(1) What is the primary advantage of “sliding-window flow-control” over “stop-and-wait flow-control”?

Sliding-window flow control allows senders to transmit more than one packet before the first ACK comes back from receivers. This improves the link utilization.

(2) Why isn’t a good idea to use “a hard-coded fixed window size” in TCP?

TCP is the default transport-layer protocol in the Internet, where:

(1) The end-to-end link distance between two communicating host computers varies for each connection (from one foot to several thousand miles). This means the signal propagation delay also varies, which changes the optimum window size for each connection.

(2) The network traffic load in the Internet constantly changes. Also, depending on the paths each connection goes through, the network traffic load varies. All these change the signal propagation delay.

(3) Each time your host computer establishes a connection to a different host computer, the performance (the processor performance, the memory capacity, the speed of the network connection the destination host computer uses) of the destination host computer also varies.

Because of the above unknown parameters above, “a hard-coded fixed window size” in TCP will result in sub-optimum performance (link utilization).
(3) Show the (whole) structure (the flowchart) of error control for packet-switching networks.

(4) Demonstrate how “undetectable errors” happen using parity-error detection.

```
<table>
<thead>
<tr>
<th>SENDER</th>
<th>RECEIVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001011</td>
<td>1100111</td>
</tr>
</tbody>
</table>
```

“No Odd Parity” is used

(5) Can “undetectable errors” be “correctable”?

Never. If you do not see any problem, you would never try to fix it.