WORKSHEET

A12.10 Project: a reliable link

Networks and the web · A level · OCR H446 1.3.3, AQA 7517 4.9.3.1, Eduqas A500QS 2.2 · about 35 min

BugBotLab

Name _____ Class _____ Date _____

What this lesson is about

Receive a route as packets out of order, reject the damaged one, and drive it.

- What the project uses
- The packet format
- How a single bit is caught
- Designing the program

Questions

5 marks in all

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

```
def checksum(text):
    return format(sum(ord(c) for c in text) % 256, '02X')
print(checksum('1/3:forward 30'))
```

2. A receiver recalculates a packet's checksum and it does not match. What should it do? [1 mark]

- ☐ **A** Discard the packet and ask for it to be sent again
- ☐ **B** Use the packet, because the header is still readable
- ☐ **C** Change the checksum so it matches
- ☐ **D** Stop the whole transfer permanently

3. Which error would the checksum 'sum of character codes modulo 256' fail to detect? [1 mark]

- ☐ **A** Two characters swapped
- ☐ **B** One bit flipped in one character
- ☐ **C** A character missing
- ☐ **D** An extra character added

4. Why does the receiver send an ACK for packets that arrive intact? [1 mark]

- ☐ **A** So the sender can tell a delivered packet from one that was lost and needs sending again
- ☐ **B** So the packet's checksum can be recalculated
- ☐ **C** Because routers need ACKs to find a route
- ☐ **D** To encrypt the next packet

5. Which layer of the TCP/IP stack numbers segments, acknowledges them and resends missing ones? [1 mark]

- ☐ **A** Transport
- ☐ **B** Application
- ☐ **C** Network
- ☐ **D** Link

The task: a reliable link

The robot starts facing forward. The Base sends a route of three packets in the format above, one every half second, out of order. One packet arrives damaged. When the Base hears `NAK <number>`, it sends that packet again. 1. Receive: check `messages()` every 0.1 s. For each packet, recalculate its checksum. If it matches, store the command under the packet's number and send `ACK <number>`. If it does not, print `packet <number> corrupt` and send `NAK <number>`. Keep going until you hold a good copy of every packet, as many as the total in the header says. 2. Act: for each number from 1 to the total, in order, print `run: <command>`, then drive it at speed 50. A command is a direction (`forward`, `backward`, `left` or `right`, where left and right slide sideways) and a whole number of centimetres. 3. Send `DONE`. Do not drive until every packet is held, and read the route from the packets rather than typing it. If you act on the damaged packet, you will end up in the red zone.

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

def checksum(text):
    total = 0
    for ch in text:
        total = total + ord(ch)
    return format(total % 256, "02X")

wait(3)
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/a12-10-project-a-reliable-link/

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

Challenges

1. Count how many packets you received in total, including the damaged one, and print the fraction that were good.
2. Ignore a duplicate: if a good packet you already hold arrives again, acknowledge it but do not store it twice. Why must you still send the ACK?
3. Replace the checksum with one that notices two characters being swapped, for example by multiplying each code by its position before adding. Test it with the swapped example in the cell.