14 Communication and internet technologies

In this chapter, you will learn about

- the need for protocols during communication
- the implementation of protocols such as a stack
- TCP/IP protocols, including the four layers (Application, Transport, Internet and Link), the purpose and function of the four layers, and application when a message is sent from one host to another on the internet
- HTTP, FTP, POP3/4, IMAP, SMTP and BitTorrent protocols (such as BitTorrent provides peer-to-peer file sharing)
- circuit switching (including benefits and drawbacks)
- the benefits and drawbacks of packet switching
- the function of a router in packet switching
- the use of packet switching to pass messages across the network (including the internet).

14.1 Protocols

WHAT YOU SHOULD ALREADY KNOW

In Chapter 2, you learned about networks. Try these five questions before you read the first part of this chapter.

1 **a)** Explain the terms *IP* and *TCP*.

b) What are *protocols* and why are they used?

2 **a)** Explain how peer-to-peer networks operate.

b) What are the pros and cons of peer-to-peer networks?

3 **a)** Explain the term *stack*.

b) Explain the term *queue*.

c) Give examples of the use of a stack and the use of a queue.

4 **a)** Explain what is meant by *Ethernet*.

b) What is meant by *IP conflicts* when using Ethernet?
Explain how they can be overcome.

5 **a)** What is a *DNS*?

b) What is meant by *HTTP*?

c) Explain the role of *status flags*.

Key terms

Protocol – a set of rules governing communication across a network: the rules are agreed by both sender and recipient.

HTTP – hypertext transfer protocol.

Packet – a message/data is split up into smaller groups of bits for transmission over a network.

Segment (transport layer) – this is a unit of data (packet) associated with the transport layer protocols.

FTP – file transfer protocol.

SMTP – simple mail transfer protocol.

Push protocol – protocol used when sending emails, in which the client opens the connection to the server and keeps the connection active all the time, then uploads new emails to the server.

Binary file – a file that does not contain text only. The file is machine-readable but not human-readable.

MIME – multi-purpose internet mail extension. A protocol that allows email attachments containing media files as well as text to be sent.

POP – post office protocol.

IMAP – internet message access protocol.

TCP – transmission control protocol.

Pull protocol – protocol used when receiving emails, in which the client periodically connects to a server, checks for and downloads new emails from a server and then closes the connection.

Host-to-host – a protocol used by TCP when communicating between two devices.

Host – a computer or device that can communicate with other computers or devices on a network.

BitTorrent – protocol used in peer-to-peer networks when sharing files between peers.

Peer – a client who is part of a peer-to-peer network/file sharing community.

Metadata – a set of data that describes and gives information about other data.

Pieces – splitting up of a file when using peer-to-peer file sharing.

Tracker – central server that stores details of all other computers in the swarm.

Swarm – connected peers (clients) that share a torrent/tracker.

Seed – a peer that has downloaded a file (or pieces of a file) and has then made it available to other peers in the swarm.

Leech – a peer with negative feedback from swarm members.

Lurker – user/client that downloads files but does not supply any new content to the community.

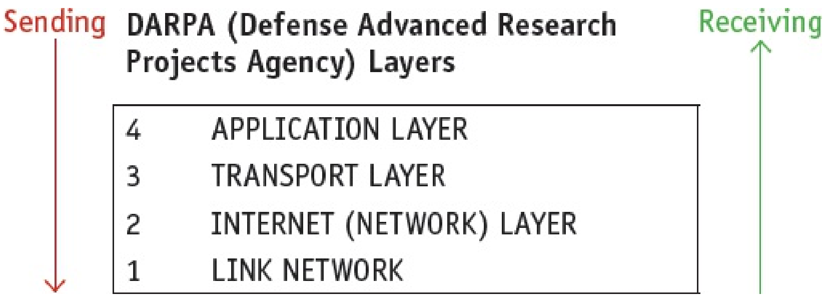
14.1.1 The need for protocols

When communicating over networks, it is essential that some form of **protocol** is used by the sender and receiver of the data. Both parties need to agree the protocol being used to ensure successful communication takes place. In Chapter 6, we discussed parity checking as a way of determining whether data was transmitted correctly. With this method, it was essential to agree the protocol: even or odd parity. Without agreeing this protocol, it would be impossible to use parity checking. Many different protocols exist since there are several activities taking place over the internet.

The next section considers one of the most common sets of protocols, which are implemented by using a stack structure with several layers.

14.1.2 TCP/IP protocols

This is the four-layer structure for TCP/IP protocols:



DARPA (Defense Advanced Research Projects Agency) Layers

4	APPLICATION LAYER
3	TRANSPORT LAYER
2	INTERNET (NETWORK) LAYER
1	LINK NETWORK

Figure 14.1 Four layer structure for TCP/IP

Using layers breaks the process down into manageable self-contained modules (this process is known as **decomposition**), making it easier to develop and easier to make software and hardware compatible.

When sending data across the internet (network), the layers are used in the order layer 4 to layer 1; when receiving data across the internet (network), the layers are used in the order layer 1 to layer 4. Each of the layers is implemented using software.

Application layer

The application layer contains all the programs that exchange data, such as web browsers or server software; it sends files to the transport layer. This layer allows applications to access the services used in other layers and also defines the protocols that any app uses to allow the exchange of data.

There are several protocols associated with the application layer:

HTTP	hypertext transfer protocol; this is a protocol responsible for correct transfer of files that make up web pages on the world wide web
SMTP	simple mail transfer protocol; this handles the sending of emails
POP3/4	post office protocol; this handles the receiving of emails
IMAP	internet message access protocol; this handles the receiving of emails
DNS	domain name service; protocol used to find the IP address, for example, when sending emails
FTP	file transfer protocol; this is a protocol used when transferring messages and attachments
RIP	routing information protocol; this is the protocol routers use to exchange routing information over an IP network
SNMP	simple network management protocol; protocol used when exchanging network

	management information between network management and network devices (such as routers, servers and other network devices)

Table 14.1 Protocols associated with the application layer

It is worth re-visiting the terms *packet* and *router*.

