



# Flow and Congestion Control

**CS2520/TELCOM2321**  
**Wide Area Networks**

**Prof. Taieb Znati**  
**Department Computer Science**  
**Telecommunication Program**

# Flow Control Objectives

- Limiting delay and buffer overflow.
- Fairness.
- Simplicity and ease of implementation.
- Efficiency, using the least possible network resources in terms of bandwidth and buffers.
- Scalability, when used by a large number of source nodes.
- Often these objectives are mutually contradictory
  - Simplicity and Fairness
    - ◆ Trade-offs must be considered.

# Delay and Buffer Control

- Long delays due to queue build up cause slow acknowledgements.
  - Slow acknowledgements force source nodes to retransmit packets mistakenly believed lost.
- Retransmission causes buffer overflow to occur and packets to be discarded.
  - Discarded packets cause retransmission.

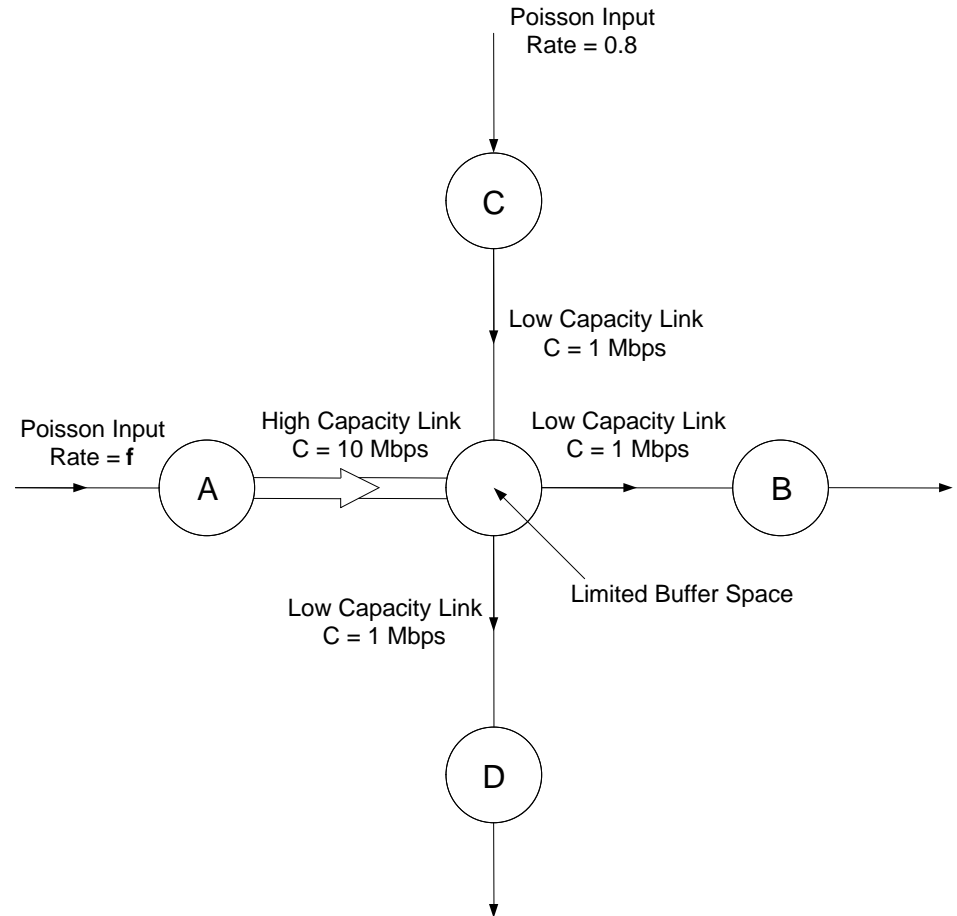
# Throughput Degradation

For small  $f$ , the buffer rarely fills and the throughput is  $0.8 + f$

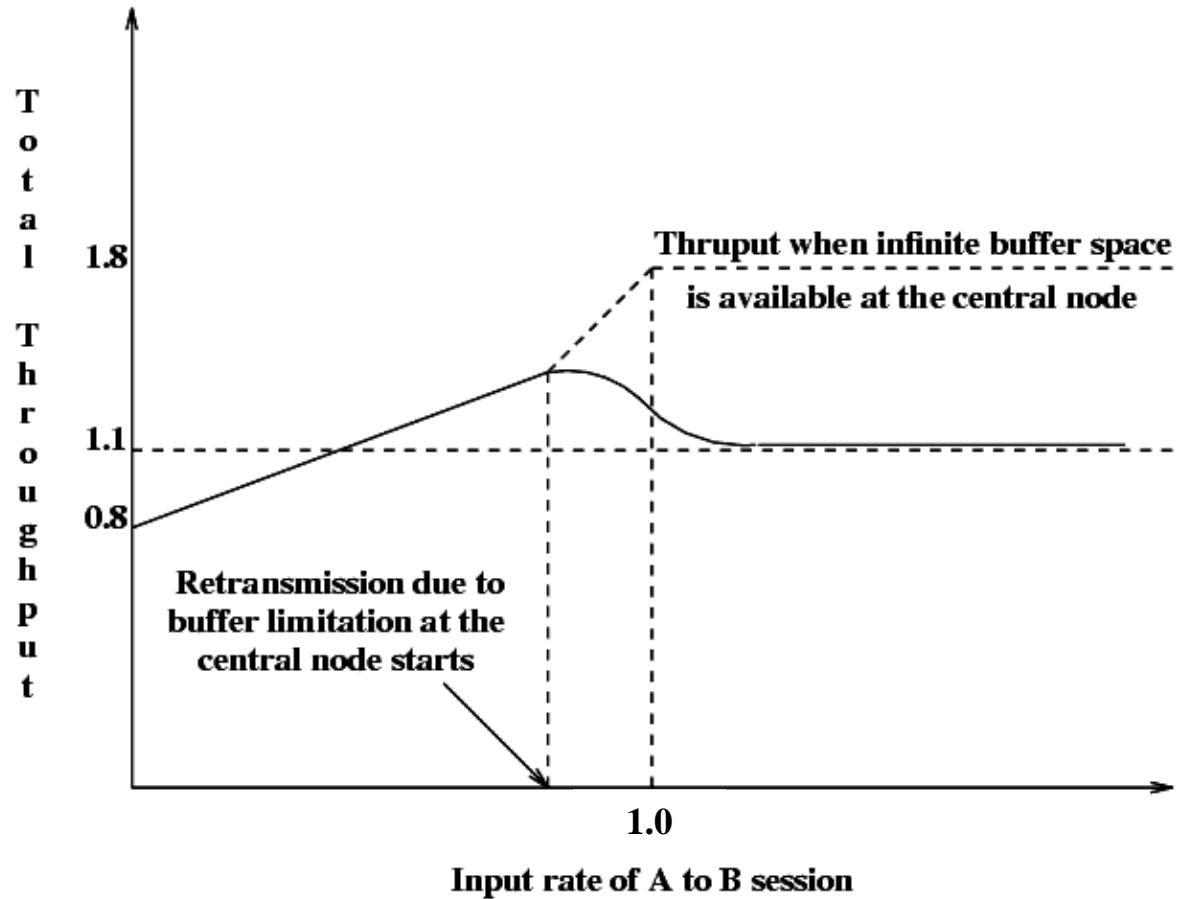
For  $f$  close to 1, the node transmits at full capacity and the buffer is almost full.

A transmits 10 times faster than C and has 10-fold greater chance for a buffer.

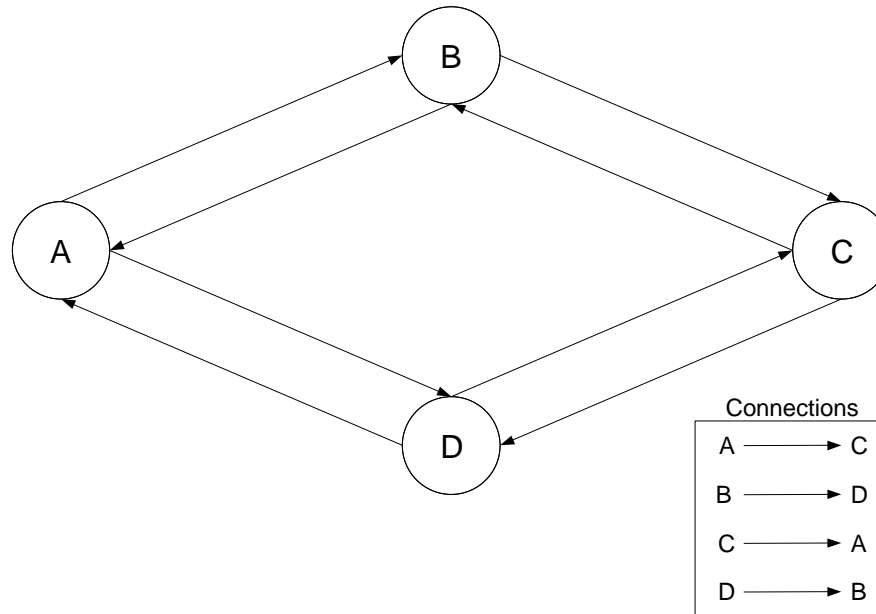
C is busy most of the time retransmitting packets.



