COMP3203 Final Exam Notes

William Findlay

December 13, 2018

Contents

1 Test 1 Stuff (Brief and Important Only)

- test 1 stuff here

2 ARQs

- **(A)**utomatic **(R)**epeat Re**(Q)**uests
- strategy to handle errors detected by the CRC
  - or whatever other detection method
- main types
  - **stop and wait**
  - sliding window
    - **go back N**
    - **selective reject**

2.1 Sliding Window

2.1.1 Go Back N

- most commonly used sliding window
- sequential frames numbered $n \mod N$
- send up to $N-1$ frames **before an ACK is received**
- **unbounded sequence numbers** is a hurdle for sliding window in **non-FIFO** channels

ACKs and NAKs

- if no error
  - send RR (ACK) for frame[n]
- if error
  - send REJ (NAK) for frame[n]
- if frame lost, send a NAK
- if no ACK or NAK received before *timeout*, **assume lost**

When Sender Receives a NAK[n]

- resend frame[n] and all frames sent since

When a Sender Receives No ACK or NAK

- go back to the previous ACK and resend all frames sent since

2.1.2 Selective Reject

- similar to go back N
- **BUT** we only resend the **lost frame**
  - out of order!
  - receiver needs *sorting logic* to store frames after a NAK

- in general, smaller window size

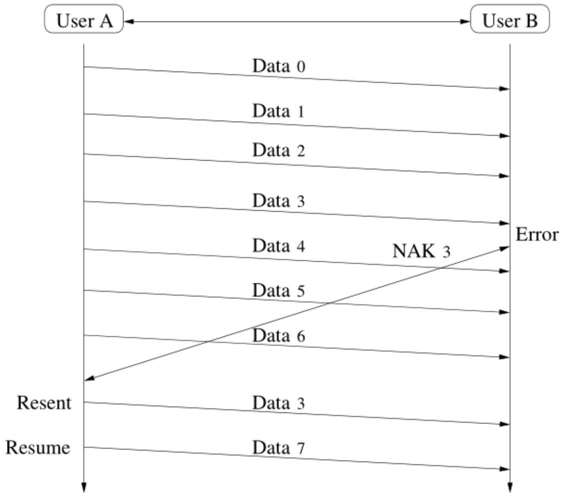

Figure 1: An example of the Selective Reject protocol.

2.2 Stop and Wait

- also called an **ABP**
  - *alternating bit protocol*
  - because the label bits alternate between `0` and `1`
- you can think of it as sliding "window" with a **window size of 1**
- works only in **FIFO queues**
  - suitable for **data link layer**

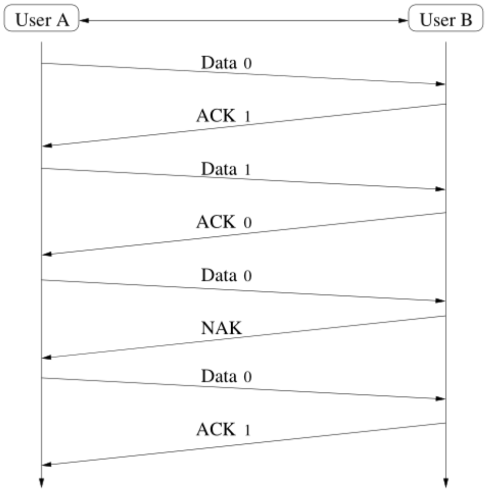

Figure 2: A diagram of the Stop and Wait ARQ protocol.