Messages are split up into small groups of bits called *packets* (for example, a web page would be split up into a number of packets before sending over the network).

A router is used to transmit packets of data; routers contain connections to many other routers; when packets arrive at a router it decides where next to send them.

Important terminology: packets are known as *frames* at the data-link layer, *datagrams* at the internet layer and **segments** at the transport layer. Different names are used as each layer adds its own header to the packet.

Do not confuse ‘frames’ in this context with ‘frames’ when discussing paging memory management in Chapter 16.

Hypertext transfer protocol (HTTP)

HTTP is probably the most important application layer protocol. Essentially, this protocol underpins the world wide web. It is used when, for example, fetching an HTML document from a web server (Figure 14.2).

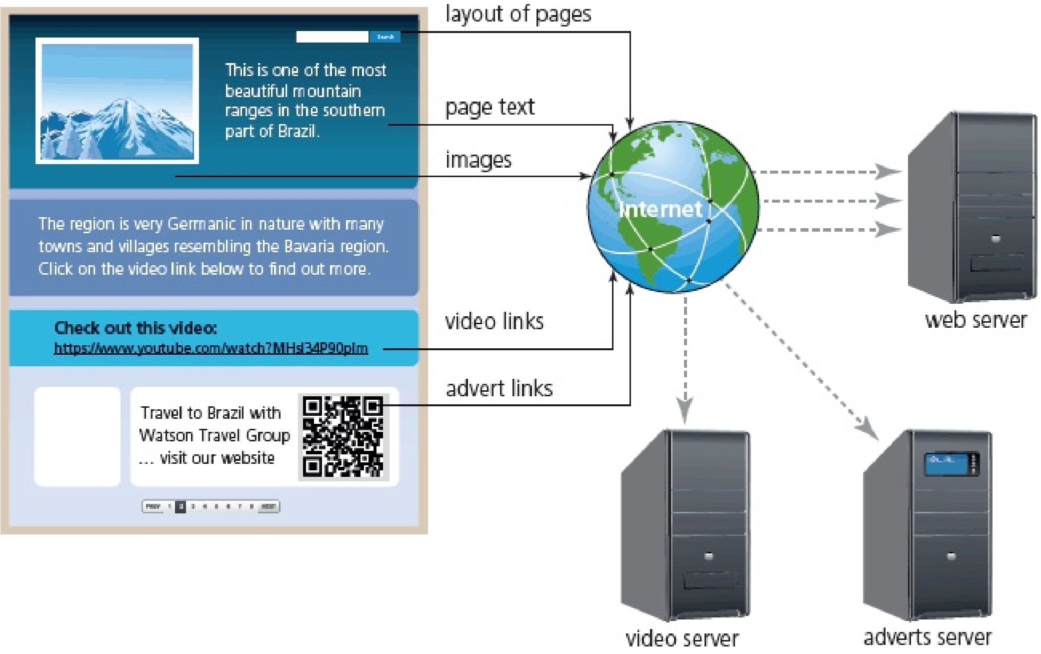

This is one of the most beautiful mountain ranges in the southern part of Brazil.
The region is very Germanic in nature with many towns and villages resembling the Bavaria region. Click on the video link below to find out more.
Check out this video:
https://www.youtube.com/watch?MHsl34P90plm
Travel to Brazil with Watson Travel Group ... visit our website
layout of pages
page text
images
video links
advert links
internet
web server
video server
adverts server

Figure 14.2 Fetching an HTML document from a web server

This makes use of hyperlinks (rules for the transferring of data over the internet). HTTP is a client/server protocol: request messages are sent out to the web servers which then respond.

HTTP protocols define the format of the messages sent and received. The web browser (which is part of the application layer) initiates the web page request and also converts HTML into a format which can be displayed on the user's screen or can be played through their media player.

The following summarises what happens when a user requests a web page from a website.

- The user keys the URL into their browser.
- HTTP(s) transmits the request from the application layer to the transport layer (TCP).
- The TCP creates data packets and sends them (via port 80) to the destination port(s).
- The DNS server stores a database of URLs and matching IP addresses.
- The DNS server uses the domain name typed into the browser to look up the IP address of the appropriate website.
- The server TCP sends back an acknowledgement (see the section on host-to-host communication on page **333**).
- Once communication has been established, the web server sends the web page back in HTML format to the browser.
- The browser interprets the page and displays it or sends the data in the correct format to the media player.

File transfer protocol (FTP)

The **FTP file transfer protocol (FTP)** is a network protocol used when transferring files from one computer/device to another via the internet or other networks. It is similar to HTTP and SMTP, but FTP's only task is the application protocol for the transfer of files over a network. Web browsers can be used to connect to an FTP address in a way similar to HTTP, for example, ftp://username@ftp.example.gov/

Additional features of FTP include

- **anonymous ftp** – this allows a user to access files without the need to identify who they are to the ftp server; for example, '331 Anonymous access allowed' would be a message received to confirm anonymous access
- **ftp commands** – a user is able to carry out actions that can change files stored on the ftp server; for example, delete, close, rename, cd (change directory on a remote machine), lcd (change directory on a local machine)
- **ftp server** – this is where the files, which can be downloaded as required by a user, are stored.

A session would be started by typing in the ftp host_name (of remote system), followed by a user id and password. The user would then be able to use ftp commands to carry out a number of actions.

Simple mail transfer protocol (SMTP)

Simple mail transfer protocol (SMTP) is a text-based (and connection-based) protocol used when sending emails. It is sometimes referred to as a **push protocol** (in other words, a client opens a connection to a server and keeps the connection active all the time; the client then uploads a new email to the server).