# Session Input Rate

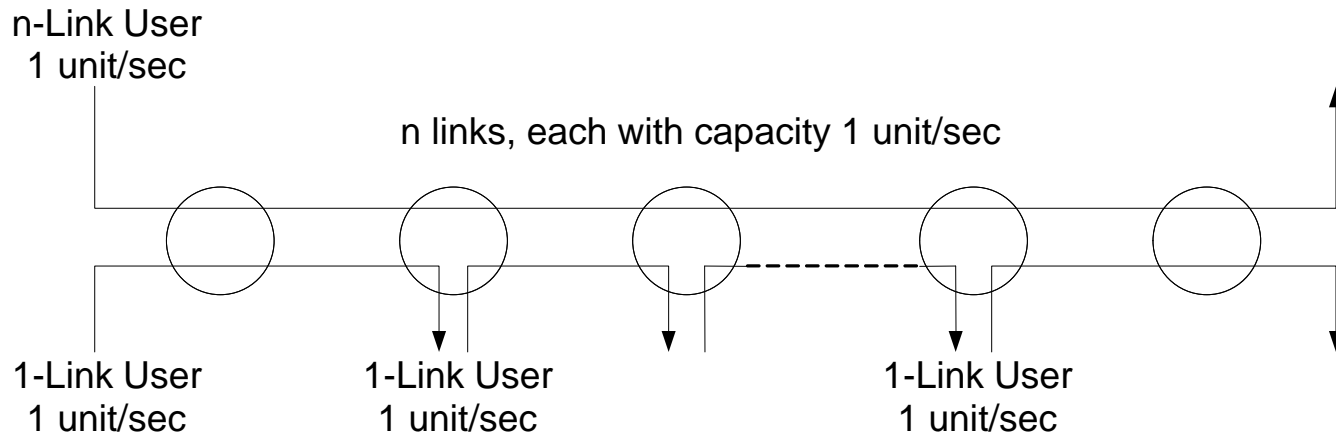


# Deadlock



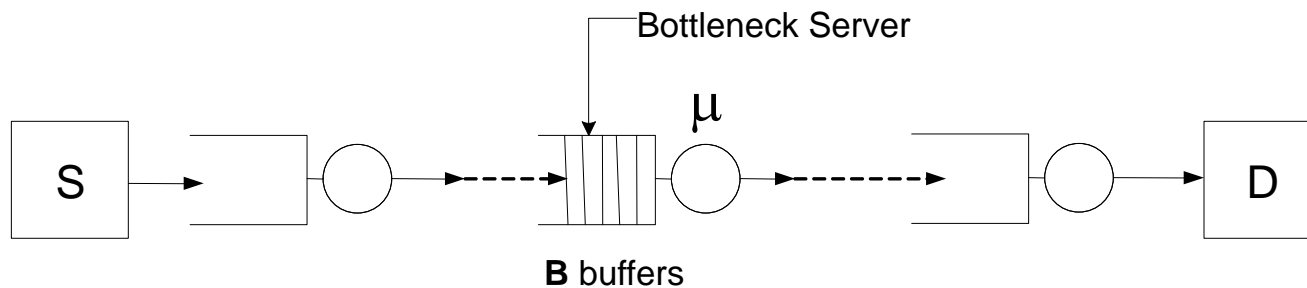
All buffers are full at A, B, C, and D.  
Deadlock due to buffer overflow.

