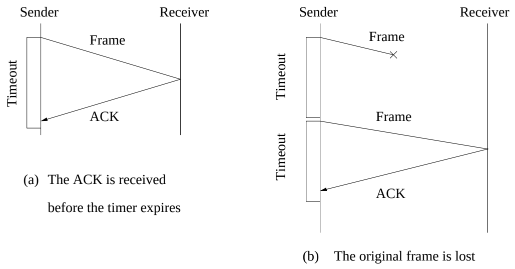
Slide 131

- Corrupted frames are generally discarded (lost) and must be retransmitted by the sender
- Protocols responsible for reliable (re-)transmission form part of the data link layer
- Our focus shall be automatic repeat request (ARQ) protocols:
  - Upon receiving a frame, the receiver sends an acknowledgement frame (ACK) to the sender
  - If the sender receives the ACK, it knows the frame was transmitted successfully
  - Otherwise, if a certain length of time elapses (the timeout) and the sender receives no ACK, the frame is sent again

This is the *stop and wait algorithm*, the simplest ARQ scheme

Slide 132

The stop and wait algorithm: Possible scenarios

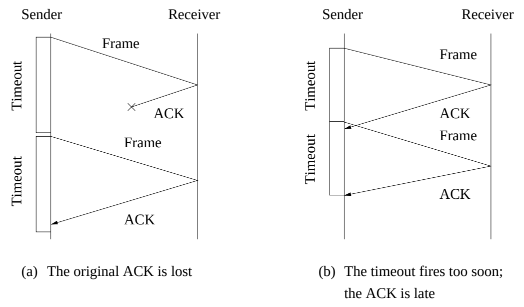


(a) The ACK is received before the timer expires

(b) The original frame is lost

Figure 26. Possible scenarios of the stop and wait algorithm. (a) Successful receipt and acknowledgement. (b) ACK is not received within timeout and the sender retransmits the frame.

133



(a) The original ACK is lost

(b) The timeout fires too soon; the ACK is late

Figure 27. More possible scenarios of the stop and wait algorithm. (a) ACK not received and thus the sender retransmits the frame. (b) The same frame is sent twice because the ACK is late.

134

Consider Figures 26 and 27.

- Figure 26(a): The ideal scenario
- Figure 26(b): The original frame is lost and thus an ACK is not received. The sender retransmits the frame and an ACK is received before the timer expires
- Figure 27(a): The frame is received by the receiver but the ACK is lost. The sender retransmits the frame, and an ACK is received before the timer expires
- Figure 27(b): The frame is received by the receiver but the ACK arrives at the receiver after the timeout. The sender retransmits the frame, and an ACK is received before the timer expires

Slide 135

Potential problem

Problem: Duplicate copies of a frame and ACK may be delivered.

Solution: Include a 1 bit sequence in the protocol's header, which alternates with each frame.

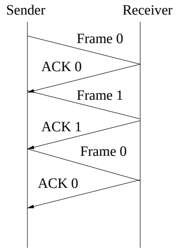


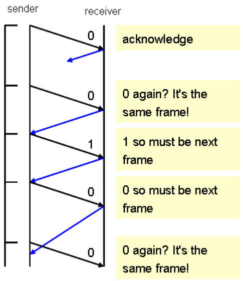
Figure 28: Timeout for stop and wait with 1 bit sequence.

Slide 136

In figures 27(a) and 27(b), the same frame is sent twice.

How does a 1 bit sequence solve this problem?

Since only one frame is allowed on the link at a time, the addition of a 1 bit sequence allows the receiver to deduce that the *same* frame was sent twice because they have the *same* bit appended to them.



137

Disadvantage of stop and wait algorithm

The sender is only allowed one frame on the link at a time, and this may be far below the link's capacity.

Example Consider a 1.5 Mbps link with a 45 ms RTT. The product of the delay and bandwidth is 67.5 Kb, or about 8 KB. Assuming a frame size of 1 KB, and since the sender can only send one frame per RTT, the maximum sending rate is

$$1024 \times 8/0.045 = 182 \text{ Kbps}$$

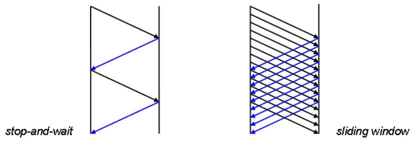
which is about $1/8$ of the capacity of the link. □

Conclusion: The product of the delay and bandwidth is the maximum capacity of the link. For reasons of efficiency, we wish to send this quantity of information before waiting for an acknowledgement. The aim is to keep the link full. This is achieved by the sliding window algorithm.

138

The sliding window algorithm

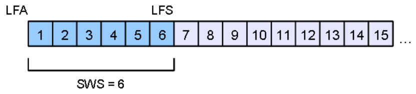
- With the stop-and-wait algorithm, the sender has to wait until an acknowledgement is received before it can transmit the next frame. It does not therefore exploit the capacity of the link - it does not 'keep the pipe full'.
- Solution: The sliding window algorithm transmits the next frames whilst waiting to acknowledge the earlier ones. We will (for now) require an infinite range of sequence numbers.



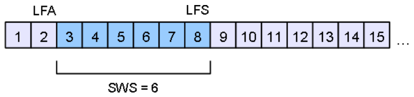
Slide 139

Sender sliding window

- The sender maintains a send window, which contains the sequence numbers of the sent frames awaiting acknowledgement



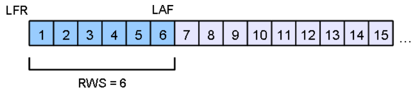
- Three variables: last frame acknowledged (LFA), last frame sent (LFS), and (maximum) send window size (SWS)
- If an acknowledgement is received for Frame X, then we set LFA to X and shift the window. For instance, if Frame 2 is acknowledged:



Slide 140

Receiver sliding window

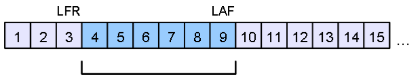
- The receiver also maintains a receiver window, which contains the sequence numbers of the received frames not yet acknowledged



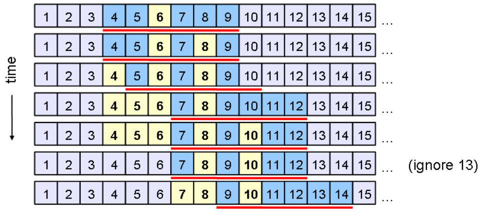
- The variables are: Last frame received (LFR), largest acceptable frame (LAF) and (maximum) receive window size (RWS)
- Receiver discards frames with sequence numbers outside the receive window
- Receiver acknowledges a received frame as soon as all frames in the window with lower sequence numbers have been received

141

Example If the receiver window has the state



and the frames are received in the order: 6, 8, 4, 5, 10, 13, 7, ...



Note that the minimum size of the sending window is 7 frames.