Since SMTP is text-based only, it doesn't handle **binary files** (a binary file is a file containing media/images as well as text and is regarded as being computer-readable only). If an email contains attachments made up of, for example, images, video, music then it is necessary to use the **multi-purpose internet mail extension (MIME)** protocol instead. A MIME header is used at the beginning of the transmission; clients use this header to select which media player is needed when the attachment is opened.

POP3/4 and IMAP (post office protocol and internet message access protocol)

POP Post office protocol (POP3/4) and **internet message access protocol (IMAP)** are protocols used when receiving emails from the email server. These are known as **pull protocols** (the client periodically connects to a server; checks for and downloads new emails from the server – the connection is then closed; this process is repeated to ensure the client is updated). IMAP is a more a recent protocol than POP3/4, but both have really been superseded by the increasing use of HTTP protocols. However, SMTP is still used when transferring emails between email servers.

Figures 14.3 and 14.4 give an overall view and a more detailed view of the email protocol set up.

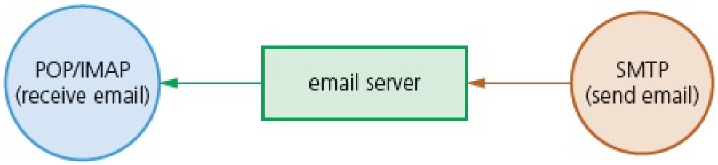


Figure 14.3 Overview of email protocol set up

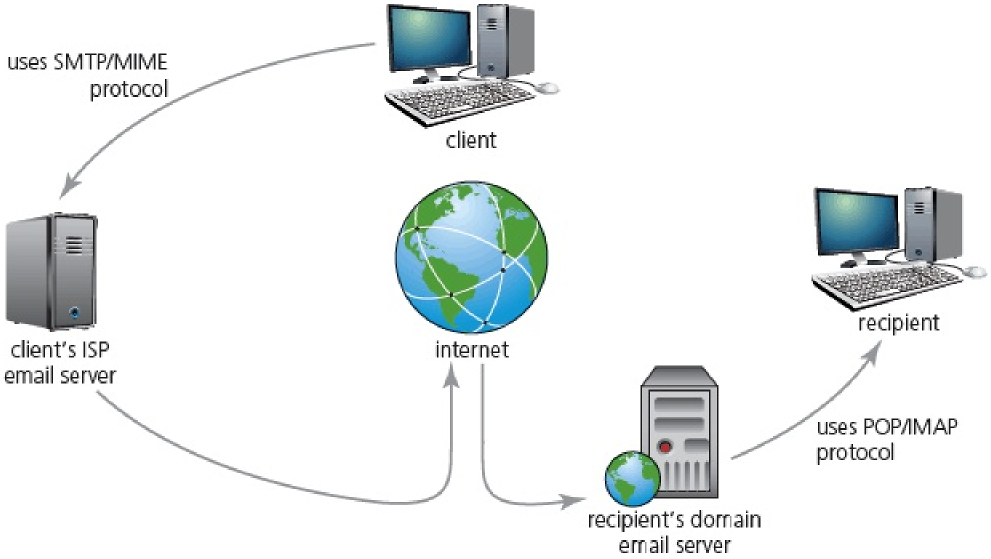


Figure 14.4 Detailed view of email protocol set up

The main difference between POP3/4 and IMAP is synchronisation:

POP3/4	IMAP
POP3/4 does not keep the server and client in synchronisation; when emails are downloaded by the client, they are then deleted from the server which means it is not further updated.	IMAP keeps the server and client in synchronisation; only a copy of the email is downloaded with the original remaining on the server until the client manually deletes it.

Table 14.2

Transport layer

The transport layer regulates the network connections; this is where data is broken up into packets which are then sent to the internet/network layer (IP protocol). The transport layer ensures that packets arrive in sequence, without errors, by swapping acknowledgements and retransmitting packets if they become lost or corrupted. The main protocols associated with the transport layer are **transmission control protocol (TCP)**, user datagram protocol (UDP) and SCTP. We will only consider TCP.

Transmission control protocol (TCP)

TCP is responsible for the safe delivery of a message by creating sufficient packets for transmission. It uses positive acknowledgement with re-transmission (PAR) which means it automatically re-sends a data packet if it has not received a positive acknowledgement. TCP is also connection-orientated since it establishes an end-to-end connection between two host computers using handshakes. For this last reason, TCP is often referred to as a **host-to-host** transmission protocol.

The term **host** has been used previously; this refers to a computer or device that can communicate with another computer/device (host). Hosts can include clients and servers that send/receive data, provide services or apps.

These are the steps taken when host computer ‘X’ communicates with another host ‘Y’ (this is an expansion of what happens during TCP involvement shown in the HTTP algorithm earlier on):

- Host ‘X’ will first of all send host ‘Y’ a segment (packet) which will include synchronisation sequence bits so that segments will be received in the correct order.
- Host ‘Y’ will now respond by sending back its own segment (containing an acknowledgement together with its own synchronisation sequence bits).
- Host ‘X’ now sends out its own acknowledgement that the segment from ‘Y’ was received.
- Transmission of data between ‘X’ and ‘Y’ can now take place.

Internet/network layer and network/data-link layer

The internet layer identifies the intended network and host. The common protocol is IP (internet protocol). The concept of IPv4 and IPv6 was covered in depth in Chapter 2.

The network/data-link layer identifies and moves traffic across local segments, encapsulates IP packets into frames for transmission, maps IP addresses to MAC (physical) addresses and ensures correct protocols are followed. The physical network layer specifies requirements of the hardware to be used for the network. The data-link layer identifies network protocols in the packet header (TCP/IP in the case here) and delivers packets to the network.

This is a summary of the IP functions:

- Ensure correct routing of packets of data over the internet/network.
- Responsible for protocols when communicating between networks.
- Take a packet from the transport layer and add its own header which will include the IP addresses of both sender and recipient.
- The IP packet (datagram) is sent to the data-link layer where it is assembles the datagrams into

frames for transmission.

