The Two Generals Problem

Two army divisions (blue) surround enemy (red). Each division led by a general.

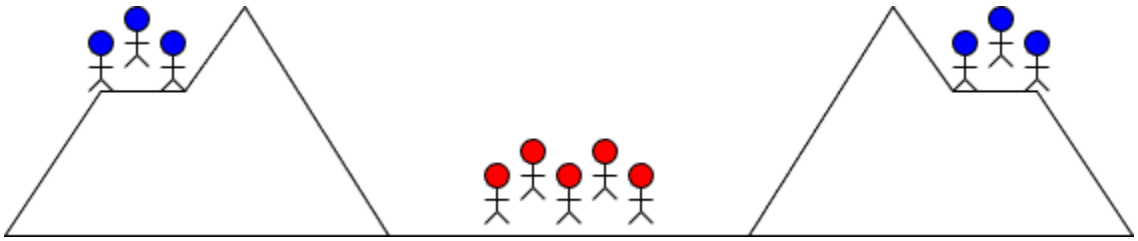
- Both sides must agree when to simultaneously attack
- If either side attacks alone, they suffer defeat
- Generals can only communicate via messengers and Messengers may get captured (unreliable channel)

Side A

- Send messenger: “Attack at dawn”
- What if messenger doesn’t make it?

Side B:

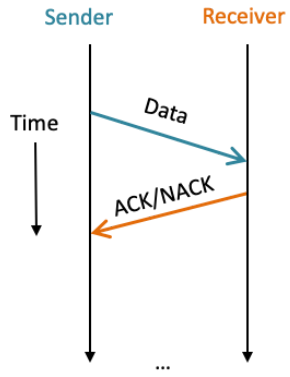
- Agrees to send (poor messenger) back with an acknowledgment: “I delivered message”



Q1. In the “two generals problem”, can the two armies reliably coordinate their attack? (using what we just discussed)

A. Yes (explain how)
B. No (explain how)

Q2. Could we do this with just ACKs or just NACKs?

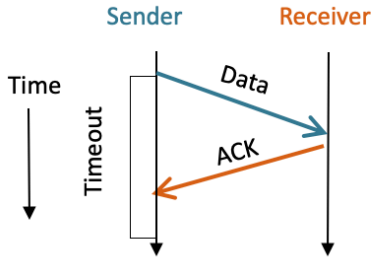


Error detection mechanism: checksum

- Data good – receiver sends back ACK
- Data corrupt – receiver sends back NACK

A. No, we need them both
B. Yes, we could do without one of them, but we’d need some other mechanism.
C. B.Yes, we could get by without one of them.

Adding timeouts might create new problems for us to worry about. How many? Examples?



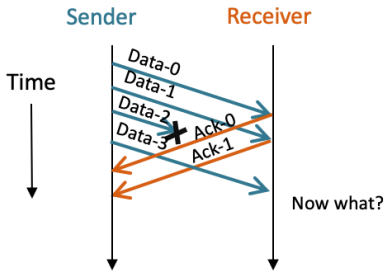
A. No new problems (why not?)
B. One new problem (which is..)
C. Two new problems (which are..)
D. More than two new problems (which are..)

What is our link utilization with a stop-and-wait protocol?

A. < 0.1 %
B. ≈ 0.1 %
C. ≈ 1 %
D. 1-10 %
E. > 10 %

System parameters:
Link rate: 8 Mbps (one megabyte per second)
RTT: 100 milliseconds
Segment size: 1024 bytes

What should the sender do here?



What information does the sender need to make that decision?

What is required by either party to keep track?

A. Start sending all data again from 0.
B. Start sending all data again from 2.
C. Resend just 2, then continue with 4 afterwards.