

AQA Computer Science A-Level
**4.9.4 The Transmission Control Protocol /
Internet Protocol (TCP/IP)**
Advanced Notes



Specification:

4.9.4.1 TCP/IP:

Describe the role of the four layers of the TCP/IP stack (application, transport, network, link)

Describe the role of sockets in the TCP/IP stack

Be familiar with the role of MAC (Media Access Control) addresses

Explain what the well-known ports and client ports are used for and the differences between them.

4.9.4.2 Standard application layer protocols:

Be familiar with the following protocols:

- FTP (File Transfer Protocol)
- HTTP (Hypertext Transfer Protocol)
- HTTPS (Hypertext Transfer Protocol Secure)
- POP3 (Post Office Protocol (v3))
- SMTP (Simple Mail Transfer Protocol)
- SSH (Secure Shell)

Be familiar with FTP client software and an FTP server, with regard to transferring files using anonymous and non-anonymous access

Be familiar with how SSH is used for remote management

Know how an SSH client is used to make a TCP connection to a remote port for the purpose of sending commands to this port using application level protocols such as GET for HTTP, SMTP commands for sending email and POP3 for retrieving email

Be familiar with using SSH to log in securely to a remote computer and execute commands

Explain the role of an email server in retrieving and sending email

Explain the role of a web server in serving up web pages in text form

Understand the role of a web browser in retrieving web pages and web page resources and rendering these accordingly



4.9.4.3 IP address structure:

Know that an IP address is split into a network identifier part and a host identifier part

4.9.4.4 Subnet masking:

Know that networks can be divided into subnets and know how a subnet mask is used to identify the network identifier part of the IP address

4.9.4.5 IP standards:

Know that there are currently two standards of IP address, v4 and v6
Know why v6 was introduced

4.9.4.6 Public and private IP addresses:

Distinguish between routable and non-routable IP addresses

4.9.4.7 Dynamic Host Configuration Protocol (DHCP):

Understand the purpose and function of the DHCP system

4.9.4.8 Network Address Translation (NAT):

Explain the basic concept of NAT and why it is used

4.9.4.9 Port forwarding:

Explain the basic concept of port forwarding and why it is used

4.9.4.10 Client server model:

Be familiar with the client server model

Be familiar with the Websocket protocol and know why it is used and where it is used

Be familiar with the principles of Web CRUD Applications and REST:

- CRUD is an acronym for create, retrieve, update, delete
- REST enables CRUD to be mapped to database functions (SQL)

Compare JSON (Javascript object notation) with XML

4.9.4.11 Thin- versus thick-client computing

Compare and contrast thin-client computing with thick-client computing



TCP / IP

TCP / IP stands for [transmission control protocol / internet protocol](#). The protocol is used in [all parts of the Internet](#) to enable different devices to communicate.

The protocol is formed from [four distinct layers](#) that form the TCP / IP stack. These layers are application, transport, network, link and each is [responsible for a separate part of communication](#) over the Internet.

| Layer | Role |
|-------------|---|
| Application | <p>Selects and uses the correct protocol to transmit data. The layer interacts with the user with application software like a web browser.</p> <p>Let's suppose we're sending the following message over the Internet:</p> <p style="text-align: center;"><i>Would you like to go bowling tonight?</i></p> |
| Transport | <p>Establishes what's called a virtual path: an end to end connection between the sender and the receiver. The layer then splits the transmission into packets.</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px; text-align: center;"> Port: 443 Would you Packet 1 of 3 </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> Port: 443 like to go Packet 2 of 3 </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> Port: 443 bowling tonight? Packet 3 of 3 </div> </div> <p>Each packet has a sequence number which identifies a packet's position in relation to other packets that form part of the same message. Also contained in each packet is the port number to be used which identifies the protocol in use. In this case, port 443 is HTTPS.</p> |
| Network | <p>Provides the correct IP addresses for each packet's source and destination.</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px; text-align: center;"> Port: 443 Would you Packet 1 of 3 <hr/> 41.23.128.5 114.26.20.199 </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> Port: 443 like to go Packet 2 of 3 <hr/> 41.23.128.5 114.26.20.199 </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> Port: 443 bowling tonight? Packet 3 of 3 <hr/> 41.23.128.5 114.26.20.199 </div> </div> <p>Routers work within the network layer, using the IP addresses on a packet to send it to its destination.</p> |