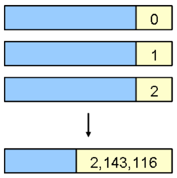
142

- At time $t = 1$, the number 6 arrives
- At time $t = 2$, the number 8 arrives but the window does not advance because the left hand digits (4 and 5) have not yet arrived
- At time $t = 3$, the number 4 arrives
- At time $t = 4$, the number 5 arrives
- At time $t = 5$, the numbers $4, 5, 6$ have arrived, and the window now occupies positions $7, 8, 9, 10, 11, 12$ because the left hand cells $(4, 5, 6)$ in the window have arrived
- At time $t = 6$, the number 13 arrives but this is ignored because it is outside the window, which is of length 6
- At time $t = 7$, the number 7 arrives and thus the window can advance to occupy cells 9 to 14 inclusive

Slide 143

Finite sequence numbers

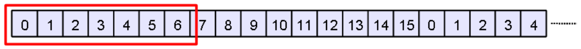
- We have assumed that sequence numbers can increase without an upper limit (i.e., infinite sequence numbers)
- As sequence numbers take up space in the header, clearly there is a limit to how many sequence numbers are allowed.



- For example, if four bits are reserved for the sequence number field in the header, what range of sequence numbers can be used?

Slide 144

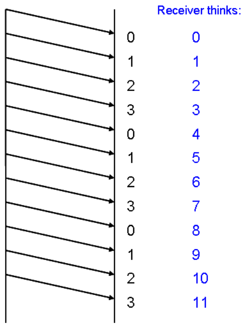
- The solution is to recycle the sequence numbers
- For example, if the sequence numbers range from 0 to 15, then the 16th frame will start over at 0, the 17th at 1, the 18th at 2, etc.



Is the problem solved?

145

An example of finite sequence numbers that range from 0 to 3.



146

Scenario 1:

- The send window size and receive window size are both four.
- The sequence numbers range from 0 to 3.
- The first four frames are all lost on the way.
- What happens?

Scenario 2:

- The send window size and receive window size are both four.
- The sequence numbers range from 0 to 3.
- All the frames arrive, but the ACKs for frames 0, 1, 2 and 3 are lost.
- What happens?

Slide 147

- If the sequence numbers range from 0 to 3, the receiver might think that the first frame is 0, 4, 8, 12, ..., depending on how many acknowledgements are lost.
- How do we overcome the problem of ambiguous sequence numbers?
- If the send window size and the receive window size are equal, SWS = RWS, then use at least twice as many sequence numbers as the send/receive window size.

The optimal range of sequence numbers is a function of the latency and bandwidth of the link

Slide 148

Ethernet

Review

- What terminology do we use to discuss networks? What do we need to consider when designing a network?
  - Links, nodes, switches, routers
  - Scalability, error detection, resource sharing, etc.
- How do we measure a network's performance?
  - Latency, bandwidth, throughput, packet size, queuing time.
- How do we turn everyday information such as images and text into bits (1's and 0's)?
  - Numbers encoded as ones and zeros e.g., 'B' is encoded as 66, or 01000010_2

149

- How do we transmit those bits along (e.g.) a cable?
  - NRZ or Manchester encoding
- How can we be sure the data has not been corrupted along the way?
  - Parity checks, Internet checksum, CRC
- What action do we take if it is corrupted?
  - ARQ protocols such as stop-and-wait and sliding window

150

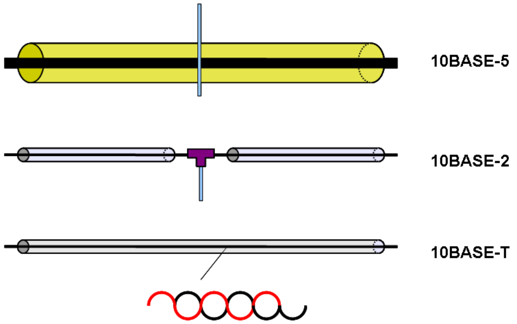
The Ethernet is an example of a CSMA/CD networking technology.

- Multiple–access (MA): Several nodes are connected to the same cable
- Carrier sense (CS): A node can distinguish between a busy and an idle link
- Collision detect (CD): A node listens when it transmits a frame in order to detect if a frame collides with a frame transmitted by another node
- CSMA / CD, like Ethernet, can be compared to a (polite!) conversation among a group of people in a room.
  - Multiple access: Everybody's speech is transmitted into the same space (so messages are public)
  - Carrier sense: People can hear when nobody else is speaking, so they know when they can speak themselves
  - Collision detection: People realise if they start speaking at the same time as somebody else.

Slide 151

Ethernet Cables

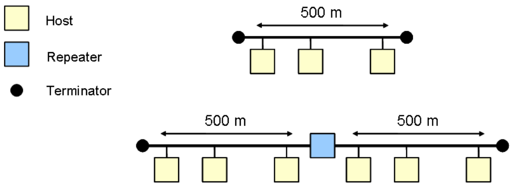
Historically up to the present day, there are a variety of Ethernet cables, which have been used.



Slide 152

Ethernet Topologies

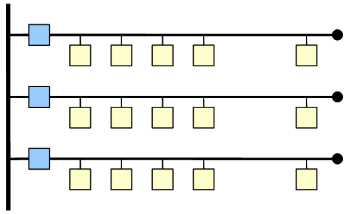
- A 10BASE5 and 10BASE2 segment can be up to 500 and 200 metres long, respectively.
- What happens if the network needs to span a larger distance?
- Repeaters are used to boost the signal over a long distance. Up to four repeaters can appear between any pair of hosts



153

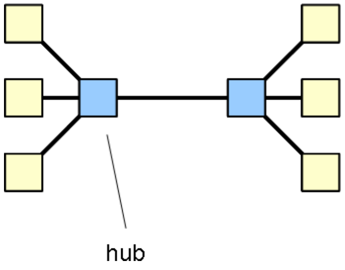
Ethernet Topologies

A multi-storey office might use a 10BASE5 backbone with repeaters attached on each floor.



154

Ethernets wired with 10BASE-T cable are linked together by hubs (multi-way repeaters) in star topologies.



Slide 155

Addressing

- How does an Ethernet node transmit to another node?
- Each Ethernet adaptor has a unique, hard-wired MAC (media access control) address, which is six bytes long.
- Typically notated as six hexadecimals separated by colons.

