**Activity 1: Protocols**

Selective-Repeat would make more efficient use of network bandwidth, as it only retransmits messages lost at the receiver. In GBN, the sender retransmits the first lost message as well as all following messages up until the latest packet.

**Activity 2: Reliable Data Transfer Protocol**

If a sender sends data infrequently, ACK-only would be better, because if data is lost at time T, receiver only knows this at T+1. If there are long waiting periods between receiving data, then it would also take a awhile to send a NAK.

NAK-only would benefit in a situation where sender sends frequently, as a receiver would know of errors quickly.

**Activity 3: Additive Increase Multiplicative Decrease (AIMD) algorithm**

**Activity 4: TCP Senders**

**Activity 5: RTT from London to Sydney**

RTT = 120ms | 155Mbits/s rate.

1000ms = 1s

Therefore, 120ms = 120/1000s = 0.12 seconds.

0.12 seconds \* 155Mbit = 18.6Mbits = 1000 \* 18.6 = 18,600Kb = 1000 \* 18,600Kb = 18,600,000bits = 18,600,000 / 8 = 2,325,000 bytes.

**Activity 6: GBN and SR protocols**

**Activity 8: Protocol RDT 3.0**

RDT 3.0 is a stop-and-wait protocol, so there is no need to track the order of the packets.

**Activity 9: Finite State Model for receiver side RDT 3.0**

**Activity 10: Detecting an error**

1011 0101 1010 1000  
0101 1001 0000 0101 are received,  
along with another 16-bit checksum, 1101 0001 0101 0001

0000 1110 1010 1110 = sum

1111 0001 0101 0001 = the real checksum

This is not the same as the received checksum, therefore is wrong.

**Activity 11: Reliable Data Transfer**

Yes it is possible for an application to enjoy reliable data transfer over UDP, through services provided by the application layer. UDP may be preferred over TCP with RDT in-built as two hosts may be using an application where packet loss is acceptable and where TCP may cause a lot of congestion.

**Activity 12: TCP Segments**

(a) seq #100 – seq #90 = 20 bytes for first packet.

(b) ACK = 90

**Activity 13: TCP Connection**

SRC = y | DEST = x

**Activity 14: TCP Connection**

1. A: SYN

2. B: SYN-ACK

3. A: ACK + DATA SEGMENT

4. A+B: Do rest of data exchange

5. A: FIN

6. B: ACK  
7. B: FIN  
8. A: ACK  
9. A: TIME-WAIT

10. B: CLOSE

**Activity 15: Internet Checksum**

No, the receiver can’t be absolutely sure that no bit errors have occurred, as the computed / received UDP segment with bit errors can coincidentally match with the checksum field.