| | | | | | | | | | | | | | | | | |
|--|---|--|-----------|-----------|---------------|------------------------------|--|-----------|------------|---------------|------------------------------|--|-----------|------------------|---------------|------------------------------|
| Link | <p>Controls physical connections between pieces of hardware in a network. Adds MAC addresses to packets which it receives from the network layer. MAC (which stands for media access control) addresses are assigned to every device that can connect to a network by their manufacturer and are unique to that device.</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <table border="1" style="border-collapse: collapse; text-align: center; width: 150px;"> <tr><td>A4:42:4E:D2:21:7C 3C:AC:B4:53:11:D1</td></tr> <tr><td>Port: 443</td></tr> <tr><td>Would you</td></tr> <tr><td>Packet 1 of 3</td></tr> <tr><td>41.23.128.5 114.26.20.199</td></tr> </table> <table border="1" style="border-collapse: collapse; text-align: center; width: 150px;"> <tr><td>A4:42:4E:D2:21:7C 3C:AC:B4:53:11:D1</td></tr> <tr><td>Port: 443</td></tr> <tr><td>like to go</td></tr> <tr><td>Packet 2 of 3</td></tr> <tr><td>41.23.128.5 114.26.20.199</td></tr> </table> <table border="1" style="border-collapse: collapse; text-align: center; width: 150px;"> <tr><td>A4:42:4E:D2:21:7C 3C:AC:B4:53:11:D1</td></tr> <tr><td>Port: 443</td></tr> <tr><td>bowling tonight?</td></tr> <tr><td>Packet 3 of 3</td></tr> <tr><td>41.23.128.5 114.26.20.199</td></tr> </table> </div> <p>The MAC addresses identify the hardware to which a packet should be sent. These MAC addresses change with every hop through a network.</p> | A4:42:4E:D2:21:7C 3C:AC:B4:53:11:D1 | Port: 443 | Would you | Packet 1 of 3 | 41.23.128.5 114.26.20.199 | A4:42:4E:D2:21:7C 3C:AC:B4:53:11:D1 | Port: 443 | like to go | Packet 2 of 3 | 41.23.128.5 114.26.20.199 | A4:42:4E:D2:21:7C 3C:AC:B4:53:11:D1 | Port: 443 | bowling tonight? | Packet 3 of 3 | 41.23.128.5 114.26.20.199 |
| A4:42:4E:D2:21:7C 3C:AC:B4:53:11:D1 | | | | | | | | | | | | | | | | |
| Port: 443 | | | | | | | | | | | | | | | | |
| Would you | | | | | | | | | | | | | | | | |
| Packet 1 of 3 | | | | | | | | | | | | | | | | |
| 41.23.128.5 114.26.20.199 | | | | | | | | | | | | | | | | |
| A4:42:4E:D2:21:7C 3C:AC:B4:53:11:D1 | | | | | | | | | | | | | | | | |
| Port: 443 | | | | | | | | | | | | | | | | |
| like to go | | | | | | | | | | | | | | | | |
| Packet 2 of 3 | | | | | | | | | | | | | | | | |
| 41.23.128.5 114.26.20.199 | | | | | | | | | | | | | | | | |
| A4:42:4E:D2:21:7C 3C:AC:B4:53:11:D1 | | | | | | | | | | | | | | | | |
| Port: 443 | | | | | | | | | | | | | | | | |
| bowling tonight? | | | | | | | | | | | | | | | | |
| Packet 3 of 3 | | | | | | | | | | | | | | | | |
| 41.23.128.5 114.26.20.199 | | | | | | | | | | | | | | | | |

At the receiving end

Once a packet has been received by its intended recipient, it is **stripped of its extra information** by reversing the TCP / IP stack.

