UNIVERSITY COMPUTER SCIENCE • EXERCISE SHEET

9.1 Layering and architecture

Name: _____ Class: _____ Date: _____ **Total: 24 marks**

KEY WORDS layering • service

Objective

Reason about layers, interfaces and encapsulation sizes.

1 Key knowledge and worked examples

■ The stack

Physical, link, network, transport, application ✓. Services point up; protocols point down; encapsulation adds one header per layer going down ✓.

■ Worked example: the sizes

Ethernet 14 + IPv4 20 + TCP 20 = 54 header bytes around a 100-byte payload; the receiver strips 14, then 20, then 20 ✓.

2 Practice

2.1 Name the layer that moves frames between neighbours, and the one that moves packets between strangers. [3]

2.2 A message crosses 3 layers going down. How many headers does it gain? [3]

2.3 What does a service interface guarantee the layer above? [3]

3 Exam-style questions

3.1 An application message of 200 bytes is sent over Ethernet carrying IPv4 and TCP.

Compute the bytes on the wire (headers + payload). Compute the fraction of the frame that is header. State which layer strips which header at the receiver, in order. Explain in one line why replacing Ethernet with a new link technology need not change TCP, naming the principle. [8]

3.2 A colleague proposes reading the IP TTL field inside the browser to detect bad routes.

Name the violation. State what happens to that browser when the network stack below changes. Give the correct layer for TTL logic and the interface it should respect. State the hourglass argument for why IP itself should stay narrow. [7]

Solutions

2.1

- Link (frames, neighbours); network (packets, strangers) ✓.

2.2

- Three headers – one per layer crossed ✓.

2.3

- The promised service, unchanged – whatever the layer does below ✓.

3.1

- Wire bytes: $14 + 20 + 20 + 200 = 254$ ✓.
- Header fraction: $54/254 \approx 21.3\%$ ✓.
- Strips: NIC strips the 14-byte Ethernet header; the IP layer strips 20; TCP strips its 20 – each layer sees only its own promise ✓.
- The link service interface (best-effort frames) held, so TCP never noticed – the principle of layer independence ✓.

3.2

- Violation: a leaky interface – the application reading a network layer header below it ✓.
- When the stack changes, that browser breaks: it was written against an implementation detail, not a service ✓.
- TTL belongs in the network layer (routers decrement it); anything needing it should sit at or below that layer's interface ✓.
- The hourglass: one narrow waist protocol (IP) lets every application above run over every link below – widening the waist multiplies the combinations nobody can maintain ✓.