Token Ring

CS570

presented by Leon Poutievski

Problem

- What problem are we solving?

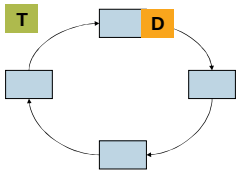
  Allow a group of stations to communicate as if directly connected
  - but with linear cost instead of quadratic

Basic Concept

- "Token" = permission to transmit
- Required some way of "passing the token" from one station to the next
- Assumptions
  - same connection
  - same algorithm

Token Ring

- Unidirectional channels
- Ring is viewed as a single shared medium



Token-Passing

- Only one station holds the token.
  Only that station may transmit.
- Passing the token = transmitting a special frame (bit pattern) to the next station
- Topologies
  - Bus: IEEE 802.4
  - Ring: IEEE 802.5,
    Fiber Distributed Data Interface (FDDI)
- Token Holding Time (THT) =
  maximum time a station may hold a token

Ring Latency

- Each station regenerate each received bit
- Station Delay: time needed for a station to read and regenerate
  - Station Delay <= THT
- Ring Latency: time it takes a bit to go around
  = total propagation delay + sum of station delays

Ring Latency

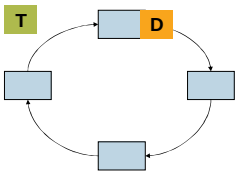
- Normalized Ring Latency (NRL):
  number of bits "stored" on the ring
  = Ring Latency / duration of a bit
  - must be larger than the token
- Token Rotation Time (TRT): time it take for a token to go around
  TRT <= ActiveNodes x THT + Ring Latency

Operation

- Nothing to transmit (repeat mode)
  - Every bit in transmitted without modification
- Ready to transmit
  - Wait for the token
  - Recognize the token, remove it from the ring (actually flips bits in the Start-of-Frame Sequence)
  - Transmit data (no more that THT)
  - Replace the token on the ring

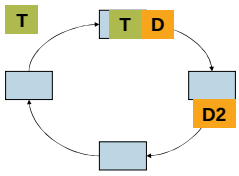
Token Replacement

- Same-frame
  - After the last bit of the frame is received by the sender
  - Used in 802.5



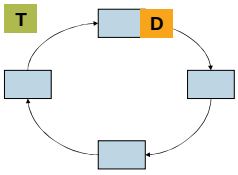
Token Replacement

- Multiple-token
  - After the last bit of the last frame is transmitted



Token Replacement

- Single-token
  - After the last bit of the last frame is transmitted AND
  - start of sequence is received by the sender



802.5 Frame Format

SD	AC	FC	DA	SA	Data	FCS	ED	FS

- SD = Start Delimiter = JK0JK000
- AC = Access Control = PPPTMRRR
  - PPP = Priority
  - T = Token bit
  - M = Monitor bit
  - RRR = Reservation bits

802.5 Frame Format

SD	AC	FC	DA	SA	Data	FCS	ED	FS

- FC = Frame Control = FFZZZZZZ
  - FF = type, one of
    - 00 = MAC frame 01 = LLC frame
- ED = End Delimiter = JK1JK1IE
  - I = Intermediate bit (0 = last frame, 1 = more)
  - E = Error detection bit

802.5 Frame Format

SD	AC	FC	DA	SA	Data	FCS	ED	FS

- DA = Destination Address
- SA = Source Address
- FS = Frame Status = ACxxACxx
  - A = Address Recognized
  - C = Frame Copied

802.5 Token Format

SD	AC	ED

- SD = Start Delimiter
- AC = Access Control
  - Token bit in AC field is 0
- ED = End Delimiter

Performance

- Where do we waste bandwidth?
  - Waiting for token

  (similar time need in polling methods)
- Absolute throughput =

$$\frac{\text{time spent transmitting DATA}}{\text{time spent transmitting DATA + time spent waiting for the token}}$$

Performance

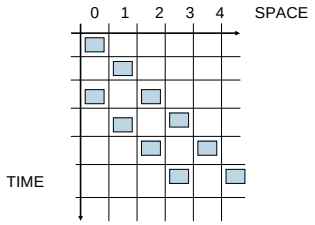
- Parameters
  - Transmission rate
  - Ring Latency
  - Frame size
- Token replacement policy affect efficiency
- Performance is similar to that of the polling methods

Ring Maintenance

- What if token is lost?
- Solution: Ring Monitor
  - Makes sure there is always a token in the ring
  - Detects a missing token when it does not see the token for

    NumStations x THT + Ring Latency
  - Creates a new one, if the token is missing
  - Any node can become a ring monitor

Time-Space Diagram



FDDI

- Fiber Distributed Data Interface (FDDI)
- Uses fiber
- Dual ring
  - second in the reverse direction, for backup only

FDDI: Timed Token Algorithm

- Target Token Rotation Time (TTRT) – desired maximum time of token appearances at any station
  - If observed TRT > TTRT, then token is late station does not transmit data
  - If observed TRT < TTRT, then token is early node can hold token for (TTRT – TRT)
- TTRT bidding is combined with monitor election