# Fairness



- A maximum throughput of  $n$  units/sec can be achieved if  $n$ -Link user is blocked. Allocating equal rates of  $1/2$  unit/sec to all session achieves a throughput of only  $(n+1)/2$  units/sec.
  - Allocating equal resources to all sessions results in  $n/(n+1)$  units/sec to single link users, and  $1/(n+1)$  to  $n$ -link user.

# Flow Control Model



- Flow control can be viewed, as matching rate, between source and bottleneck, with delays
  - Bottleneck server's current drain rate is known only after a round trip time.



# Flow Control Classification

- Closed loop flow control
  - A source dynamically adjusts its flow in quest of its current share of resources based on network feedback.
    - ◆ Congestion detection and recovery.
- Open loop flow control
  - A source describes its traffic during call establishment and the network reserves corresponding resources, if available.
  - Source shapes its traffic to match its traffic descriptor.
    - ◆ Congestion avoidance.
- Hybrid flow control
  - Minimum amount of resources is reserved and other resources are allocated as they become available.

# Closed Loop Flow Control Classification

- Closed loop schemes can be further categorized in three different ways:
  - Explicit or Implicit Feedback.
  - Window based or Rate based.
  - End-to-End or Hop-by-Hop.

# Closed Loop Flow Control Feedback Strategy

- In explicit feedback, control messages are sent back from the point of congestion to the source of congestion.
  - More precise control.
  - Requires both communication and computation overhead.
- In implicit feedback, the source infers the existence of congestion based on local observations, such as the Round Trip Time (RTT).
  - Less accurate control.
  - Minimum overhead.

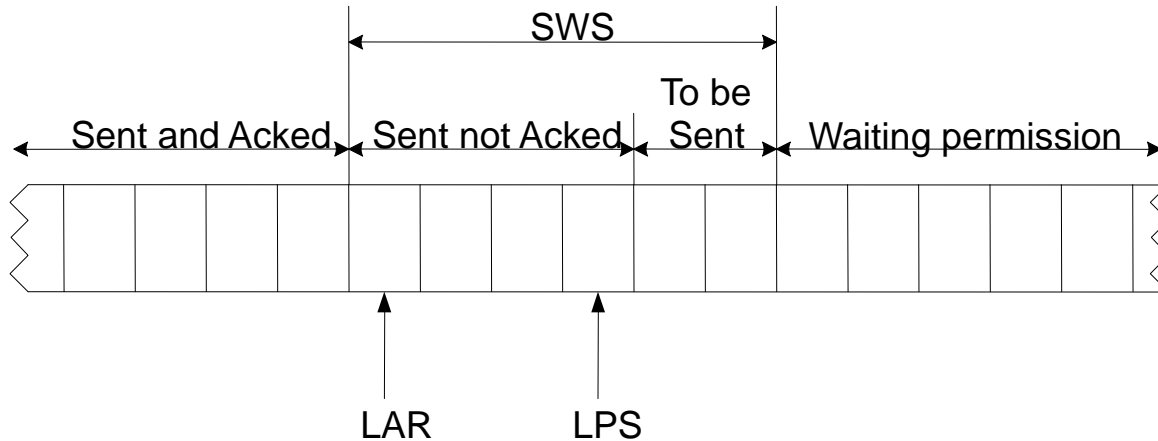
# Closed Loop Flow Control Strategy

- Window based flow control
  - An upper bound on the number of data units sent by a sender and not acknowledged by the receiver is imposed.
    - ◆ Referred to as a Window.
  - Static or dynamic window schemes.
- Rate based flow control
  - Source rate is directly controlled to adopt to current resource availability
    - ◆ Source regulates its traffic by sending packets every  $1/R$  sec, where  $R$  is the currently allowed transmission rate.

# Closed Loop Flow Control Control Level Strategy

- End-to-End flow control
  - Control is exercised between the end points of a connection.
    - ◆ Very sensitive to RTT.
- Hop-by-Hop flow control
  - Control is applied between pairs of adjacent nodes
    - ◆ Typically more effective, since control delay is smaller.
    - ◆ Router complexity increases.