Firstly, the link layer **removes MAC addresses** from the packet. Next, the network layer **removes IP addresses** before the transport layer uses the packet's **port number** to determine the correct **application** to send the packet to. The transport layer also uses the packet's **sequence number** to ensure that it is in the **correct position** relative to other packets in the same transmission.

Finally, the application layer receives the packets and **displays the information** to the user accordingly.

Socket addresses

When an IP address is **combined** with a port number, a **socket address** is formed. These are formed from an IP address, followed by a colon, followed by the port number in use.

114.26.20.199:443

IP address

Port number

Socket address

A socket address identifies which of the applications on the recipient device a packet should be sent to.



Well-known ports

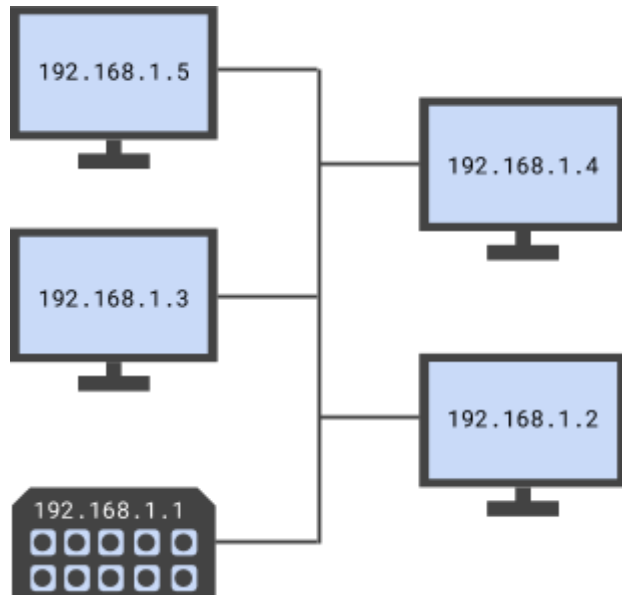
The table below lists some well-known ports, their port number and their purpose.

| Protocol | Port(s) | Purpose |
|---|---------------|--|
| FTP (File transfer protocol) | 20 & 21 | Used for sending files between devices . FTP client software can run on devices, allowing them to connect to FTP servers which send files requested by the client. Clients can access FTP servers anonymously or non-anonymously by logging in with a username and password. |
| SSH (Secure shell) | 22 | Used for remote management of computers , allowing computers to be accessed and controlled remotely. In order to access a remote computer, SSH requires a username / password combination and encrypts information during transmission. SSH clients are pieces of software which can be used to make a TCP connection to a remote port. Once a connection is established, commands for application-level protocols can be sent to the remote computer. |
| HTTP (Hypertext transfer protocol) | 80 | Web servers hold web pages in text form, which they can deliver to clients using HTTP. Application software on the client receives these text files and renders them as web pages. |
| HTTPS (Hypertext transfer protocol secure) | 443 | HTTPS performs the same function as HTTP, but encrypts information during transmission. This doesn't just keep information sent by clients secure, but helps to prevent information from being tampered with or modified during transmission. |
| POP3 (Post office protocol 3) | 110 & 995 | Used for retrieving emails from an email server. Email servers are responsible for receiving and sending emails. POP3 communicates with email servers to check for and download any new messages. |
| SMTP (Simple mail transfer protocol) | 25, 587 & 465 | Used for sending emails. SMTP is used between a client and an email server. |



The structure of IP addresses

An IP address is split into **two parts**: a **network identifier** and a **host identifier**. Each of the computers in a network shares the same network identifier but has its own host identifier.



Networks can be divided into **smaller networks**, called **subnets**. Each subnet has a different network identifier.

The network identifier part of an IP address can be determined with a **subnet mask**.

Let's say that a device has an IP address of 192.168.3.24 and a subnet mask of 255.255.255.0.

In order to work out which subnet the device belongs to, we have to apply the subnet mask as follows.

