

Solutions

Each answer shows the steps, then the **result**.

2.1

- a packet has three parts
- **header, payload, trailer**

2.2

- **serial**: one bit at a time down one wire
- **parallel**: several bits at once down several wires

2.3

- USB sends bits one after another, so **serial**
- it can send both ways, but not at the same time, so **half-duplex**

2.4

- count the 1s in 1011011: five, which is already odd
- odd parity needs an odd number of 1s, so the parity bit is **0**; byte sent **10110110**

2.5

- any **two** of: connector fits only one way / device detected automatically and driver loaded / carries power as well as data / industry standard

3.1 B

- only one person speaks at a time, but both can speak, so **half-duplex**
- simplex is one way only; full-duplex is both ways at once; parallel is about wires, not turn-taking

3.2

(a)

- any **two** of: only a lost packet is resent / packets can take different routes / many users share the connection

(b)

- packets take different routes and can arrive out of order
- **packet numbers** put them back in the right order

3.3

(a)

- **serial** transmission
- over 200 m, parallel wires would **skew** (bits arrive at different times)
- the wires would also interfere, so serial is more reliable and needs less cable

(b)

- data only ever goes one way
- **simplex**

(c)

- 0110110 has four 1s (already even) → parity bit **0**
- 1111011 has six 1s (already even) → parity bit **0**
- 0000001 has one 1 (odd) → parity bit **1**

(d)

- a parity check only counts the number of 1s
- if one bit flips 0→1 and another 1→0, the count is unchanged, so the **error is missed**

(e)

- the receiver sends a copy straight back; the sender compares it with what it sent
- disadvantage: **doubles the traffic**, or cannot tell whether the error happened on the way out or back

(f)

- any **two** of: connector fits only one way / device and driver detected automatically / carries power / universal standard