00001000 00000000 00101011 11100100 10110001 0000010
= 8 : 0 : 2b : e4 : b1 : 2

0000 = 0	0100 = 4	1000 = 8	1100 = c
0001 = 1	0101 = 5	1001 = 9	1101 = d
0010 = 2	0110 = 6	1010 = a	1110 = e
0011 = 3	0111 = 7	1011 = b	1111 = f

Slide 156

IEEE 802.3 Frame Format

- The IEEE 802.3 Ethernet frame format is as follows:

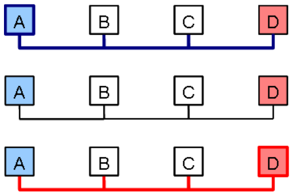
Preamble	Destination	Source	Type	Body	CRC
64 b	48 b	48 b	16 b	≥ 46 B, ≤ 1500 B	32 b

- The frame protocol is bit-oriented
- The preamble is a sequence of 64 alternating 0s and 1s to synchronise network adaptors
- The destination and source are 6 byte (48 bit) MAC addresses
- The type field records the type or length of the body
- A 32-bit CRC is transmitted finally

157

Transmission

- How do we share the link between multiple nodes?
- Each network adaptor that is ready to transmit a frame:
  - Waits until the line is free (carrier sense) then transmits a frame
  - Each frame must not exceed 1,500 bytes (limited transmit time)
- What happens if two hosts transmit at the same time?



A transmitting, B and C idle, D waiting to transmit.

Link no longer occupied.

D senses line is free, and transmits.

158

The transmitter algorithm

When an adaptor has a frame to send and

- the line is idle, it transmits the frame
- the line is busy, it transmits the frame immediately after the line goes idle – not necessarily a good strategy

If more than one adaptor transmits a frame at the same time, then the frames collide:

- Each sender can detect independently if a collision occurred. (An Ethernet LAN does not have a centralised control.)
- Upon detecting a collision, a jamming sequence is sent. This sequence, in effect, instructs all the other nodes on the network to cease transmission.

Slide 159

Exponential backoff

Once an adaptor has detected a collision, it retransmits the frame according to the following rule:

- 1st attempt: It delays either 0μs or 51.2μs, selected at random
- 2nd attempt: If the 1st attempt fails, it then waits 0μs, 51.2μs, 102.4μs or 153.6μs, selected at random before it tries again, that is, it waits $k \times 51.2\mu$s, $k = 0, \ldots, 3$, selected at random
- 3rd attempt: If the 2nd attempt fails, it then waits $k \times 51.2\mu$s, $k = 0, \ldots, 2^3-1$, selected at random, before it tries again
- nth attempt: If the $(n-1)$th attempt fails, it then waits $k \times 51.2\mu$s, $k = 0, \ldots, 2^n - 1$, selected at random, before it tries again

Ethernets typically use 16 retransmission attempts before reporting a transmit error to the host.

Slide 160

Example

Let A and B be two hosts that are attempting to transmit on an Ethernet. Each host has an unlimited supply of frames, labelled $A_1, A_2, \ldots,$ and $B_1, B_2, \ldots$ Describe the exponential backoff algorithm. Assume $T = 51.2\ \mu$s.

Answer

1. **1st attempt:** A and B each attempt to send a frame but the frames collide. Each hosts waits either $0T$ or $1T$. Note that $1 = 2^1 - 1$. Assume that A wins and so A_1 is transmitted.
2. **2nd attempt:** A tries to transmit A_2 and B tries to transmit B_1. Assume that they collide. Since A won the first race, its backoff time remains randomly chosen from $(0T, 1T)$. Since, however, B lost, its backoff times are $(0, \ldots 2^2-1)T = (0T, 1T, 2T, 3T)$. Note that there is a higher probability of A transmitting A_2 than of B transmitting B_1. Assume, however, that B wins the race and therefore transmits B_1.

161

3. **3rd attempt:** A tries to transmit A_2 and B tries to transmit B_2, and assume that the frames collide. Since A lost the second race, its backoff times are now chosen from $(0T, 1T, 2T, 3T)$. Since B won the second race, its backoff times remain $(0T, 1T, 2T, 3T)$, that is, the same as those of A.

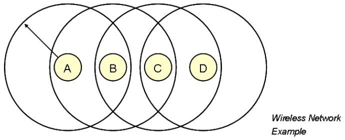
□

162

Wireless networks

Wireless networks are

- rapidly becoming the technology for connecting computers
- designed for use in a limited geographic area
- Another multiple access technology
- Unlike the Ethernet, wireless nodes may not be able to reach each other, e.g., A and D cannot see each other



Slide 163

Collision avoidance

Wireless networks employ similar techniques to the Ethernet for collision avoidance – wait till the link becomes idle before transmitting, and back off if a collision occurs.

A problem of *hidden nodes* can occur with wireless networks because not all nodes are within reach of each other:

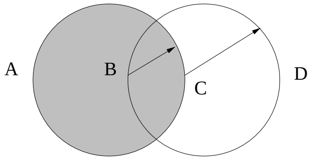


Figure 29: Example wireless network.

Slide 164

Consider the 4 node wireless network in Figure 29.

- Each node is able to send and receive messages that just reach the nodes to its immediate left and right:
  - A can exchange frames with B
  - B can exchange frames with A and C, but not with D
  - C can exchange frames with B and D, but not with A
  - D can exchange frames with C
- A and C are unaware of each other
- If A and C each send a frame to B, then the frames collide at B
- Unlike an Ethernet, A and C are not aware of the collision
- A and C are called *hidden nodes* with respect to each other

165

Consider again the 4 node wireless network in Figure 29.

- Suppose that B is sending to A
- C is aware of this transmission because it is within the required range
- But C still can still transmit to D because it does not interfere with B's transmission to A
- Note that it would interfere with A transmitting to B, but B is transmitting to A in this example

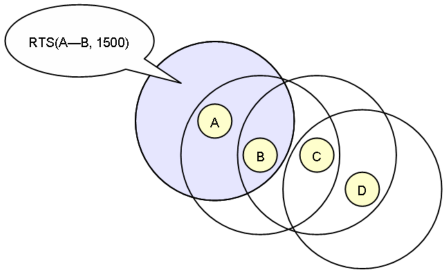
166

Collision avoidance

- IEEE 802.11 wireless uses multiple access collision avoidance (MACA)
- Collision avoidance prevents collisions by 'reserving' a transmission slot between sender and receiver
- The sender initiates a transmission by a request to send (RTS) frame, which indicates the size of the transmission
- The receiver responds with a clear to send (CTS) frame, which also contains the size of the transmission:
  - Any node which sees the CTS frame knows to remain silent
  - Any node which sees only the RTS frame is free to transmit

Slide 167

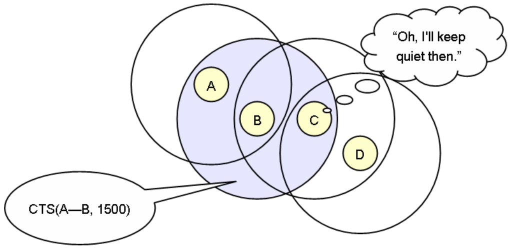
Example A wants to transmit to B, C wants to transmit to D, B wants to reply to A.



A requests a transmission to B.