192.168.3.24
IP Address of device

11000000.10101000.00000011.00011000
IP Address of device (**binary**)

255.255.255.0
Subnet mask

11111111.11111111.11111111.00000000
Subnet mask (**binary**)

11000000.10101000.00000011.00000000
IP address **AND** subnet mask

192.168.3.0
Network identifier (IP address AND subnet mask, in decimal)

As it happens, 255.255.255.0 is a **fairly common** subnet mask that is **easy to use** in examples. However, the same procedure can be applied to any subnet mask.

The more bits that are assigned to the **network identifier** of an IP address, the more **different subnets** a network can have. In the same way, the more bits that are assigned to the **host identifier**, the more **different devices** can be connected to each subnet simultaneously.



IP address standards

There are **two types** of IP address in common use: versions **four** and **six** (IPv4 and IPv6).

IPv4

IPv4 addresses are **dotted quad numbers**, meaning that they consist of **four parts** that are **separated by dots**. Each of the four parts of an IPv4 address is assigned **one byte** (eight bits) allowing for numbers from 0 to 255 to be represented.

192 . 168 . 34 . 7

This allows for a total of slightly **over 4 billion** (256^4) unique IPv4 addresses. That may sound like a lot, but IPv4 addresses are in **short supply**. The number of devices on the Internet that require a routable IP address is **increasing so rapidly** that a new version of IP address had to be created. This was IPv6.

IPv6

IPv6 addresses are formed of **eight blocks** separated by **colons**. Each block contains **four hexadecimal characters** (a-z and 0-9).

2071 : 0eb8 : 85a3 : 8a2f : 0000 : 0000 : 0370 : 7264

IPv6 addresses use 128 bits which, compared to IPv4's use of 32, allows for **far more** (in the order of 10^{37}) unique permutations.

Public and private IP addresses

An IP address is said to be either **routable** or **non-routable** (public or private). If every device that is connected to a network had its own public IP address, there **wouldn't be enough IP addresses** to go around.

Instead, each home or business that requires internet access has a **small number** of public IP addresses. Most homes have **just one** public IP address while some businesses will have **multiple addresses**.

Routable IP addresses are **globally unique** whereas millions of devices can have **the same non-routable IP address**, provided they are not on the same network. **Global authorities** are responsible for assigning routable IP addresses which ensures that the same address is **never issued twice**.



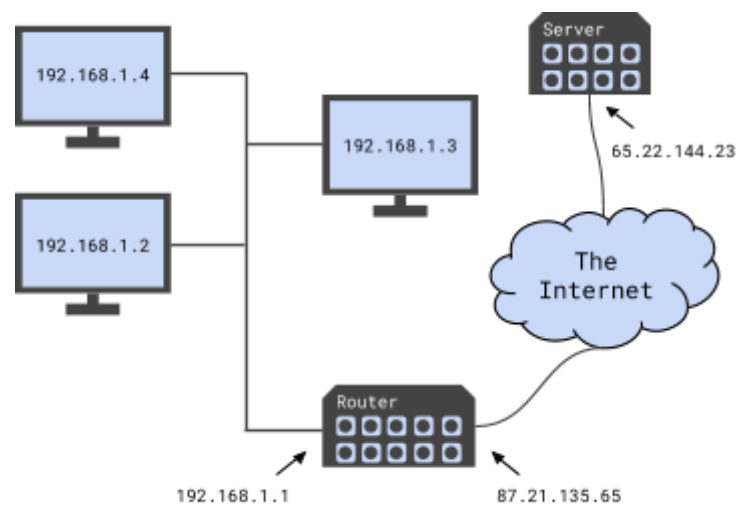
Dynamic host configuration protocol (DHCP)

The number of available **private** IP addresses within a **private** network is **limited**. Assigning each device on a network its own private IP address would not be sensible, as that device may leave and **never join again**, resulting in a **wasted** IP address.