# Sliding Window Flow Control Static Window Size



LAR : Last Ack Received

LPS : Last Packet Sent

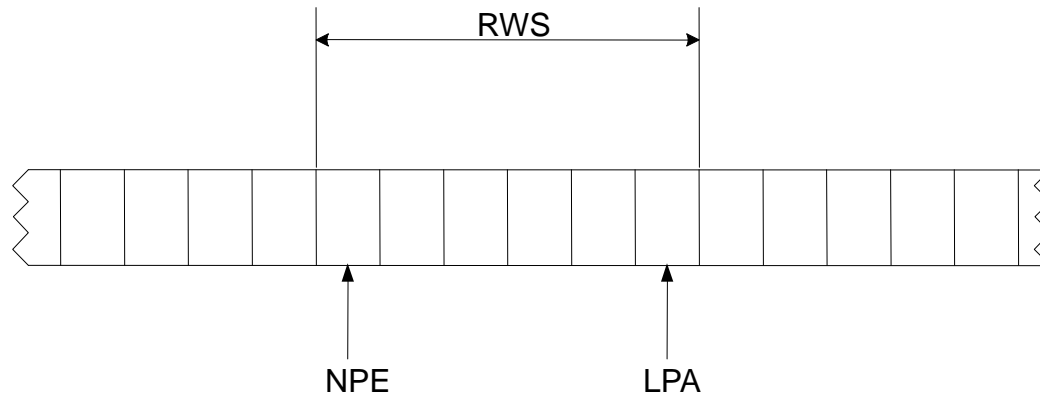
SWS : Sender Window Size (  $< (MaxSeq + 1) / 2$  )

# Sliding Window Management

## Sender Side

- The sender maintains the following invariant
  - $LPS - LAR + 1 \leq SWS$
- LAR is moved to the right after each ACK received.
- A timer is associated with each packet
  - Packet is retransmitted if corresponding timer expires.

# Sliding Window Management Receiver Side



NPE : Next Packet Expected

LPA : Last Packet Accepted

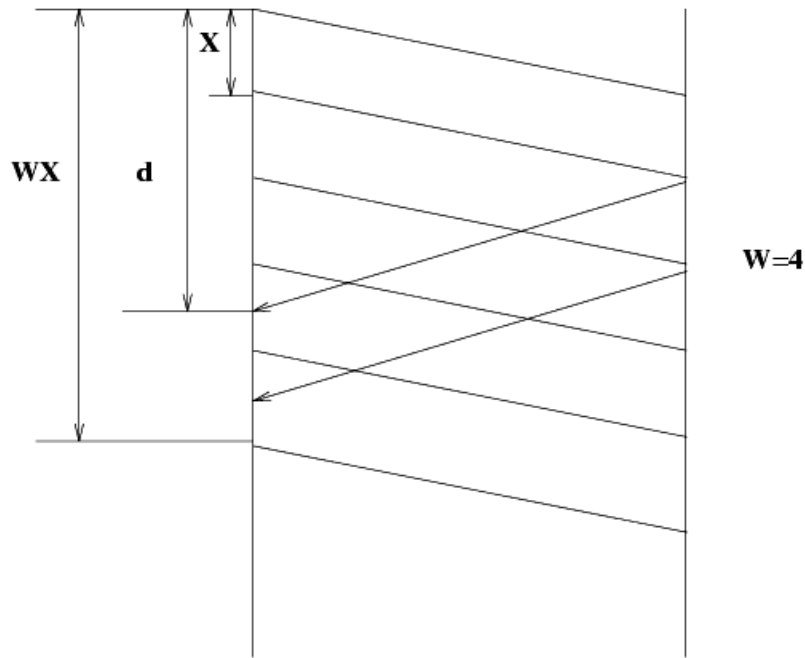
RWS : Receiver Window Size



# Sliding Window Management Receiver Side

- The receiver maintains the following invariant:
  - $LPA - NPE + 1 \leq RWS$
- On packet reception, the receiver takes the following action
  - If (  $SeqNum < NPE$  or  $SeqNum > LPA$  ) then discard the packet.
    - ◆ Packet outside the window.
  - If (  $NPE \leq SeqNum \leq LPA$  ) then accept packet
    - ◆ Send ACK, possibly cumulatively.

