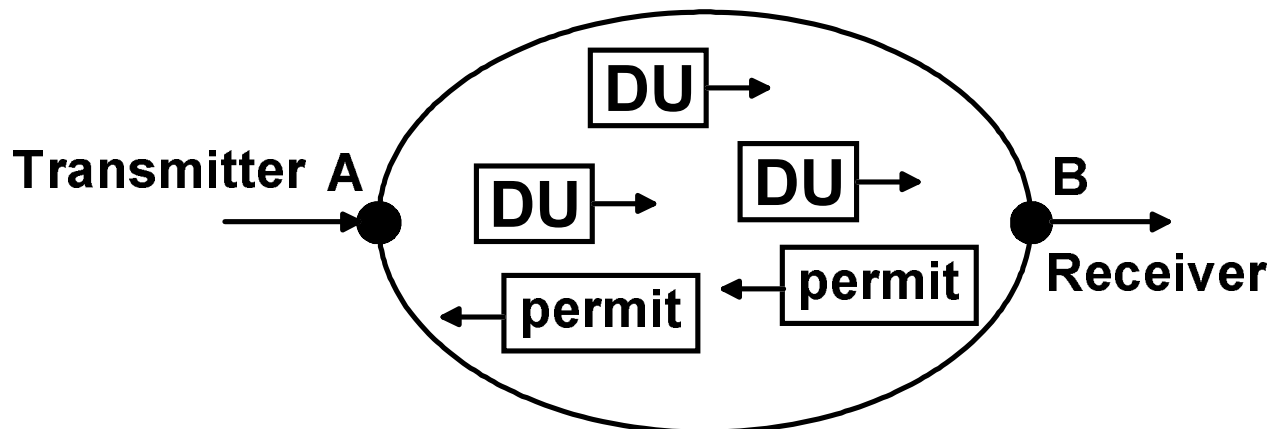


Window Flow Control

Poses an upper bound (the “window”) on the maximum number of data units of a session that can be sent by A and are not yet known by A to have been received by B.



$$(\# \text{ DU's}) + (\# \text{ permits}) \leq W_{AB}$$

B notifies A that it has disposed of a data unit by sending a permit to A; upon receiving a permit, A can send new data unit.

(Data units may be packets, bytes, messages, etc..)

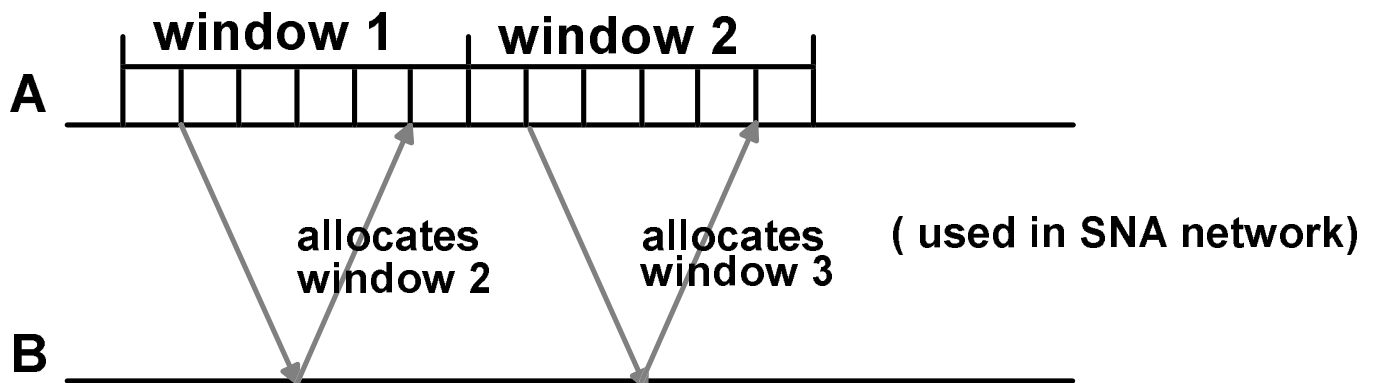
End-to-End Windows

The window exists between source machine & destination machine (or between source & destination node of subnet)

Variations:

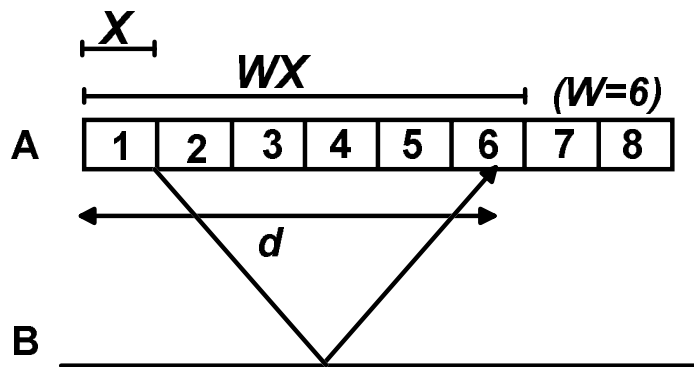
a) A request number RN is sent acknowledging all packets before RN and sliding window to $[RN, RN+W-1]$.

b) When the first packet of the window is received, a permit is sent allocating entire new window



Only one permit per window is sent (saves # permits)

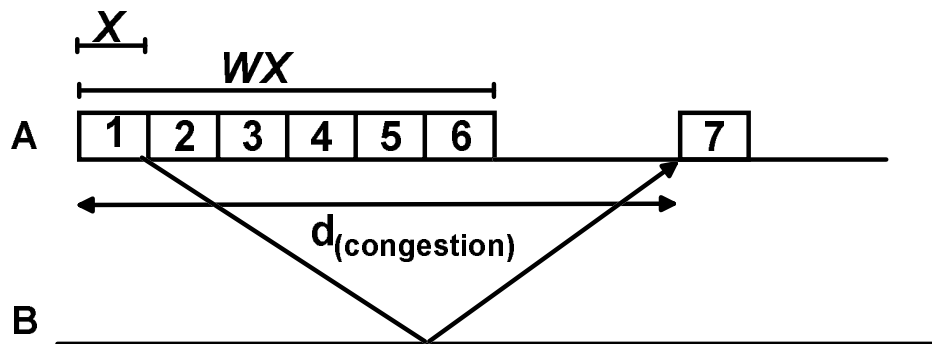
X = expected packet transmission time
 W = window size
 d = end-to-end delay (roundtrip)



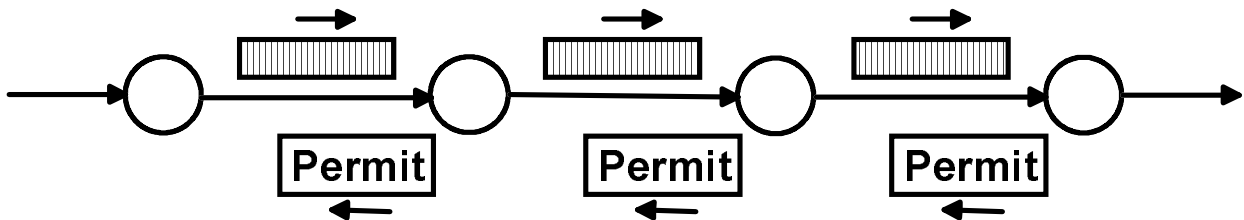
We would like to inhibit traffic only under congestion; thus we should have