Ethernet protocols

Ethernet is a system that connects a number of computers or devices together to form a LAN. It uses protocols to control the movement of frames between computers or devices and to avoid simultaneous transmission by two or more devices. It is a local protocol and does not provide any means to communicate with external devices; this requires the use of IP which sits on top of the Ethernet protocol.

Figure 14.5 shows the make-up of a typical frame used by the Ethernet protocol.

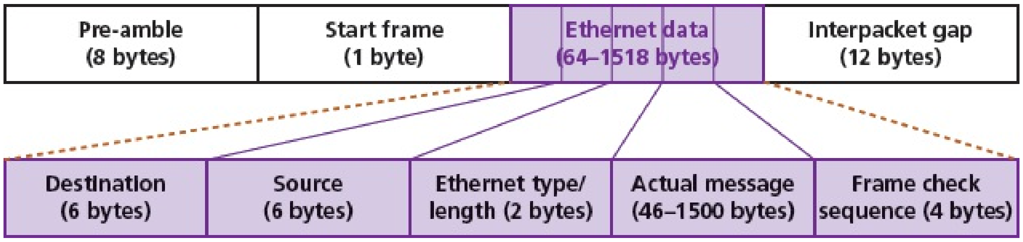


Figure 14.5 A typical frame used by the Ethernet protocol

If VLAN is used, the Ethernet data size increases from 1539 bytes to around 9000 bytes per frame.

The components that make up Ethernet data are

- **destination** – this is the MAC address of the destination computer or device (it is possible to use the value FF:FF:FF:FF:FF:FF as the MAC address if the sender wishes to target every device (for example, to advertise services) or if they do not know the MAC address of the destination device)
- **source** – this is the MAC address of the source computer (using the usual MAC address format of 6 bytes)
- **Ethernet type or length** – if the frame length ≤ 1539 then the value here is the length of the Ethernet frame; if the frame length > 1539 then the value here is the Ethernet type (IPv4 or IPv6 in our example)
- **frame check** – this will include a checksum to provide a method of checking data integrity following transmission of the frame.

Wireless (WiFi) protocols

Wireless LANs (standard IEEE 802.11 protocol) use a MAC protocol called carrier sense multiple access with collision avoidance (CSMA/CA) (not to be confused with CSMA/CD considered in Chapter 2, since this is a totally different concept).

CSMA/CA uses distributed control function (DCF) to ensure a WiFi device can only transmit when there is a free channel available. Since all transmissions are acknowledged when using DCF, if a device does not receive an acknowledgement it will assume a collision will occur and waits for a random time interval before trying again. This is an important protocol to ensure the security and integrity of data being sent over a wireless network (such as WLAN).

Bluetooth protocols

Bluetooth was considered in Chapter 2; it uses the standard IEEE 802.15 protocol for short-range data transmission/communication. There are numerous additional Bluetooth protocols due to the many applications that may use this wireless connectivity; this is outside the scope of this textbook.

WiMax

Worldwide interoperability for microwave access (WiMax) runs under IEEE 802.16 protocol. This connectivity was designed originally for wireless MANs (WMAN). Fixed WiMax networks are based on the IEEE 802.16-2004 protocol, whereas mobile WiMax is based on IEEE 802.16-2005 protocol.

Peer-to-peer file sharing/BitTorrent protocol

The **BitTorrent** is a protocol which is based on the peer-to-peer networking concept (this was covered in Chapter 2). This allows for very fast sharing of files between computers (known as **peers**). While peer-to-peer networks only work well with very small numbers of computers, the concept of sharing files using BitTorrent can be used by thousands of users who connect together over the internet. Because user computers are sharing files directly with each other (rather than using a web server) they are sharing files in a way similar to that used in a peer-to-peer network; the main difference is that the BitTorrent protocol allows many computers (acting as peers) to share files.

Suppose computer ‘A’ wishes to share a file with a number of other interested peers. How can we use the BitTorrent protocol to allow this file sharing?

Initially, to share a file, the peer (computer ‘A’) creates a small file called a torrent (for example, MyVideoFile.torrent). The torrent contains **metadata** about the file about to be shared.

The actual file is broken up into equal segments known as **pieces** (typically a 20 MiB file may be broken up into 20 × 1 MiB pieces).

Other peers who wish to download this file must first obtain the torrent and connect to the appropriate **tracker** – a central server that contains data about all of the computers connected to it.

As each peer receives a piece of file they then become a source for that piece of file. Other peers connected to the tracker will, therefore, know where to find the piece of file they need.

Once a peer has downloaded a file completely and they make the file (or required pieces of the file) available to other peers in the **swarm** (a group of peers connected together), they become a **seed**. The more seeds in the swarm, the faster the file downloading process between peers.

Logging off once the full file download has been completed is frowned upon by the swarm community; such a peer is termed a **leech**.

Usually, once a file is fully downloaded, a peer is requested to remain online so they can become part of the seeding process until all peers have received the whole file. Note that file pieces may not be downloaded sequentially and have to be rearranged in the correct order by the BitTorrent protocol to produce the final file (quite important if the file is a video!).

At the time of writing, BitTorrent was responsible for about 12% of the video file sharing, for example, being carried out over the internet. This is only a fraction of the video file activity which uses YouTube (which is about 50% of all of the video file sharing over the internet), but is still a considerable amount of data.

Here is a summary of some of the terms used when discussing BitTorrent:

- **Swarm** – a group of peers connected together is known as a swarm; one of the most important facts when considering whether or not a swarm can continue to allow peers to complete a torrent is its availability; availability refers to the number of complete copies of torrent contents that are distributed amongst a swarm. Note: a torrent is simply the name given to a file being shared on the peer-to-peer network.
- **Seed** – a peer that has downloaded a file (or pieces of a file) and has then made it available to other peers in the swarm.
- **Tracker** – this is a central server that stores details about other computers that make up the swarm; it will store details about all the peers downloading or uploading the file, allowing the peers to locate each other using the stored IP addresses.
- **Leech** – a peer that has a negative impact on the swarm by having a poor share ratio, that is, they are downloading much more data than they are uploading to the others; the ratio is determined using the formula:

  $$\text{ratio} = \frac{\text{amount of data the peer has uploaded}}{\text{amount of data the peer has downloaded}}$$

  If the ratio > 1 then the peer has a positive impact on the swarm; if the ratio < 1 then the peer has a negative effect on the swarm.
- **Lurker** – a peer that downloads many files but does not make available any new content for the community as a whole.