# Window Flow Control ( $d \leq WX$ )

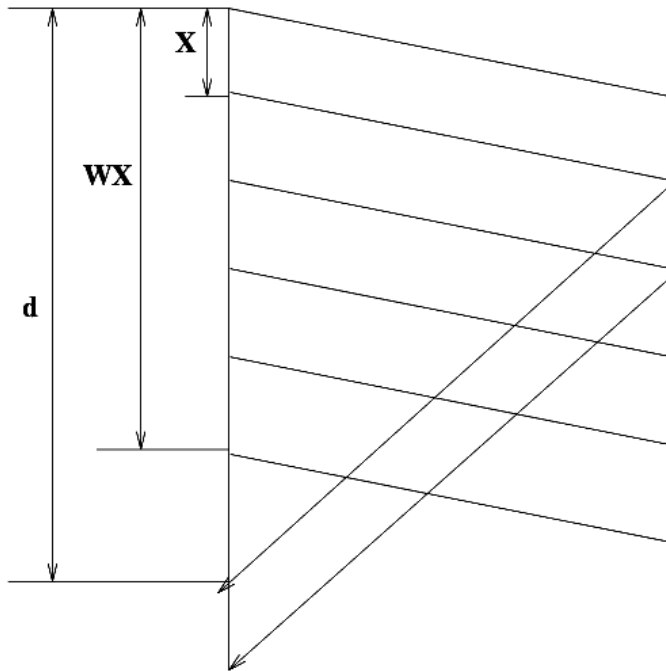


*Max Rate = 1 / X packets / sec*

**RTT = d**

**WX = Time to transmit full window**

# Window Flow Control ( $d > WX$ )



**RTT =  $d$**

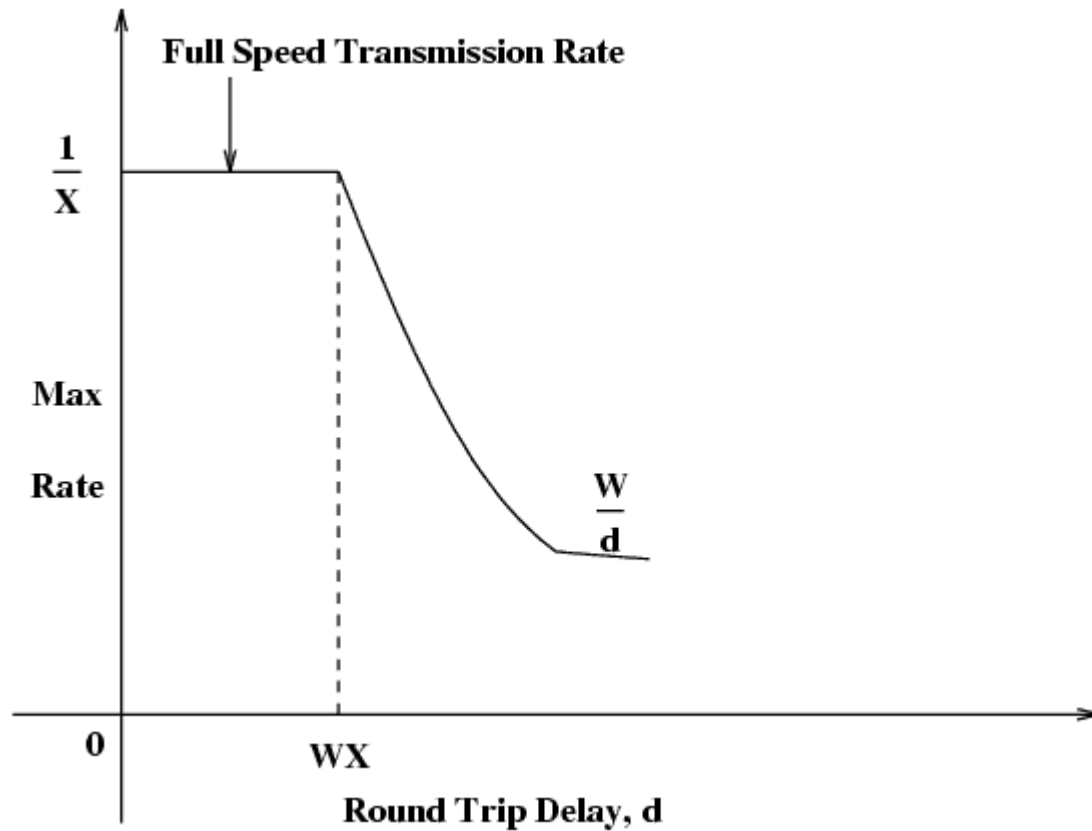
**$WX$  = Time to transmit full window**