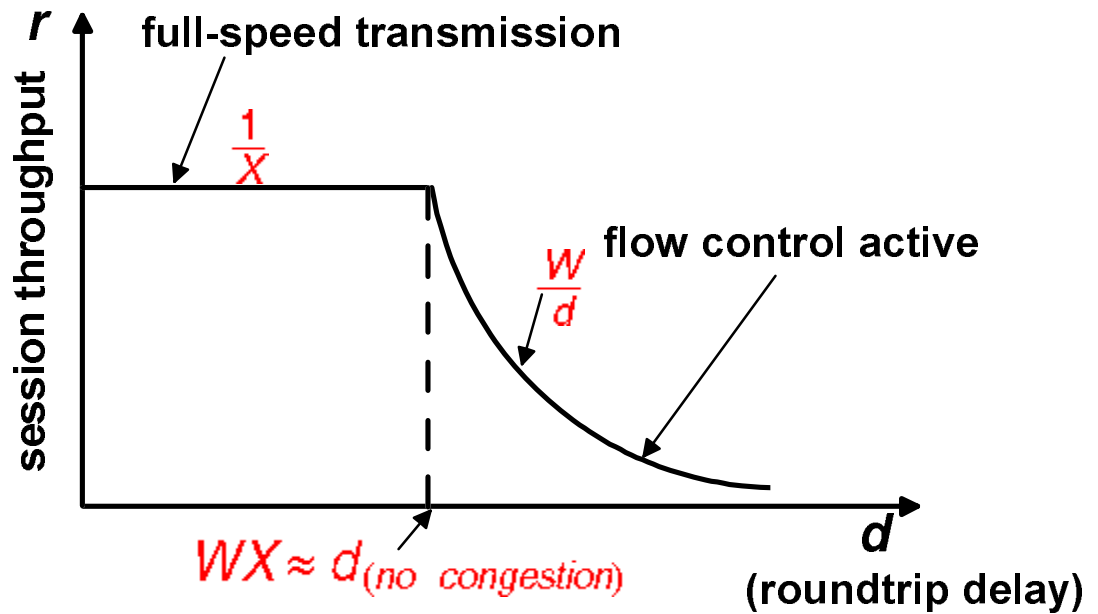
$$WX > d_{(\text{no congestion})}$$

With increasing congestion, d increases, and the window inhibits flow:



In low speed networks W should be chosen larger than h (usually between h and $3h$), where h is the number of hops between source and destination.





$$r = \min \left(\frac{1}{X}, \frac{W}{d} \right)$$

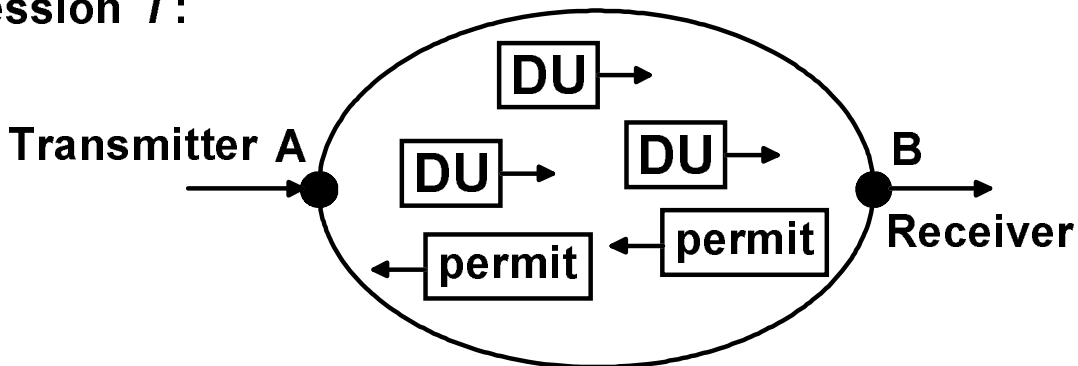
r : session rate (packets/sec), (assuming packets always available at transmitter)

The smaller W is, the more r is throttled when congestion arises, and the faster the reaction to congestion is.

Effect of the number of sessions

Suppose there are n actively controlled sessions with windows W_1, W_2, \dots, W_n .

Session i :



$$(\# \text{ DU's}) + (\# \text{ permits}) \approx W_i$$

Let $\beta_i = \frac{\# \text{ DU's}}{(\# \text{ DU's}) + (\# \text{ permits})} \in (0, 1)$

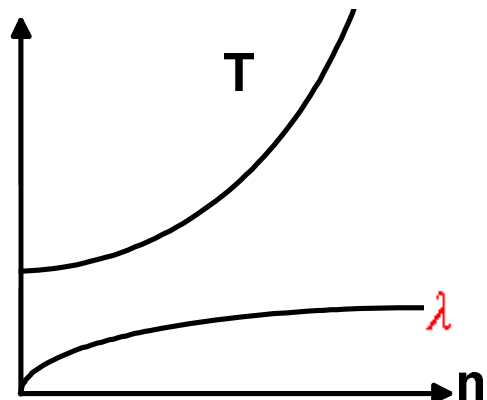
By Little's Theorem: $\beta_i = \frac{(A \rightarrow B \text{ delay})}{(A \rightarrow B \text{ delay}) + (B \rightarrow A \text{ delay})}$

Total number of packets in the network = $\sum_{i=1}^n \beta_i W_i$

Let λ = total throughput of all sessions (= total packet acceptance rate)

Little: $T = \frac{\sum_{i=1}^n \beta_i W_i}{\lambda}$

Need to adjust window sizes with number of sessions



Problem with End-to-End Windows

- (a) Window size must change with the number of sessions.
- (b) Both throughput and delay have no guarantees
- (c) For high-speed sessions on high-speed networks, windows must be large.