Slide 168

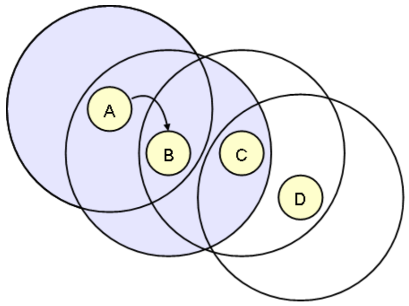
Example cont'd



B replies to A, both A and C receive the reply. C doesn't transmit.

169

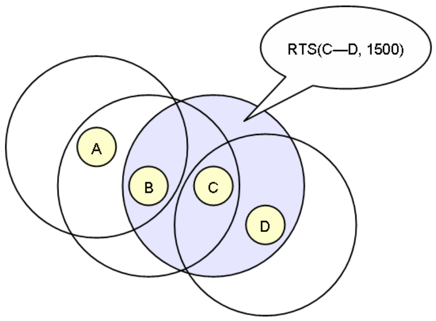
Example cont'd



For 1500 bytes, A transmits to B whilst C remains silent.

170

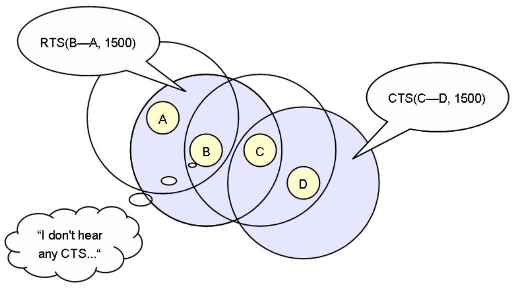
Example cont'd



After 1500 bytes, C can begin transmitting to D.

Slide 171

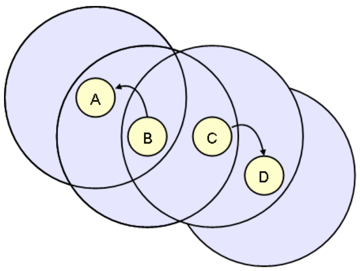
Example cont'd



D replies with a CTS, but B doesn't hear it, so is free to transmit.

Slide 172

Example cont'd



B can safely transmit to A, whilst C transmits to D.

□

173

Some extra points

What about acknowledgements (ACKs)?

- Frames waiting to transmit also wait for an ACK frame from the receiver before doing so

What if two RTS frames collide?

- If a node transmits an RTS and doesn't receive a CTS, it concludes (after a timeout) that there was a collision and starts exponential back-off

174

Distribution system

The system described so far is adequate for an ad–hoc network; communication between any two nodes is governed by their distance apart.

But one of the advantages of a wireless network is that nodes are free to move around. So there are two classes of node:

- Some nodes are allowed to roam, eg., a laptop computer, PDA and mobile phone
- Some nodes, called *access points*, are connected to each other by a wired network infrastructure called a distribution system. This may be an Ethernet or token ring.

Slide 175

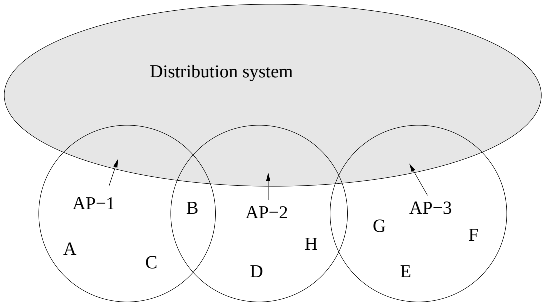


Figure 30: Access points contributed to a distribution network. Nodes A...F are allowed to roam.

Slide 176

- Each access point (AP) services the node in a particular region:
  - Each of these regions is analagous to a cell in a mobile telephone network, and each AP plays the same role as a base station
- Two nodes can communicate with each other directly if they are within reach of each other
- If they cannot communicate with each other directly, the distribution system is used:
  - If A wants to communicate with E, it sends a frame to its access point AP-1, which forwards the frame across the distribution system to AP-3, which transmits the frame to E

177

Selecting access points

The procedure for selecting an AP is called *scanning* and involves four steps:

- The node sends a Probe frame
- All APs within reach reply with a Probe Response frame
- The node selects one of these APs and sends the AP an Association Request frame
- The AP replies with an Association Response frame

This procedure is invoked when

- a node joins the network
- an existing node is unhappy with its current AP, for example, when the signal is weak

and it is called *active scanning* because the node is actively looking for an AP.

178

Packet switching

Networks in which each host is connected to every other host by a physical medium (that is, directly connected networks) have been considered thus far but they have two limitations:

1. There is a limit on the number of hosts that can be attached: 2 hosts for a point–to–point link, and up to 1024 hosts for an Ethernet
2. There is a limit on the size of the geographical area that a single network can serve: An Ethernet link can only span 2500 m, and a point–to–point link does not serve the area between the ends

Question: Since it is desired to build a global network, how can we create a network between hosts that are not connected directly?

Slide 179

Packet switches

Functions of a switch:

- A switch accepts packets that arrive at the input and forwards (or switches) them to the correct outputs so that each packet will arrive at its correct destination
- The input and output sockets of a switch are called *ports*

Challenges in switching

- Every link has finite bandwidth:
  - If packets destined for a specified output arrive at a switch such that their arrival rate exceeds the capacity of the output, then *contention* arises and packets need to be stored in buffers
  - The switch stores the packets until contention subsides, but *congestion* may arise if the buffer space is insufficient and some packets must be discarded
- The determination of the correct output port for each packet

Slide 180

Switching, forwarding and scalability

What is a switch?

A switch is a mechanism that allows several links to be connected together to form a larger network

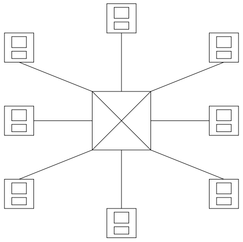


Figure 31: A switch provides a star topology.

181

- A switch is a multi–input multi–output device that transfers packets from an input to one or more outputs. This leads to a *star topology*

A star topology has several advantages:

- A switch has a fixed number of inputs and outputs, which limits the number of hosts that can be connected to each switch, but
- switches can be connected to each other and to hosts using point–to–point links, and thus networks over a large geographical area can be built up
- The addition of a new host to a network does not necessarily lead to inferior performance of the hosts already connected to the network, but *switches must have enough capacity for the new host*

Summary: Switches support scalability

182

How does a switch decide on which output port to place each packet?

The information is in an identifier in the header of the packet.

There are three different methods in which it uses the identifier:

- Connectionless (datagram)
- Connection–oriented (virtual circuit)
- Source routing

Assumptions used in these methods:

- Each node in the network is given a unique address, for example, an Ethernet address
- Each input and output port of a switch are given a unique number, relative to its switch

Slide 183

The connectionless (datagram) method

Simple idea: Provide every packet with enough information that enables a switch to transmit the packet to its correct destination.