Instead, DHCP is used to assign IP addresses to devices **as they join** a network. DHCP uses a **pool of available IP addresses** to allocate IP addresses to new devices **for the duration** of their session. Once a device leaves the network, the IP address that the device was using is **returned to the pool** for allocation to a new device.

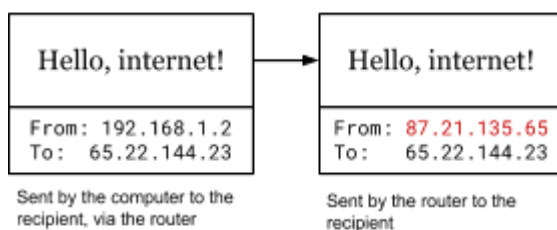
Network address translation (NAT)

The diagram shows a network consisting of three computers, each of which is allocated a **private** (non-routable) IP address. The network's router has two IP addresses, **one private address** on the private network's side **and a public (routable) address** on the Internet's side. There is a server connected to the Internet with a **routable** IP address.

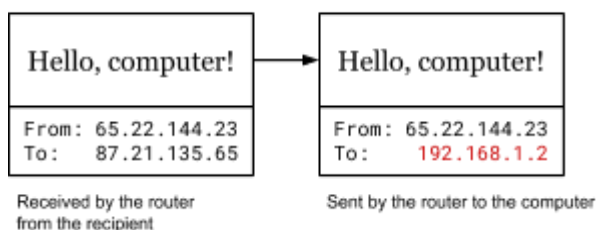


If a device on the network sent a packet to the server, the server **couldn't respond to the computer directly** because the computer's IP address is **non-routable** - not globally unique.

Message



Response



NAT gets around this problem. When a device on the **private network** (let's use 192.168.1.2) needs to communicate with a device **on the Internet** (let's use the server), it sends packets **through the router**, which makes a **record of the packet** before **replacing the private IP address** of the computer with **its own routable IP address**.

When a response is received, it is **sent to the router's public IP address**, which then **forwards the response** to the correct private IP address by **using the record** it made when sending the packet.

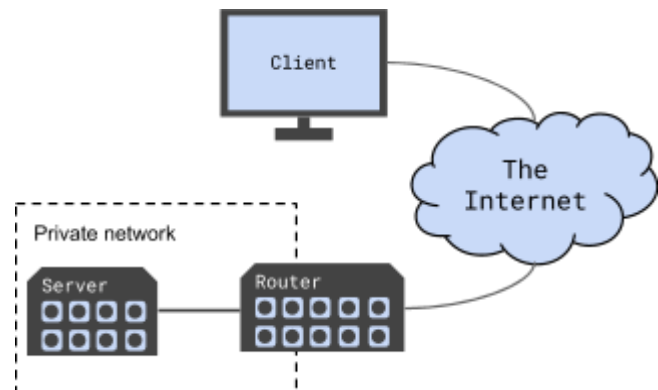


Port forwarding

Port forwarding is used when a client needs to communicate with a server that is connected to a [private network](#).

The client sends packets to the [public IP address of the router](#) belonging to the server's private network.

The packets sent by the client contain the [port number](#) of the application running on the server that the client wishes to access.



The private network's router then [forwards the packets](#) to the server using NAT.

The client server model

In a network that uses the client server model, [clients send request messages to servers](#), which [reply to the clients with response messages](#). These messages may contain requested information, a confirmation that a requested action has been completed, or a message explaining why the requested action hasn't been completed.

There are a number of different types of servers, each of which [specialise](#) in a certain task. For example: file servers, database servers and email servers.

APIs

An API (application programming interface) is a name given to a [set of protocols](#) relating to how [different applications](#) communicate with each other. They define how interaction between the applications should be carried out, allowing applications to [make use of](#) other applications.

The websocket protocol

The websocket protocol is an [example of an API](#) which operates in the [application layer](#) of the TCP / IP stack. The protocol can be used to provide [a constant stream of information](#) between two devices, usually a client's web browser and a server.

The connection created by the websocket protocol is [full-duplex](#), meaning that data can be transmitted in [both directions](#) at the same time.