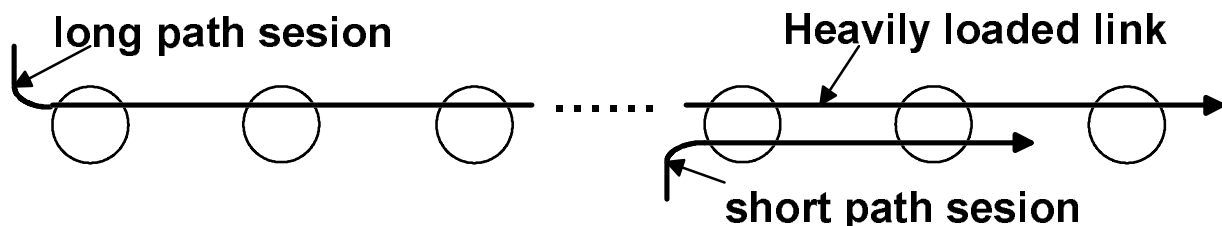
e.g.

$r = 1 \text{ Gb/s}, L = 3000 \text{ km},$

$c = \text{speed of light in fiber}, 450 \text{ bits/packe}$

$$\Rightarrow W \approx \frac{d_{no \text{ congestion}}}{X} = \frac{2L}{c} \cdot \frac{r}{450} \approx 100,000 \text{ packet}$$

- (d) Sessions on long paths with large windows are better treated than short path sessions \Rightarrow unfair to other sessions.

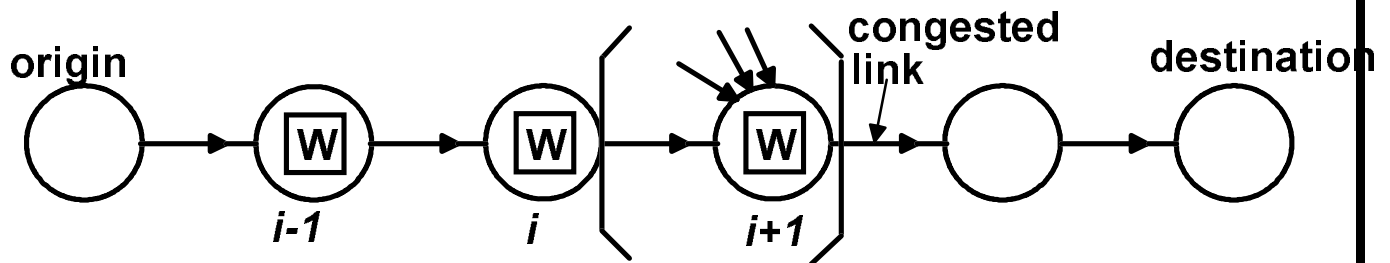


Node-by-Node windows

(for virtual circuits, e.g. TYMNET Network)

There is a separate window for each link on a VC path.

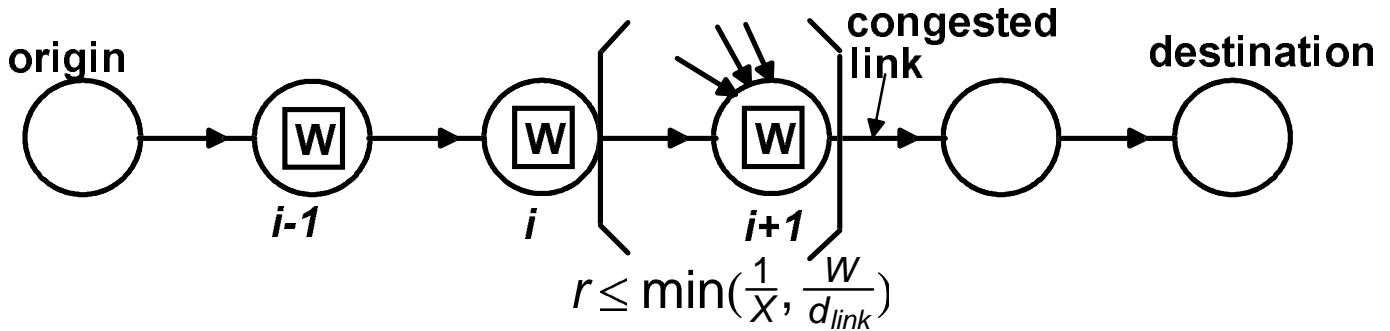
Each node can store up to W packets of a session.