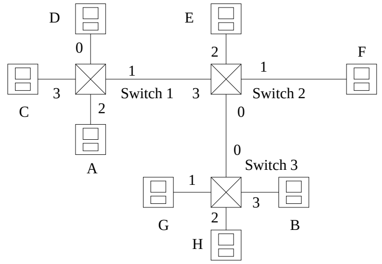


Figure 32: Datagram forwarding for three switches and hosts A to H. The numbers 0,...,3 refer to the ports of the switch.

Slide 184

Use a *routing (forwarding)* table to to decide how to forward a packet.

Destination	A	B	C	D	E	F	G	H
Port	3	0	3	3	2	1	0	0

Forwarding table for switch 2 in Figure 32.

- A routing table is easy for a small and simple network
- They are harder to build for larger, more complex networks that evolve dynamically and have multiple paths between destinations

185

Advantages of datagram (connectionless) networks:

- A host can send a packet anywhere at any time because a switch can forward a packet as soon as it arrives (assuming a correctly populated forwarding table).
- The failure of a switch or link may not have serious consequences if there exists a physical link round the failure. This may require modifying the forwarding table.

Disadvantages of datagram (connectionless) networks:

- A sending host does not know if the network is capable of delivering a packet, or if the destination host is alive.
- Each packet is forwarded independently of other packets. Thus two successive packets from host A to host B may follow completely different paths, and therefore arrive out of order.

186

The datagram is analogous to posting a letter.

- A letter is a sealed entity and the route to its destination exists before the letter is sent:
  - Each letter is treated as an independent entity that is not connected to other letters
  - Each letter is received and forwarded by each node (sorting office) along the defined route

This method is best for short single–packet messages

Slide 187

The connection–oriented method

Analogy: When a telephone call is made, the connection is established before communication takes place. This corresponds to the connection–oriented method:

Compare with the datagram.

Sequence of operations in the connection–oriented method:

- The sending host sends a special *call–request* packet to its switch
- This packet contains the address of the required destination host, and a reference number called the *virtual circuit identifier* (VCI), which is noted by the switch
- The packet is forwarded through the network (*The method used will be explained later*)
- A VCI is attached at each switch along the route, before the packet is forwarded to the receiving host

Slide 188

- If the call–request packet is accepted, a response packet is returned to the sending host
- A virtual circuit between the source and receiving hosts has now been established
- Data transfer now occurs, and all subsequent data packets relating to this call are assigned the same reference numbers on each link in the network
- This enables the source and receiving hosts to distinguish between packets arriving on the same link, but originating from different calls

Notes:

- The VCIs are stored in a VCI table
- All packets in a message travel along the same path in the network
- This method is used for messages that contain multiple packets. Compare with the datagram.

189

A virtual circuit (connection state) can be set up in two ways:

- *Permanent virtual ciruits* (PVCs):
  - These are set up (configured) by the administrator
  - They can be deleted by the administrator
- *Temporary (dynamic) virtual circuits*:
  - A host sends messages into the network, and these cause the connection state to be established
  - This is called *signalling*, and the resulting virtual circuits are called *switched virtual circuits* (SVCs). They are more popular than PVCs.
  - An SVC can be deleted dynamically without the involvement of the network administrator

Note: A virtual circuit is usually cleared and all the channel identifiers are released after the data has been exchanged.

190

Example It is required to create a connection from host A to host B in order to send a message.

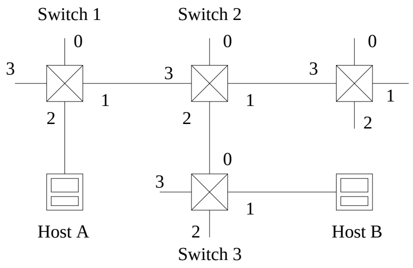


Figure 33: An example of a virtual circuit network. The numbers 0,...3 refer to the ports of the switch.

Slide 191

Consider Figure 33, in which host A wants to send packets to host B. A virtual link between these hosts A and B is to be created manually.

Note: There is only one link from A to B, but there are, in general, multiple paths between any two hosts.

1. The administrator picks a VCI value that is currently unused for each link in the connection
2. Let the VCI value for the link from host A to switch 1 be 5
3. Let the VCI value for the link from switch 1 to switch 2 be 11

Similarly

1. Let the VCI value for the link from switch 2 to switch 3 be 7
2. Let the VCI value for the link from switch 3 to host B be 4

Slide 192

Incoming port	Incoming VCI	Outgoing port	Outgoing VCI
2	5	1	11

Virtual circuit table for switch 1 in Figure 33.

Incoming port	Incoming VCI	Outgoing port	Outgoing VCI
3	11	2	7

Virtual circuit table for switch 2 in Figure 33.

Incoming port	Incoming VCI	Outgoing port	Outgoing VCI
0	7	1	4

Virtual circuit table for switch 3 in Figure 33.

The ‘outgoing’ VCI value at one switch is the ‘incoming’ VCI value at the next switch.

193

Once the VC tables have been set up, data transfer occurs:

1. A puts the value 5 in the header of the packet to be transmitted, and sends it to switch 1
2. Switch 1 receives the packet on port 2
3. It now uses its VC table in order to forward the packet and to put 11 in its header when the packet is sent
4. The packet arrives at port 3, with VCI 11, at switch 2
5. Switch 2 uses its VCI table in order to send the packet to switch 3, after updating the header on the packet to 7
6. The process continues until the packet arrives at host B with VCI value 4 in its header
7. This VCI value informs host B that the packet arrived from host A

□

194

Advantages and disadvantages of virtual circuit switching

Advantages:

- Each packet need only include a small identifier (the VCI) in its header, rather than the full address of the receiver. This reduces the overhead caused by the headers
- When a virtual circuit is established, the sender knows that
  - There is a route to the receiver
  - The receiver is willing and able to receive data
  - There are enough resources along the route

Disadvantages:

- There is at least 1 RTT before any data is sent
- A switch/link failure leads to a broken connection
- Released/broken virtual circuits must be destroyed

Slide 195

Source routing

Another approach to switching, in addition to the datagram and virtual circuit method.

- All information about network topology that is required to switch a packet across a network is provided by the source host – and hence the name *source routing*

One method of implementation:

- Assign a number to each output of each switch and place the number in the header of the packet
- For each packet that arrives on an input, the switch reads the port number in the header and transmits the packet on that output
- Since there may be mutiple paths between the sending and receiving hosts, the switch must contain enough information to determine the output port of every switch on which the packet must be placed
- Use an ordered list of switch ports, and before forwarding a packet,

Slide 196

- *either* strip the number from the packet,
- *or* rotate the ordered list such that the next port number is at the front

Notes:

- Every host must know details of the network's topology in order to build up a packet header:
  - Similar to the problem of building up a forwarding (routing) table in the datagram, and routing a call–request packet in a virtual circuit network
  - Suffers from a scaling problem because it is difficult to get complete routing information in large networks
- Source routing is used in
  - the Internet protocol
  - virtual–circuit networks for routing the call–request packet

197

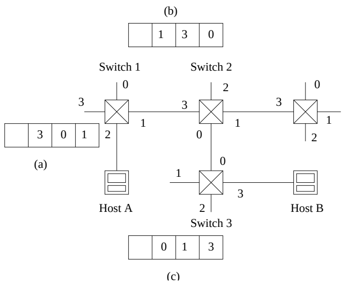


Figure 34: Source routing in a switched network for transmission of a message from host A to host B: (a), (b) and (c) are the ordered list of switch ports for switches 1, 2 and 3 respectively. The numbers 0,...3 refer to the ports of the switch.

