



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

Name _____

Class _____

Date _____

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

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

3. In the URL <https://www.bbc.co.uk/news>, what is the fully qualified domain name? [1 mark]

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

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
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)
```

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)
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/a12-3-the-internet-and-dns/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Add a TTL to the DNS cell: stop a lookup after 2 servers and print `too many hops` . The lookup above needs 3, so it should now fail.
2. Drop one packet: have your program give up after 3 seconds and print which packet numbers are missing.
3. Split `https://www.bbc.co.uk/news/technology` into its scheme, FQDN, domain name and path with string methods.