*Max Rate =  $W/d$  packets/sec.*

In general,

*Max Rate =  $\min \{1/X, W/d\}$  packets/sec.*

# Window Flow Control Rate Efficiency



# Window Flow Control

## Rate Efficiency

- Determining the proper window size is difficult
  - It is desirable to make the window size small to limit the number of packets in the network.
    - ◆ Avoids congestion and large delays.
  - It is also desirable to make window sizes large to allow full speed transmission and maximal throughput under light-to-moderate traffic conditions.
- Dynamic window sizes adjustable to congestion.

# Hop-by-Hop Flow Control

- Receiver avoids the accumulation of a large number of packets in its memory by reducing the rate at which it returns acknowledgements.
- If node  $i$ 's  $W$ -packet buffer is full, then  $i$  sends an ACK to  $i-1$  only after it receives an ACK from  $i+1$ 
  - Backpressure phenomenon

# Hop-by-Hop Flow Control Evaluation

- Packet will be uniformly distributed along the path
  - End-to-End flow control tends to cause packets to concentrate at congested link.
- Fairness problem may occur when links with relatively large propagation delays are involved.
  - Weighted round-robin scheduling can be used to address this problem.

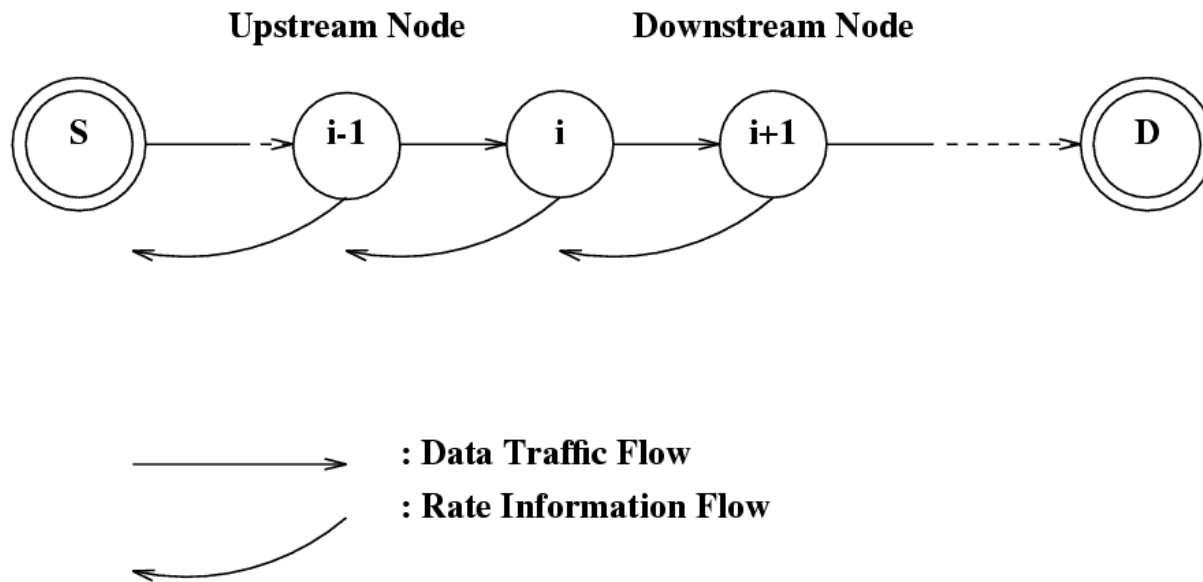
# Hop-by-Hop Flow Control

## Case Study: Mishra-Kanakia Scheme

- Every network node periodically samples a sending rate and a buffer occupancy for each connection passing through it.
  - Sampled information is sent to upstream nodes.
  - Upstream node uses information to update its transmission rate.
- In case of congestion, rates are throttled back all the way to the source.