198

Switching and congestion

Congestion is dealt with differently in virtual circuits and datagrams:

- Virtual circuits do not suffer from congestion because a switch prevents connection from being established if it cannot dedicate enough resources (in terms of bandwidth and buffer space) to the virtual circuit
- Datagram networks do not avoid congestion and must therefore be able to recover from congestion

Slide 199

Bridges and LAN switches

Thus far, we have concentrated on the basic ideas of switching, but we now consider some specific switching technologies.

Two types:

- LAN switches:
  - Used to forward packets between shared media LANs , for example, Ethernets
- Learning bridges:
  - Used to connect several LANs

Slide 200

LAN switches

Requirement: It is desired to connect a pair of Ethernets. Potential solutions include:

- Insert a repeater between them:
  - Problem: An Ethernet can have a maximum of two repeaters between any two hosts, and a maximum length of 2.5 km
- Insert a node between them:
  - This node would transmit all frames on both Ethernets
  - A node is a multi–input multi–output device that transfers packets from one input to one or more outputs – the defintion of a switch
  - This node is called a *bridge*
  - A collection of LANs connected by one or more bridges is called an *extended LAN*

201

Learning bridges

How can a bridge be optimised for efficiency?

- Not all frames that it receives need be forwarded

Example Consider the learning bridge in Figure 35.

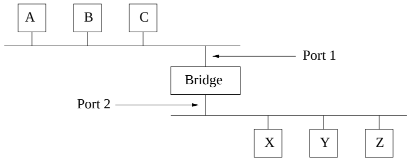


Figure 35: A learning bridge

202

Question: A frame from host A that is addressed to host B arrives on port 1. How does the bridge know that the frame need not be transmitted?

One solution: Construct the following table. The bridge consults this table and checks the destination address on the packet in order to decide its response:

Host	A	B	C	X	Y	Z
Port	1	1	1	2	2	2

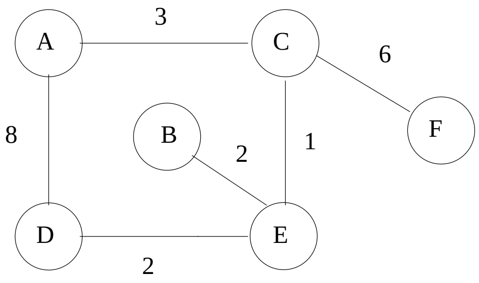
Note: This is the datagram (connectionless) model of forwarding a packet.

The manual maintenance of this table is burdensome, but there is a method that enables the bridge to learn the information itself:

- Each bridge inspects the source address of each *incoming* frame
- The pair (port number, host) is used to build up the table shown above

Slide 203

Example Consider the network below, in which each link is labelled with a relative cost. Construct the datagram for the network. The entries in the tables should show the path of lowest cost to the destination.



Slide 204

Node A		Node B		Node C	
Terminus	Next host	Terminus	Next host	Terminus	Next host
B	C	A	E	A	A
C	C	C	E	B	E
D	C	D	E	D	E
E	C	E	E	E	E
F	C	F	E	F	F

205

Node D		Node E		Node F	
Terminus	Next host	Terminus	Next host	Terminus	Next host
A	E	A	C	A	C
B	E	B	B	B	C
C	E	C	C	C	C
E	E	D	D	D	C
F	E	F	C	E	C

□

206

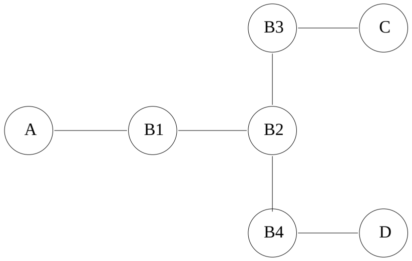
Learning algorithm

- The forwarding table is empty when the bridge first boots
- Entries are filled in over time
- If the *destination* address does not have an entry in the table, the frame is forwarded to all the other ports of the bridge
- A timeout is associated with each entry, and the bridge then discards the entry, to protect against the situation in which a host moves from one network to another network
- The table is updated dynamically

Slide 207

Example Consider the learning bridges shown below. Assuming that all of them are initially empty, give the forwarding tables for each of the bridges B1-B4 after the following transmissions:

(i) A sends to C, (ii) C sends to A, and (iii) D sends to C.



A, C and D are hosts. B1, B2 and B3 are bridges.

Slide 208

(i) When A sends to C, all the bridges see the packet and and thus they all learn where A is.

(ii) When C sends to A, the packet is routed directly, that is, B4 is not traversed. This follows from part (i).

(iii) When D sends to C, the packet is routed directly to B3, and B1 does not learn where D is.

These results can be summarised as follows (i/f denotes interface):

B1:		A–i/f:	A	B2–i/f:	C (not D)		
B2:		B1–i/f:	A	B3–i/f:	C	B4–i/f:	D
B3:		B2–i/f:	A, D	C–i/f:	C		
B4:		B2–i/f:	A (not C)	D–i/f:	D		

209

Repeat the above table when

(i) D sends to C, (ii) C sends to D, and (iii) A sends to C.

Assume that all the learning bridges are empty.

(i) All the bridges see the packet from D to C.

(ii) Only B2, B3 and B4 see the packet from C to D.

(iii) Only B1, B2 and B3 see the packet from A to C.

B1:	A–i/f:	A	B2–i/f:	D (not C)		
B2:	B1–i/f:	A	B3–i/f:	C	B4–i/f:	D
B3:	B2–i/f:	A, D	C–i/f:	C		
B4:	B2–i/f:	C (not A)	D–i/f:	D		

□

210

Loops in extended LANs

The preceding optimisation algorithm works well *if the extended LAN does not have a loop*

Reason: Frames can now (theoretically) loop through the extended LAN indefinitely

- A loop implies that there exists more than one path between two hosts

But

- Loops provide redundancy if links and/or bridges fail

Slide 211

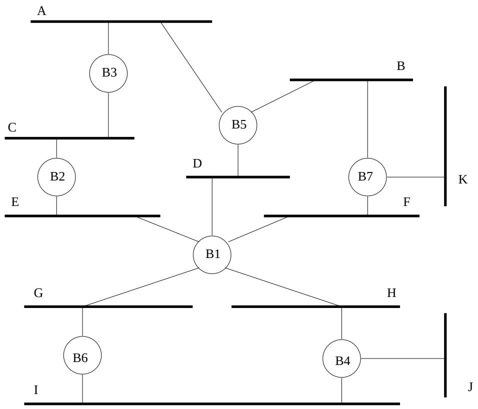


Figure 36: Extended LAN loops. Bridges B1, B4 and B6 form a loop.

