Internet Engineering

241-461

Robert Elz

kre@munnari.OZ.AU
kre@coe.psu.ac.th

http://fivedots.coe.psu.ac.th/~kre

The Window (2)

- Parameters
  - Window Size
  - Window Start
    - ACKnowledgment number
      - (Next Expected)
- Sequence Number
  - >= Window Start
  - < Window Start + Window Size
    - For all sequence numbers in packet

The Window (3)

- Window Start
  - Next Expected (ACK)
- Window Size
  - In all packets
    - Sender's Window Size

Used for packets in other direction

  - Can change packet to packet
    - Smaller as available buffer reduced

As packets arrive

    - Larger as more buffer available

As application removes data

- Sender Sequence number
  - Must be inside window
  - Sender stops sending new data

when it transmitted all sequence numbers in window

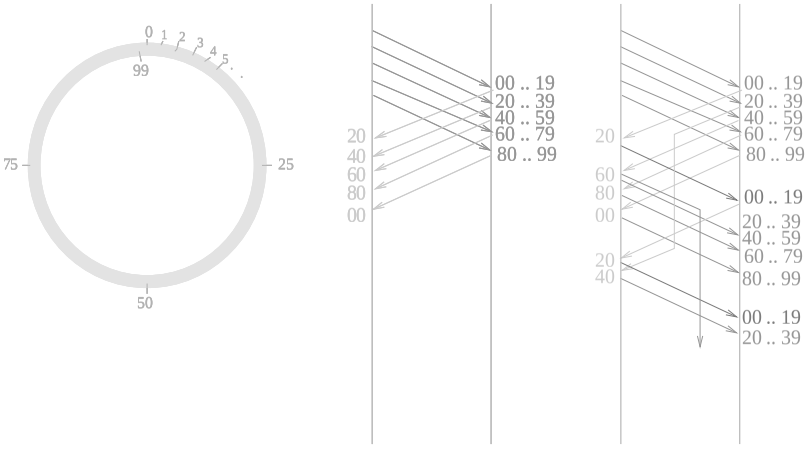
    - retransmit older data still OK
- Window size can be zero
  - No data can be sent

TCP Notes

- TCP connections are bi-directional
  - Every packet has sequence & acknowledgement numbers
    - Sequence number indicates which data are in this packet (if any)
    - Or where in sequence next data will go
    - Acknowledgement indicates which data are expected to be received next
  - Can acknowledge data received,
    - and send answer (any reply)
    - in the same packet
- Window size sent in every packet
  - Can be varied as buffer space at receiver varies

TCP Sequence Space

- Problem with reusing sequence numbers

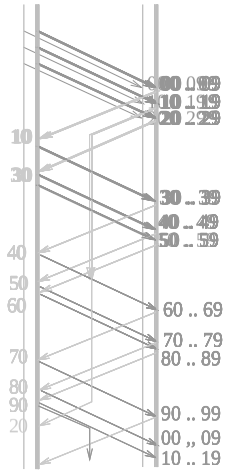


TCP Sequence Space (2)

- The window helps avoid this problem

TCP Sequence Space (3)

- But is not enough by itself ...



  - Old ACK can always cause problems
  - Need a way to make it go away
    - Limit on packet lifetime ... in network layer

TCP Connections

- Application needs to communicate with peer
  - establish data path between applications
  - TCP connection
- Must identify remote application
    - Transport Protocol Address
  - System address
    - To come later
  - Application Identifier
    - Port Number
- Sender also needs address
    - For returning packets
  - Another Port Number

TCP Connections (2)

- Connection Identity
  - (1) Network Address (system 1)
  - (2) Network Address (system 2)
  - (3) Port Number (system 1)
  - (4) Port Number (system 2)
- Port Number
  - 16 bit identifier
    - 0 .. 65535
- Each different set of identifiers
  - identifies a different connection

TCP Connections (3)

- TCP Model
  - All connections created when TCP created
    - 1980 ???
    - For IPv4:

$$2^{32} * 2^{32} * 2^{16} * 2^{16}$$

  - Each system (with 1 network address)

$$2^{32} * 2^{16} * 2^{16}$$

    - 18,446,744,073,709,551,616
  - Remain forever
    - mostly dormant

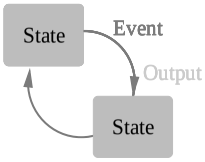
TCP Connections (4)

- Each connection has a state
  - Each of the 18,446,744,073,709,551,616 conns/addr
  - Closed
  - Opening
  - Data Transfer
  - Closing
    - Simplified model
- TCP uses Finite State Machine
  - Defines how states are manipulated

Finite State Machine

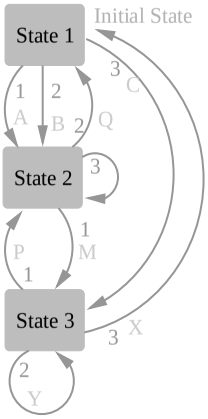
- More formal method of speificiation
  - Often depicted using drawing
- Some number of states defined
  - drawn as circles or rectangles
- In each state
  - specific events can occur
    - inputs
    - timeouts
  - cause transition to another state
  - cause output action to occur

FSM Basics



- The FSM is in some state
- An event occurs
  - drawn beside a line
  - shows transition to a new state
- Some output may accompany transition
- Transition may return to the same state
  - Or may transition to another state

FSM (example)



Three States 1 2 and 3

Three events that occur 1 2 and 3

Several output actions

State 1 is the initial state

For input events

2 1 3 1 3 2 3 2 1 1 3

What states does FSM pass through?

What output actions are performed?