# M-K Scheme



# M-K Flow Control Mechanism

- In update,  $k$ , an upstream node receives
  - $\chi(k)$ , downstream node buffer length.
  - $\mu(k)$ , downstream node service rate.
- Source estimates  $v(k+1)$ , future sending rate :
  - $v(k+1) = \alpha \times v(k) + (1 - \alpha) \times \mu(k)$
- Source estimates  $y(k+1)$ , *buffer occupancy of downstream node*
  - $y(k+1) = \chi(k) + (\lambda(k-1) + \lambda(k)) - (v(k-1) + v(k))$

# M-K Flow Control Mechanism

- Sending rate is computed as :

$$\lambda(k+1) = v(k+1) + ( \beta / R_{hop} ) ( B - y(k+1) ), \text{ where}$$

- $B$  is downstream buffer setpoint.
- $R_{hop}$ , round-trip propagation delay along the hop.
- $0 \leq \beta \leq 1$ , a damping factor that controls the time taken by the system to reach the desired value.

# Closed Loop Dynamic Rate

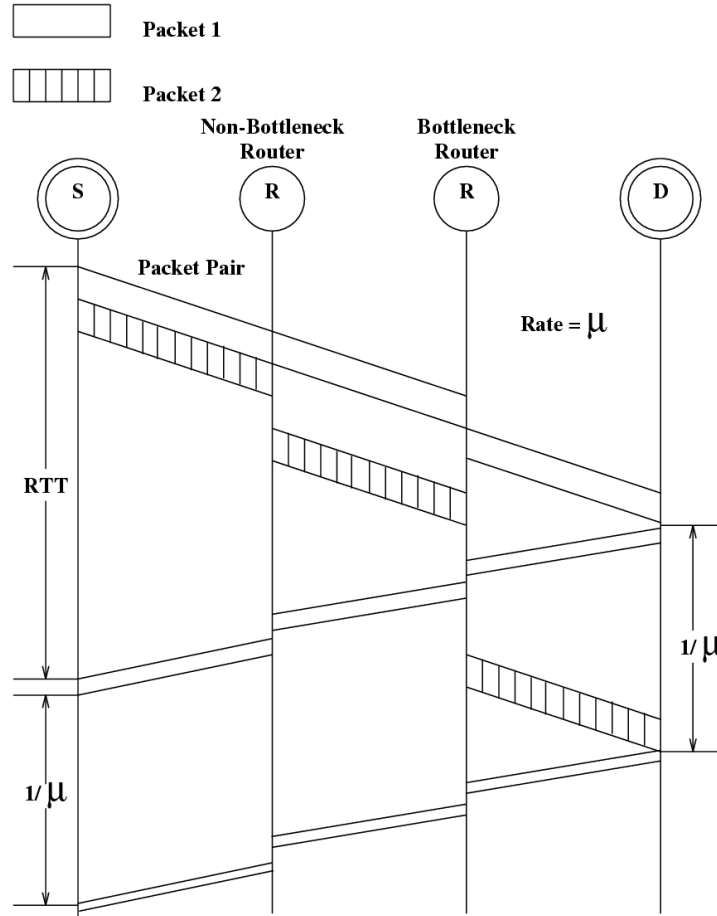
- Source adjust their rates dynamically in response to congestion.
  - Losses and retransmission, which affect window based schemes, do not affect rate adjustment directly.
    - ◆ Decoupling of error and flow control simplifies the design of both components.

# Dynamic Rate

## Case Study : Packet-Pair

- Scheme's objectives :
  - Predict bottleneck router's service rate.
  - Correct past predictions, if incorrect.
  - Adjust actual transmission rate to reach bottleneck setpoint in one RTT.
- Packet-pair explicitly assumes that scheduling is round-robin.

# Packet-Pair Scheme



# Packet-Pair Scheme

- Source computes a smoothed average of the bottleneck server

$$\hat{\mu}(k+1) = \alpha \times \hat{\mu}(k) + (1-\alpha) \times \mu(k)$$

- Source estimates the number of packets,  $Q$ , in the bottleneck buffer.

$$Q = S - R \times \hat{\mu}(k+1)$$

- $S$  is the number of outstanding packets, (i.e., send not acked)
- $R$  is round-trip propagation delay.
- Source adjusts its actual transmission rate to reach set point,  $B$ , in one  $R$

$$\lambda(k+1) = \hat{\mu}(k+1) + \frac{(B-Q)}{R}$$