In Figure 14.6, we will assume 12 peers have connected to the tracker. One peer has begun to upload a video file and six peers are acting as seeds. Two peers are behaving as leeches and three peers have just joined and have requested a download of the video file. The arrangement would look something like this:

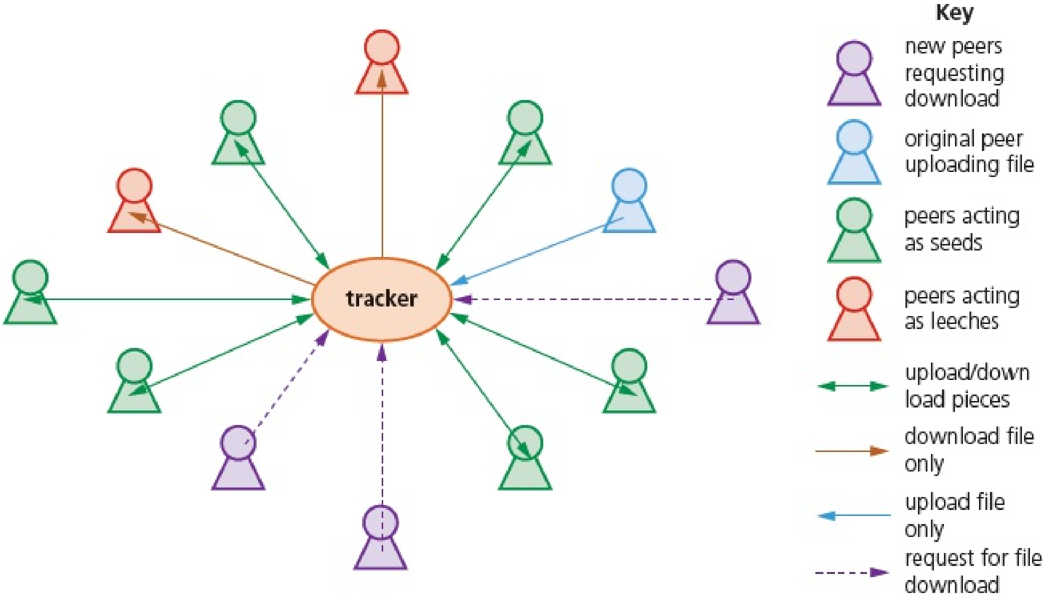


Figure 14.6 The arrangement of peers during the download and upload of file (pieces)

14.2 Circuit switching and packet switching

WHAT YOU SHOULD ALREADY KNOW

In Chapter 2, you learnt about communications. Try these two questions before you read the second part of this chapter.

1 **a)** Explain how PSTN is used when making a phone call.

b) Explain how it is possible to use VoIP to make a video call over the internet.

2 Draw a diagram to show how a message containing three packets of data could be routed from computer 'A' to computer 'B'.

Key terms

Circuit switching – method of transmission in which a dedicated circuit/channel lasts throughout the duration of the communication.

Packet switching – method of transmission where a message is broken into packets which can be sent along paths independently from each other.

Hop number/hopping – number in the packet header used to stop packets which never reach their destination from 'clogging up' routes.

Header (data packet) – part of a data packet containing key data such as destination IP address, sequence number, and so on.

Routing table – a data table that contains the information necessary to forward a package along the shortest or best route to allow it to reach its destination.

14.2.1 Circuit switching

The concept of **circuit switching** was introduced in Chapter 2 during the description of how a public switched telephone network (PSTN) was used to make a phone call. Circuit switching uses a dedicated channel/circuit which lasts throughout the connection: the communication line is effectively ‘tied up’. When sending data across a network, there are three stages:

1. First, a circuit/channel between sender and receiver must be established.
2. Data transfer then takes place (which can be analogue or digital); transmission is usually bi-directional.
3. After the data transfer is complete, the connection is terminated.

The pros and cons of circuit switching are summarised in this table. Figure 14.7 shows an example of circuit switching.

Pros	Cons
the circuit used is dedicated to the single transmission only	it is not very flexible (for example, it will send empty frames and it has to use a single, dedicated line)
the whole of the bandwidth is available	nobody else can use the circuit/channel even when it is idle
the data transfer rate is faster than with packet switching	the circuit is always there whether or not it is used
the packets of data (frames) arrive at the destination in the same order as they were sent	if there is a failure/fault on the dedicated line, there is no alternative routing available
a packet of data cannot get lost since all packets follow on in sequence along the same single route	dedicated channels require a greater bandwidth
it works better than packet switching in real-time applications	prior to actual transmission, the time required to establish a link can be long

Table 14.3 Pros and cons of circuit switching

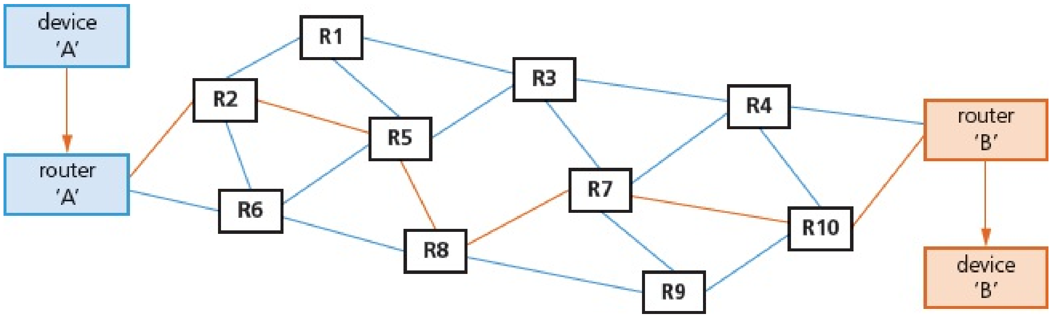


Figure 14.7 An example of circuit switching

The dedicated route from ‘A’ to ‘B’ is first of all established (shown in orange on the diagram). The following connections are then partially implemented: A–R2, R2–R5, R5–R8, R8–R7, R7–R10 and finally R10–B. All packets (frames) follow this single route and communication will take place, provided ‘B’ is not busy.

