WORKSHEET

F10.5 Addresses and packets

Networks · GCSE · OCR J277 1.3.2, AQA 8525 3.5, Edexcel 1CP2 4.1.3 · about 15 min

BugBotLab

Name _____ Class _____ Date _____

What this lesson is about

MAC and IP addresses, IPv4 and IPv6, and packet switching.

- MAC addresses
- IP addresses
- Packet switching

Questions

6 marks in all

1. Which of these is a valid IPv4 address? [1 mark]

- ☐ **A** 192.168.0.12
- ☐ **B** 192.168.300.1
- ☐ **C** 3C:61:05:1A:9F:02
- ☐ **D** 10.0.1

2. What is a MAC address used for? [1 mark]

- ☐ **A** Identifying a network interface, to deliver data within a local network
- ☐ **B** Routing data between networks
- ☐ **C** Naming a website
- ☐ **D** Encrypting data

3. How many bits is an IPv6 address? [1 mark]

4. Why do packets have sequence numbers? [1 mark]

- ☐ **A** So they can be put back in order when they arrive
- ☐ **B** So routers know the route
- ☐ **C** To encrypt them
- ☐ **D** To make them smaller

5. Which is in a packet header? [1 mark]

- ☐ **A** The destination IP address
- ☐ **B** The whole file
- ☐ **C** The user's password
- ☐ **D** The web page's title

6. What does this program print? [1 mark]

```
packets = ["2/2:Bot", "1/2:Bug"]
packets.sort()
print("".join(p[4:] for p in packets))
```

The task: packets

Write `make_packets(message, size)`, which splits `message` into pieces of `size` characters and returns them as packets in the form `<number>/<total>:<piece>`. Make the packets for `message` with a size of 10, print each one, and send each by radio. The packets arrive in the order `[packets[2], packets[0], packets[1]]`: sort them back into order by their sequence numbers, and print `reassembled: <message>` and `matches: True` (or `False`).

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

message = "robot 7 at 25,40 battery 87%"
```

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

Do it on the robot

www.bugbotlab.com/learn/f10-5-addresses-and-packets/

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

Challenges

1. Shuffle the packets with `random.shuffle` and check they still reassemble.
2. Add a checksum to each packet: the total of the `ord()` values of its characters, modulo 100. Check it on arrival.
3. How many IPv4 addresses are there exactly? How many IPv6? Use `2 **32` and `2 **128`.