



A12.3 How the Internet works

Networks and the web · A level · OCR H446 1.3.3, AQA 7517 4.9.3.1,
Eduqas A500QS 2.2 · about 25 min

What this lesson is about

Packet switching, routers and gateways, URLs, fully qualified domain names and DNS.

- Packet switching
- Routers and gateways
- The Internet's addresses
- DNS

Questions

6 marks in all

1. Which of these are found in a packet's header? Choose all that apply.

[1 mark]

Tick every answer that is true.

- ☐ A Source IP address
- ☐ B Destination IP address
- ☐ C Packet sequence number
- ☐ D Time to live
- ☐ E The web page's text

Answer: A, B, C, D. The header holds the addressing and control information; the web page's text is part of the payload.

2. When should a gateway be used rather than a router?

[1 mark]

- ☐ A When the networks being joined use different protocols
- ☐ B When the two networks use the same protocols
- ☐ C When a device needs a private IP address
- ☐ D When a domain name must be registered

Answer: A. A router joins networks with the same protocols; a gateway translates between networks that use different protocols.

3. In the URL <https://www.bbc.co.uk/news>, what is the fully qualified domain name?

[1 mark]

Answer: [www.bbc.co.uk](https://www.bbc.co.uk/news). The FQDN includes the host name (www) as well as the domain name (bbc.co.uk); the scheme and path are not part of it.

4. Put the steps of a DNS lookup for lessons.bugbot.example, with nothing cached, in order.

[1 mark]

Number the lines 1 to 5 to put them in the right order.

- ___ The resolver asks the authoritative name server for bugbot.example
- ___ The resolver caches the IP address and returns it to the computer
- ___ The resolver asks a root name server
- ___ The resolver asks the top-level domain server for .example
- ___ The computer asks its resolver

Answer:

```
The computer asks its resolver
The resolver asks a root name server
The resolver asks the top-level domain server for .example
The resolver asks the authoritative name server for bugbot.example
The resolver caches the IP address and returns it to the computer
```

Each server refers the resolver one step further down the hierarchy until the authoritative server gives the address.

5. Which is an advantage of packet switching over circuit switching?

[1 mark]

- ☐ A Links are shared by many conversations, and packets can be routed around a failed link
- ☐ B Data always arrives in the order it was sent
- ☐ C The delay is always constant
- ☐ D A dedicated path is reserved for the whole conversation

Answer: A. Packet switching uses links efficiently and is resilient; circuit switching reserves a path, which keeps data in order with constant delay.

6. What does this program print?

[1 mark]

```
packets = {3: 'ket', 1: 'pa', 2: 'c'}
message = ''
for n in sorted(packets):
    message = message + packets[n]
print(message)
```

Answer:

packet

The payloads are joined in sequence number order 1, 2, 3, whatever order they arrived in.

The task: put the packets back together

The Server on this mat sends one short message as four packets, one every 0.4 seconds, but not in order. Each packet is text in the form `<number>/<total>:<payload>`, for example `2/4:an take d`. The number counts from 1, the total is how many packets make the message, and the payload is everything after the first colon (it may contain spaces). Listen with `messages()`, checking every 0.1 s, until you hold every packet. Then print two lines: - `arrived in order: <numbers>`, the packet numbers in the order they arrived, separated by single spaces; - `message: <text>`, the payloads joined in number order. The robot does not drive.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

wait(2)
for sender, text in messages():
    print(text)
```

The hint students can ask for: Split each packet at the first colon, then split the header at the slash to get its number and the total. Keep the payloads in a dictionary by number until you hold as many as the total says, then join them in number order.

A solution

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

payloads = {}
arrived = []
total = None
while total is None or len(payloads) < total:
    wait(0.1)
    for sender, text in messages():
        header, data = text.split(":", 1)
        number, count = header.split("/")
        payloads[int(number)] = data
        arrived.append(number)
        total = int(count)
print("arrived in order:", " ".join(arrived))
message = ""
for n in range(1, total + 1):
    message = message + payloads[n]
print("message:", message)
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.