The main uses of circuit switching include public telephone networks, private telephone networks and private data networks.

14.2.2 Packet switching

Packet switching was introduced in Chapter 2 when describing VoIP, together with a diagram to show how the individual packets are routed from client to client.

Packet switching is a method of transmission in which a message is broken up into a number of packets that can be sent independently to each other from start point to end point. The data packets will need to be reassembled into their correct order at the destination. Figure 14.8 shows an example of packet switching.

Note that

- each packet follows its own path
- routing selection depends on the number of datagram packets waiting to be processed at each node (router)
- the shortest path available is selected
- packets can reach the destination in a different order to that in which they are sent.

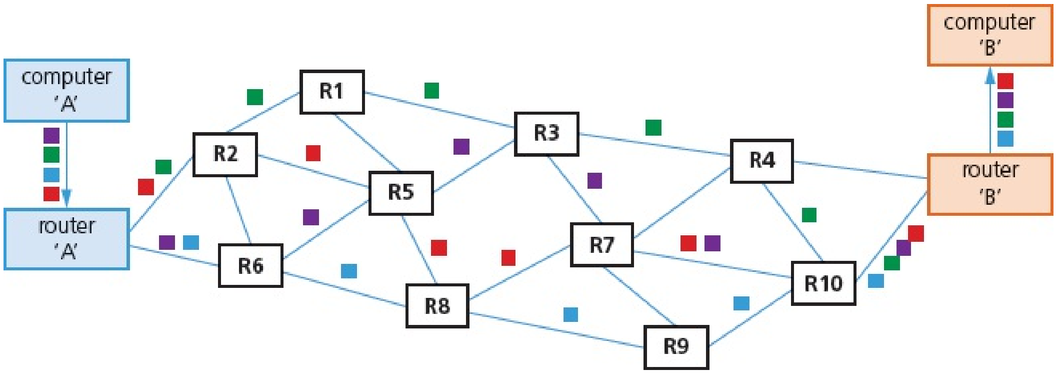


Figure 14.8 An example of packet switching

As Figure 14.8 shows, the message sent by computer 'A' was split into four packets. The original packet order was: 🟥🟦🟩🟪 and they arrived in the order: 🟥🟪🟩🟦 which means they need to be reassembled in the correct order at the destination.

The pros and cons of packet switching are summarised in this table.

Pros	**Cons**
no need to tie up a communication line	the protocols for packet switching can be more complex than those for circuit switching
it is possible to overcome failed or faulty lines by simply re-routing packages	if a packet is lost, the sender must re-send the packet (which wastes time)
it is easy to expand the traffic usage	does not work well with real-time data streams

circuit switching charges the user on the distance and duration of a connection, but packet switching charges users only for the duration of the connectivity	the circuit/channel has to share its bandwidth with other packets
high data transmission is possible with packet switching	there is a delay at the destination while packets are reassembled
packet switching always uses digital networks which means digital data is transmitted directly to the destination	needs large amounts of RAM to handle the large amounts of data

Table 14.4 Pros and cons of packet switching

Comparison of circuit switching and packet switching

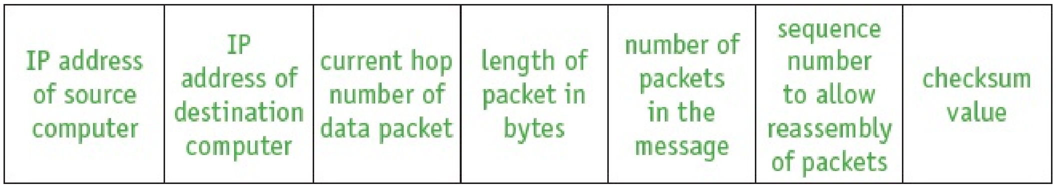
Feature	**Circuit switching**	**Packet switching**
actual route used needs to be set up before transmission can begin	✓	✗
a dedicated transmission path is required	✓	✗
each packet uses the same route	✓	✗
packets arrive at destination in the correct order	✓	✗
all the bandwidth of the channel is required	✓	✗
is bandwidth wasted?	✓	✗

Table 14.5 Comparison of circuit switching and packet switching

Sometimes it is possible for packets to get lost and keep ‘bouncing’ around from router to router and never actually get to their destination. Eventually, the network could grind to a halt as the number of ‘lost’ packets mounts up and clogs up the system. To overcome this, a method called **hopping** is used. A **hop number** is added to the header of each packet. Each packet is only allowed to hop a finite number of times (this number is determined by the network protocol and routing table being used). Each time a packet passes through a router, the hop number is decreased by 1. If the packet has not reached its destination and the hop number = 0, then it will be deleted when it reaches the next router.

Each packet also contains an error checking technique such as a checksum or parity check. If a checksum is used, this value is calculated for each packet and is added to the header. The checksum for each package is recalculated at the destination to ensure no errors have occurred. If the checksum values are different, then a request is made to re-send the packet. A priority value is sometimes also added to a header. A high priority value indicates which packet queue should be used.