The websocket protocol, which allows for [fast transmission of data](#) by [reducing the size](#) of packet headers, is used in video streaming, online games and instant messaging.



Web CRUD Applications and REST

CRUD

CRUD is an acronym for **create**, **retrieve**, **update**, **delete**; four commands that can be used to [query online databases](#). Each of the four CRUD commands has a [SQL equivalent](#).

| CRUD | SQL |
|----------|--------|
| Create | INSERT |
| Retrieve | SELECT |
| Update | UPDATE |
| Delete | DELETE |

Synoptic Link

Server query language (SQL) is used to query databases.

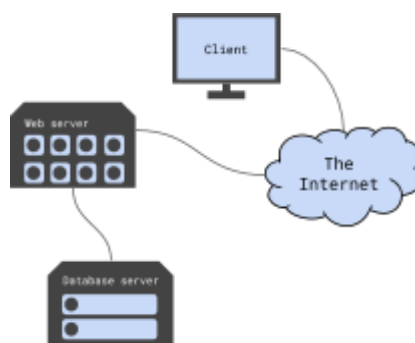
SQL is covered in detail under [fundamentals of databases](#).

REST

An acronym for **representational state transfer**, REST is a [design methodology](#) for online database applications that are queried with a web browser.

REST uses the four [HTTP request methods](#) POST, GET, PUT and DELETE to query databases.

| HTTP | SQL |
|--------|--------|
| POST | INSERT |
| GET | SELECT |
| PUT | UPDATE |
| DELETE | DELETE |



When a client needs to connect to a database, it follows these steps:

1. Client-server [request](#) made by the client to the web browser
2. Web browser [responds](#) with the requested web page (which is delivered as a text file)
3. This text file contains [JavaScript](#) which loads an API. The API uses REST to enable the database server to be [queried by the client](#) with the use of HTTP request methods.
4. The client [sends HTTP requests](#) to the database server.
5. The database server [responds](#) to the client's requests using either JSON or XML.
6. The client's browser [processes](#) the JSON or XML and displays the response to the user.



XML and JSON

Database servers deliver responses to queries using **either XML or JSON**. XML stands for **extensible markup language** and does **the same job as JavaScript object notation (JSON)**.

| XML | JSON |
|---|--|
| <pre> <Department> <Subject> <Name>Physics</Name> <Board>Edexcel</Board> </Subject> <Subject> <Name>Maths</Name> <Board>AQA</Board> </Subject> </Department> </pre> | <pre> { "Department": { "Subject": [{ "Name": "Physics", "Board": "Edexcel" }, { "Name": "Maths", "Board": "AQA" }] } } </pre> |

The table above shows **the same information** represented as both XML and JSON. As the table shows, JSON is **more compact**, **easier to read**, **easier to create** and **faster for computers to process** than XML. However, XML is sometimes seen to be **more flexible** than JSON.



Thin- and thick-client computing

Networks can be configured as either thin-client or thick-client networks.

Thin-client networks

In thin-client networks, the **majority** of the network's processing power belongs to **servers** which provide **services and resources** including storage and processing.

It's **easy to add new clients** to thin-client networks and the clients themselves are **inexpensive** machines. Thin-client networks also allow for **greater centralised control** of the network as software updates and security can be managed **from the server**.

However, thin-client networks **require a powerful server** which is **expensive** and **requires expertise** to set up and maintain.

Thick-client networks

In a thick-client network, the clients are **powerful enough to provide their own processing power and storage**. This independence **eliminates the requirement for a server**, although it is **possible** for thick-client networks can make use of a server.

Thick-client networks require **more powerful clients** than their thin-client counterparts, making the network **expensive to set up**. However, the **cost and expertise** required in setting up and maintaining an expensive server is done away with.

Thick-client networks are **harder to maintain** because there is **no facility to issue updates and manage security** from a central server.

Compared to thin-client networks which suffer from **high volumes of traffic** communicating between clients and the server, thick-client networks boast **much quieter communication channels** which **reduces the likelihood of data collisions**.