Slide 212

Spanning trees and the spanning tree algorithm

How can bridges cope with loops in order to avoid indefinite looping?

Use a *distributed spanning tree* algorithm:

- Think of an extended LAN as a graph in which the bridges act as vertices
- A loop in an extended LAN is represented by a cycle in the graph
- The *spanning tree* of graph is a subgraph of this graph that retains all the vertices but removes all the cycles, that is, some of the edges are removed

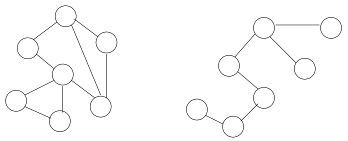


Figure 37: A cyclic graph (left) and its spanning tree (right).

213

Example The application of the spanning tree algorithm to the extended LAN in Figure 35 yields the network shown in Figure 38. □

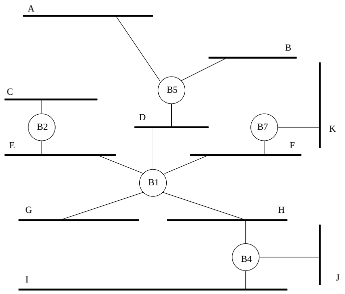


Figure 38. The spanning tree of the extended LAN shown in Figure 36. Note that the bridges B3 and B6 have been removed, and that a port from B7 has been removed.

214

Limitations of bridges

The connection of a large number (more than a 'few tens') of LANs is not practical for several reasons:

- The spanning tree algorithm scales linearly
- Bridges forward all broadcast frames:
  - This is reasonable in a limited setting, eg., a department in an office, but it is not reasonable in a larger setting, eg., the whole company
  - Thus broadcast does not scale, and hence extended LANs do not scale
- Bridges can only connect networks that use the same format in their headers, eg., Ethernet to Ethernet, and Ethernet to token rings
- If a bridge becomes congested, it may need to delete frames:
  - Compare with an Ethernet, in which this only occurs rarely

Slide 215

- The large scope of an extended LAN implies that the latency becomes larger and more highly variable:
  - Compare with the limited physical scope of an Ethernet

BUT: Bridges play an important role in networking

Slide 216

Summary

Networks considered so far:

- A single network using point–to–point, shared media, and switches

But communication is required across networks and new problems arise:

- **Heterogeneity:** Users of *different* networks must be able to communicate:
  - What happens if communication of two networks requires that a third network be traversed?
- **Scaling:** The internet has approximately doubled in size every 20 years. This forces the problem of *routing* and *forwarding*:
  - **Routing:** The construction of an efficient path through a network of millions, or even billions, of nodes
  - **Addressing:** The provision of an identifier for each node in the network

217

Routing

Assumption so far:

- Switches and routers have enough knowledge of the network topology to choose the correct port to which each packet should be sent:
  - Routing is an issue for every packet in datagram networks
  - Routing is an issue only for the call–request packet in virtual circuits

Distinguish between *forwarding* and *routing*:

- Forwarding: The process of determining, for each packet, its destination address, consulting a table, and sending the packet in the direction determined by the table
- Routing: The process of construction of a forwarding table

218

Distinguish between the *forwarding table* and the *routing table*:

- Forwarding table: This is used when a packet is being forwarded and must contain enough information to accomplish the forwarding function
- Routing table: A table that is built up by the routing algorithms in order to construct the forwarding table

Question: How is this routing information acquired?

The size of the internet requires that routing algorithms work for networks of millions or billions of nodes:

Algorithms that only work for networks of fewer than a few hundred nodes will be considered

But

They provide the building blocks for a hierarchical routing infrastructure that is used in the internet

Slide 219

Routing, scalability and graph theory

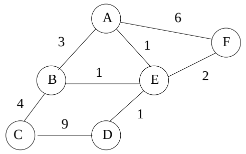


Figure 39. Network representation as a graph.

- The nodes of the graph represent hosts, routers, switches or networks
- The edges of the graph correspond to the network links
- Each edge has a *cost*, which is a measure of the desirability, or otherwise, of sending traffic over that link

Problem: Determine the lowest cost path between any two nodes, where the cost of a path is equal to the sum of the costs of all the edges that make up the path.

Slide 220

For the network shown in Figure 38, we can use a table that defines the path of lowest cost between any two nodes. This is an example of a *static algorithm*, but it has disadvantages:

- It does not deal with node or link failures
- It does not consider the addition or removal of nodes
- The cost of a link cannot be changed

Question: Why would we want to change the cost of a link?

Answer: If a link is very busy, then we may want to limit the traffic that can use it. This can be achieved by imposing a high cost on the link.

Note that a static algorithm is an example of a *centralised* algorithm, but it is not easy to scale a centralised algorithm.

221

Use a *distributed* algorithm to overcome the problems of static algorithms:

- It provides a dynamic method of determining the path of lowest cost in the presence of link and node failures, and changing edge costs

What is a distributed algorithm?

Rather than consider a *centralised* algorithm, a *distributed* algorithm divides the problem of optimal path selection into a large number of smaller problems.

Does this approach have problems?

Example Consider two routers, each of which forwards a packet to the other because each thinks that the other has a shorter path to the destination. The packet will be stuck in an infinite loop until the discrepancy is solved. □

222

Distance–vector (Bellman–Ford) routing algorithm

- Each node constructs a vector of costs to all other nodes
- Set the cost of the transfer to its immediate neighbours to the known values
- Set the cost to all other nodes to infinity
- The vector is distributed to all the immediate neighbours

Example Consider the network shown below.

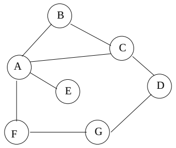


Figure 40. The cost is the same for each link and it has therefore been omitted.

Slide 223

Node		Distance to reach node						
		A	B	C	D	E	F	G
A		0	1	1	∞	1	1	∞
B		1	0	1	∞	∞	∞	∞
C		1	1	0	1	∞	∞	∞
D		∞	∞	1	0	∞	∞	1
E		1	∞	∞	∞	0	∞	∞
F		1	∞	∞	∞	∞	0	1
G		∞	∞	∞	1	∞	1	0

Figure 41: The initial distances stored at each node. A cost of 1 implies that the packet can move between the two nodes in one step, but a cost of infinity implies that the two nodes do not see each other.

Slide 224

Each row defines a vector of a node. For example, the vector of node A is:

Destination	B	C	D	E	F	G
Cost	1	1	∞	1	1	∞
Next hop	B	C	–	E	F	–

Next step in distance–vector routing algorithm: Every node sends a message to its directly connected neighbours containing its list of distances.