This is the make-up of a packet **header** (together with the data in the message being sent) if TCP/IP protocol is being used when sending packets:



IP address of source computer	IP address of destination computer	current hop number of data packet	length of packet in bytes	number of packets in the message	sequence number to allow reassembly of packets	checksum value

Figure 14.9

More generally, packet headers contain the following information (the information used with TCP/IP protocol headers is highlighted in green)

- 4 bits to identify protocol version (such as IPv4, IPv6 – in the example above we assume IP)
- 4 bits to identify header length (in multiples of four; for example, a value of six implies 6 × 4 = 24 bytes)
- 8 bits to represent packet priority
- 16 bits to identify the length of the packet in bytes
- 3 bits are used for fragmentation; the DF flag indicates whether a packet can be fragmented or not (DF = do not fragment) and the MF flag indicates whether there are more fragments of a packet to follow (MF = more fragments)

0	**DF**	MF

- 13 bits to show fragmentation offset to identify the position of the fragments within the original packet
- 8 bits that show the current hop number of the packet
- 16 bits to show the number of packets in the message
- 16 bits to represent the sequence number of the packet
- 8 bits that contain the transmission protocol being used (TCP, UDP)
- 16 bits that contain the header checksum value
- 32 bits that contain the source IP address
- 32 bits that contain the destination IP address.

Routing tables

Routing tables contain the information necessary to forward a package along the shortest/best route to allow it to reach its destination. As soon as the packet reaches a router, the packet header is examined and compared with the routing table. The table supplies the router with instructions to send the packet (hop) to the next available router.

Routing tables include

- number of hops
- MAC address of the next router where the packet is to be forwarded to (hopped)
- metrics (a cost is assigned to each available route so that the most efficient route/path is found)
- network destination (network ID) or pathway
- gateway (the same information as the next hop; it points to the gateway through which target network can be reached)
- netmask (used to generate network ID)

- interface (indicates which locally available interface is responsible for reaching the gateway).

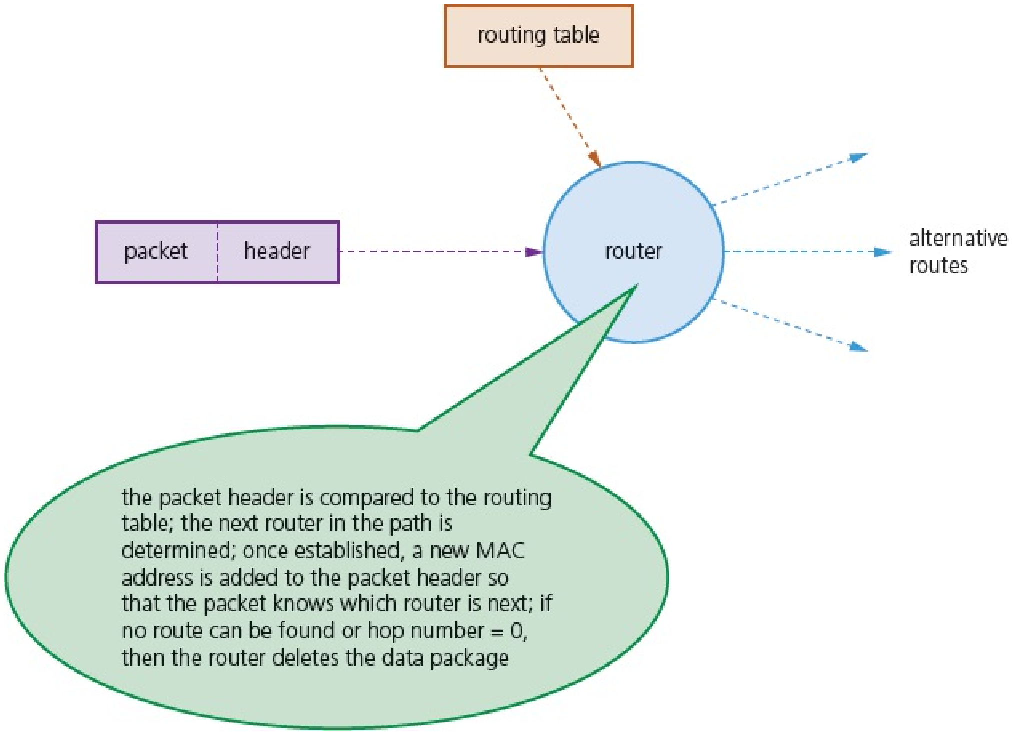

routing table
packet
header
router
alternative routes
the packet header is compared to the routing table; the next router in the path is determined; once established, a new MAC address is added to the packet header so that the packet knows which router is next; if no route can be found or hop number = 0, then the router deletes the data package

Figure 14.10

Example 14.1

Suppose we are carrying out video conferencing using packet switching as the method of routing data. The performance of the communication is not very good.

a) Describe some of the poor performance you might expect.
Give a reason for this poor performance.

b) What features of circuit switching could potentially improve the performance?

Solution

a) Answers might include: picture and sound may not be in synchronisation (packets arriving at different times); video not continuous – pauses (time delay in reassembling the packets of data); degraded quality of sound and video (possibly caused by competing traffic in communication lines); possible drop out (packets take different routes, so it is possible for packets to be lost).

b) Answers might include: only one route used in circuit switching; therefore, all packets arrive in correct order; dedicated communication channel with circuit switching; therefore, full bandwidth available; there is no loss of synchronisation with circuit switching.

Example 14.2

How would packet switching be used to download a page from a website?

Solution

Answers might include

- the web page is divided up into data packets
- each packet has a header, which includes the IP address of the destination
- the router checks the header against values stored in the routing table …
- … to determine which router the packet needs to be sent to next (hopped) …
- … the MAC address of the next router is added to the package header
- the hop value is checked to see if it equals zero
- each packet may go by a different route
- the destination computer reassembles the packets building up the final web page.

ACTIVITY 14A

1 a) Name the **four** layers which show the TCP/IP protocols.

b) Name **one** protocol associated with each layer.

c) i) Describe the use of protocols when sending and receiving emails.

ii) What is the difference between SMTP and MIME when sending emails?