2.2.1 Errors in Stop and Wait

- two main types

- **frame** errors
  - damaged frame
- **ACK** errors
  - damaged acknowledgement

Frame Errors

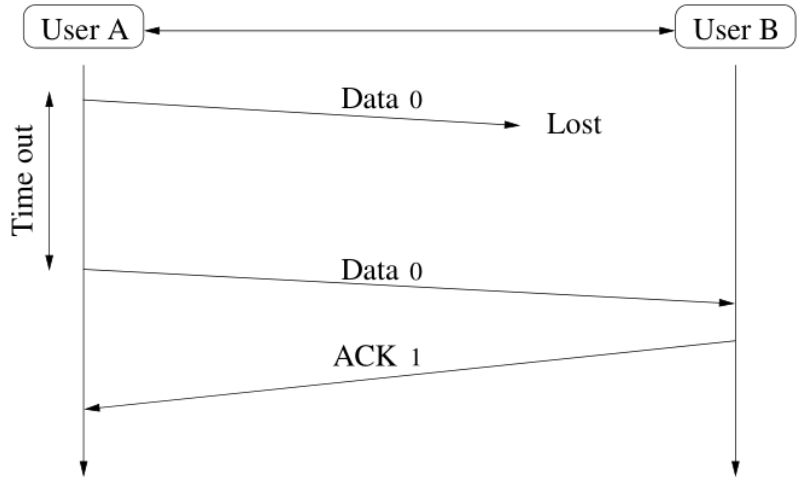

Figure 3: A lost frame error in the Stop and Wait ARQ protocol.

- frame is damaged
  - one or more bits have been altered
- discard the frame
- source waits for ACK
  - if it doesn't receive one, it will resend

ACK Errors

- frame is received but ACK is damaged
- sender will resend message
- receiver will accept the same message twice
  - so we need to label frames
  - and label ACKs
  - use a bit for this
    - ACK[b] acknowledges frame[$b + 1 \mod 2$]
    - says receiver is ready for frame[b]

2.2.2 Correctness

- satisfies:
  - safety
    - algorithm never gives an incorrect result
    - always results in a "corrected" error
  - liveness
    - never enters a deadlock condition

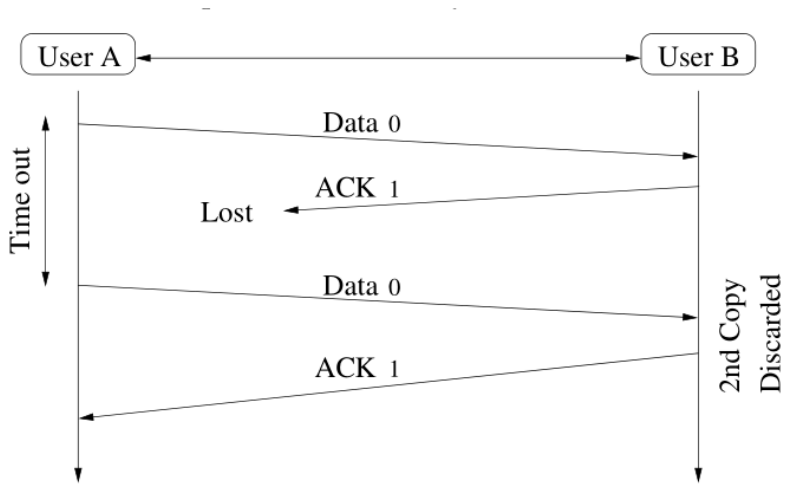

Figure 4: An ACK error in the Stop and Wait ARQ protocol.

3 Multiaccess

3.1 Uncoordinated Access Control

COME BACK HERE

3.2 Ethernet

4 Coordinated Access

4.1 Tree Algorithm

4.2 Binary Countdown

4.3 Bitmap

5 Wireless

5.1 Cellular

5.2 Ad Hoc

5.2.1 UDG

5.2.2 Compass Routing

5.2.3 Face Routing

5.3 Bluetooth

6 GPS

6.1 Three Techniques

6.2 Satellites

7 Routing

7.1 Distance Vector (RIP)

7.2 Link State Protocol (LSP)

7.3 MSTs

7.4 Dijkstra

8 IP

8.1 IPv4

8.1.1 Classes of Address

8.1.2 Subnets

8.1.3 Subnet Masks

8.2 IPv6