Example:

- F informs A that it can reach G at a cost of 1
- A also knows that it can reach F at a cost of 1
- These two costs are added together:
  - The cost of sending a packet from A to G via F is 2
- This value is less than the cost of infinity in Figure 40, and this new value is recorded

225

Similarly

- A cannot reach D directly, but C (which can be reached from A) informs A that it can reach D at a cost of 1
- This cost is added to the cost of A transmitting to C, so that the cost of A transmitting to D via C is 2
- This value is better than the infinite value in Figure 40, and this new value is recorded

Also

- A learns from C that B can be reached from C at a cost of 1
- The cost of the route A–C–B is 2, which is more than the current cost of the link A–B, and thus this new information is ignored

226

This information enables the routing table of A to be updated:

Destination	B	C	D	E	F	G
Cost	1	1	2	1	1	2
Next hop	B	C	C	E	F	F

□

Note:

- At each stage, we only make the minimum number of changes required:
  - Only the unreachable (infinite cost) entries are updated
- This method is applied to all nodes, and *convergence* is reached when the algorithm has finished

Slide 227

Example continued: The final set of costs for each node to every other node is shown in the table below.

		Distance to reach node						
Node		A	B	C	D	E	F	G
A		0	1	1	2	1	1	2
B		1	0	1	2	2	2	3
C		1	1	0	1	2	2	2
D		2	2	1	0	3	2	1
E		1	2	2	3	0	2	3
F		1	2	2	2	2	0	1
G		2	3	2	1	3	1	0

Figure 42: The final distances stored at each node.

Slide 228

Updating mechanisms

Two types: Periodic updating, and triggered updating:

- *Periodic updating:* Each node automatically sends an update message periodically, even if there have not been changes:
  - This informs the other nodes that the sending node is still active
  - This information may be required if a current route from a node becomes unviable
  - The updates are once every several seconds, to once every several minutes
- *Triggered updating:* This occurs when a node receives an update from one of its neighbours that causes it to change one of the routes in its routing table

229

The failure of a link or node

How is a failed link or node detected? Two methods:

- A node continually tests the link by sending a control packet and noting if an acknowledgement is received
- A node determines if a link, or the node at the other end of the link, is down if the expected periodic routing packet is not received for several cycles

What happens when a node or link fails?

- The nodes that first notice the failure send new lists of distances to their neighbours (some distances are set to infinity)
- The algorithm stabilises fairly quickly to a new steady state

230

Example Consider the sequence of events that occur when F detects that its link to G in Figure 40 has failed.

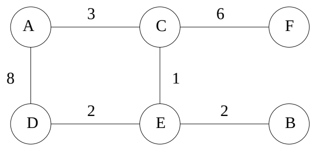
- F sets its new distance to G to infinity and passes this information to A
- Since A knows that its 2-hop path to G is through F, A sets its distance to G to infinity
- At the next update of the algorithm, A learns that C has a 2-hop path to G, and so A has a 3-hop path to G via C
- The routing table entry for A–G is changed from infinity to 3
- This information is learnt by F, which now learns that it can reach G in 4 hops
- The routing table entry for F–G is changed to 4

□

Note: If the timings are slightly incorrect, the algorithm may not stabilise. This is called the count–to–infinity problem, and there exist solutions to it.

Slide 231

Example Consider the network shown below.



Calculate the global distance–vector tables when:

1. Each node knows the distance only to its immediate neighbours
2. Each node has reported the information it had in the preceding step to its immediate neighbours
3. Step (2) is repeated

What happens if nodes B and F are removed?

Slide 232

Part 1:

	Distance to reach node					
Node	A	B	C	D	E	F
A	0	∞	3	8	∞	∞
B	∞	0	∞	∞	2	∞
C	3	∞	0	∞	1	6
D	8	∞	∞	0	2	∞
E	∞	2	1	2	0	∞
F	∞	∞	6	∞	∞	0

233

Part 2:

Node		Distance to reach node					
		A	B	C	D	E	F
A		0	∞	3	8	4	9
B		∞	0	3	4	2	∞
C		3	3	0	3	1	6
D		8	4	3	0	2	∞
E		4	2	1	2	0	7
F		9	∞	6	∞	7	0

234

Part 3:

Node		Distance to reach node					
		A	B	C	D	E	F
A		0	6	3	6	4	9
B		6	0	3	4	2	9
C		3	3	0	3	1	6
D		6	4	3	0	2	9
E		4	2	1	2	0	7
F		9	9	6	9	7	0

Note that the cost of the link A–D decreases from 8 to 6 from part 2 to part 3.

□

Slide 235

Example Nodes A and F of a network have the forwarding tables shown below. If the cost of all the links is one, calculate the smallest network that is consistent with the tables.

A

Node	Cost	Next hop
B	1	B
C	2	B
D	1	D
E	2	B
F	3	D

F

Node	Cost	Next hop
A	3	E
B	2	C
C	1	C
D	2	E
E	1	E

Slide 236

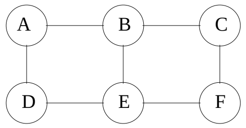
From the table for A:

1. A is connected directly to B and D
2. There are links A–B–C and A–B–E

For the table for F:

1. F is connected directly to C and E
2. There is a link F–C–B and F–E–D

The smallest network is shown in the diagram below.



□

237

Example Consider the example on slides 238–241. Assume that the forward tables have been established as shown in the example, and the link C–E then fails.

1. Give the tables for A, B, D and F after C and E have reported the news
2. Give the tables for A and D after their mutual exchange
3. Give the table for C after A exchanges information with it

Part 1:

A:

Dest'n	Cost	Next hop
B	∞	–
C	3	C
D	∞	–
E	∞	–
F	9	C

B:

Dest'n	Cost	Next hop
A	∞	–
C	∞	–
D	4	E
E	2	E
F	∞	–

238

Part 1 cont'd:

D:

Dest'n	Cost	Next hop
A	∞	–
B	4	E
C	∞	–
E	2	E
F	∞	–

F:

Dest'n	Cost	Next hop
A	9	C
B	∞	–
C	6	C
D	∞	–
E	∞	–

Slide 239

Part 2:

1. The table for D shows that the cost of the link to E is 2
2. The table for B shows that the cost of the link to E is 2
3. A is connected directly to D at a cost of 8
4. Thus the cost of the link A–B is 12

A:

Dest'n	Cost	Next hop
B	12	D
C	3	C
D	8	D
E	10	D
F	9	C

D:

Dest'n	Cost	Next hop
A	8	A
B	4	E
C	11	A
E	2	E
F	17	A

Slide 240

Part 3:

The table for C is:

Dest'n	Cost	Next hop
A	3	A
B	15	A
D	11	A
E	13	A
F	6	F

□