2 a) What is an *Ethernet*?

b) Describe the contents of an *Ethernet frame*.

c) Ethernet protocols do not provide a means to communicate with devices outside a LAN. How can external devices be communicated with when using an Ethernet?

3 a) Explain the following terms used in peer-to-peer BitTorrent.

i) peer

ii) swarm

iii) tracker

iv) leech

v) seed

b) Explain how it could be possible to deal with peers acting as leeches.

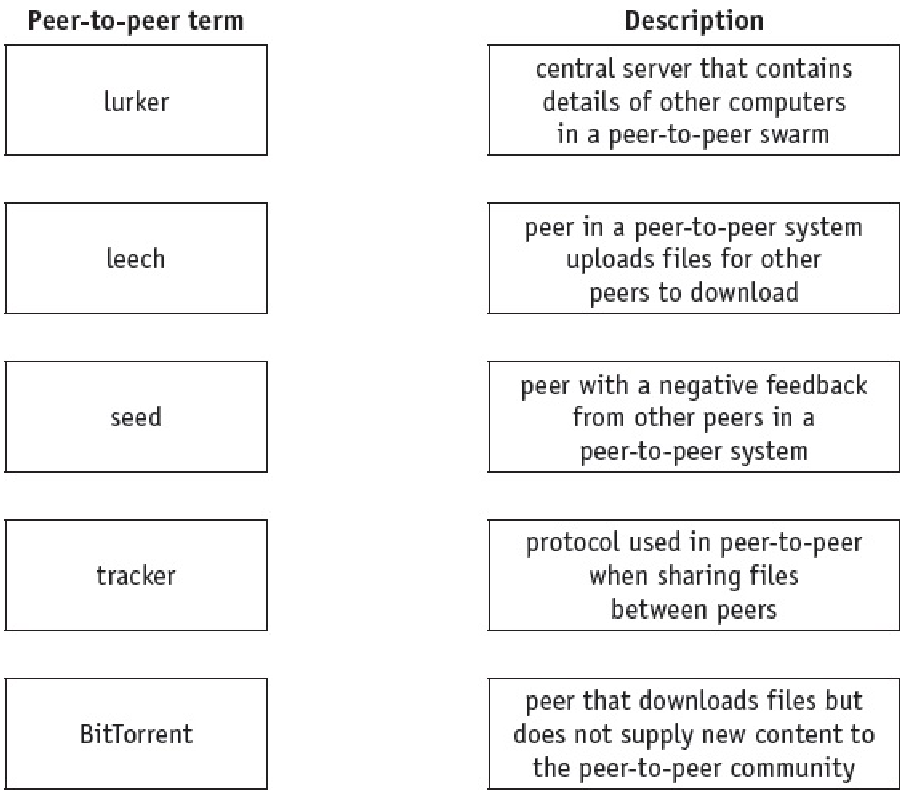
4 a) Describe the difference between a *packet header* and a *routing table*.

b) How are the packet header and the routing table used to route a package?

5 A person is making a video call using VoIP software.
Explain how packet switching could be used and describe any problems that might occur.

End of chapter questions

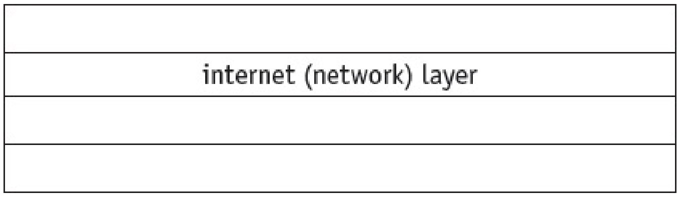
1 a) Copy the diagram below and connect each peer-to-peer term to its correct description. [5]



Peer-to-peer term	Description
lurker	central server that contains details of other computers in a peer-to-peer swarm
leech	peer in a peer-to-peer system uploads files for other peers to download
seed	peer with a negative feedback from other peers in a peer-to-peer system
tracker	protocol used in peer-to-peer when sharing files between peers
BitTorrent	peer that downloads files but does not supply new content to the peer-to-peer community

b) Copy and complete the diagram to show the layers in a TCP/IP protocol.

[3]



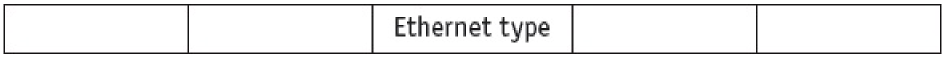
internet (network) layer

c) Describe the protocols used when sending and receiving emails.

[4]

2 a) An Ethernet frame contains a section called Ethernet data.
Copy and complete this diagram to show the other four items missing from the Ethernet data section.

[4]



		Ethernet type		

b) State what is meant by the term *metadata*.

[1]

c) Describe how files can be shared using the BitTorrent protocol.

[4]

3 a) Explain what is meant by circuit switching.

[2]

b) There are many applications in which digital data are transferred across a network. Video conferencing is one of these.
For this application, circuit switching is preferable to the use of packet switching.
Explain why this is so.

[6]

c) A web page is transferred from a web server to a home computer using the Internet.
Explain how the web page is transferred using packet switching.

[3]

Cambridge International AS & A Level Computer Science 9608
Paper 32 Q3 November 2015

4 a) This table shows some statements about circuit switching and packet switching.
Copy the table and indicate which statements are true (✓) and which are false (✗).

[5]

statements	**circuit switching**	**packet switching**
a dedicated circuit/path is needed at all times		
the same route/circuit is used for every packet in the message		
bandwidth is shared with other packets of data		
none of the bandwidth available is wasted during transmission		
packets arrive at the destination in the correct order		

b) Explain the following terms and why they are used when sending packets across a network.

i) hop number/hopping

[2]

ii) checksum

[2]

c) Describe how headers and routing tables are used to route packets efficiently from a sender to recipient.

[5]