- For a packet of a session to proceed from i to $i+1$, a buffer space must empty node $i+1$ (when this happens a permit is sent to i)
- This creates a window of size W at each link
- If a link becomes congested, packets remain in queue, and permits do not go back to previous node (“back-pressure”).

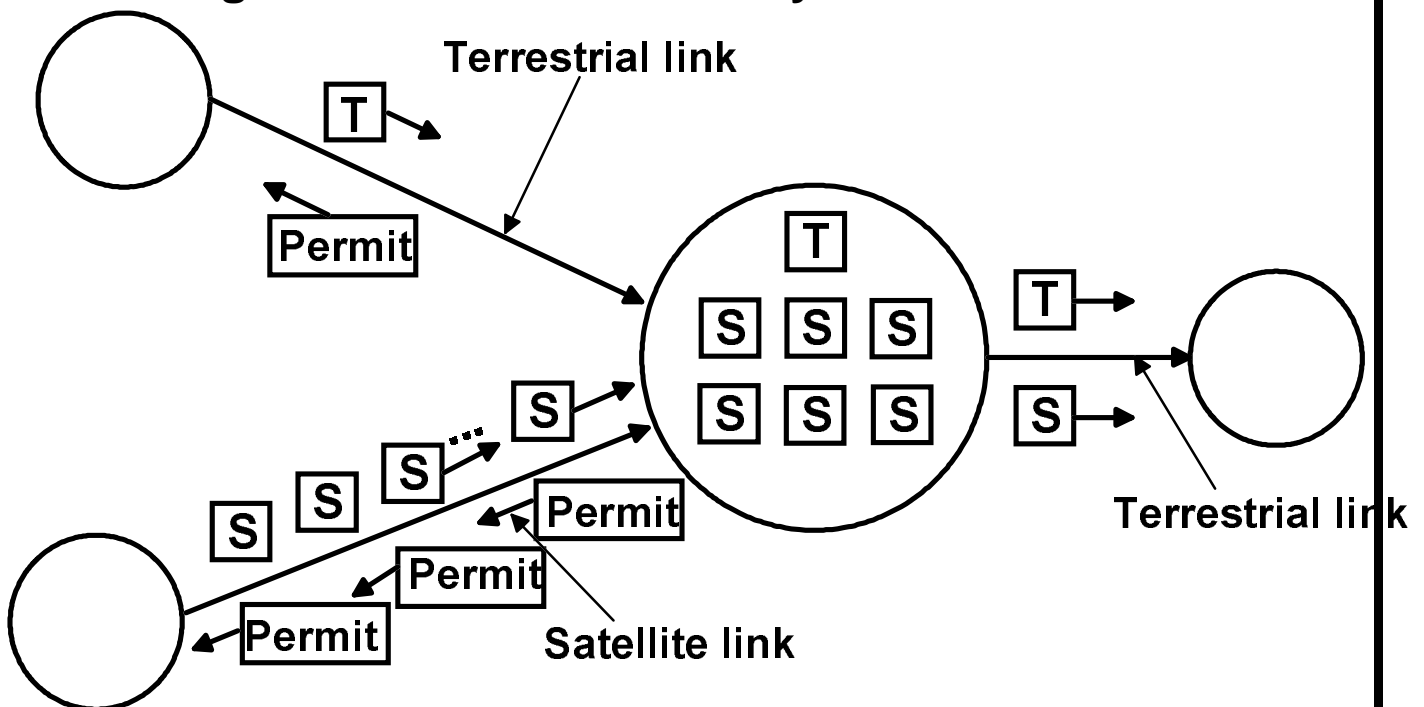
Advantages

- Smaller buffer requirements
- Reduces fairness problem



Disadvantages

- Fairness problem may still exist when satellite link (or other large propagation links) exist.
- Buffer requirements increase with the number n of sessions
- Still no guaranteed rate and delay.

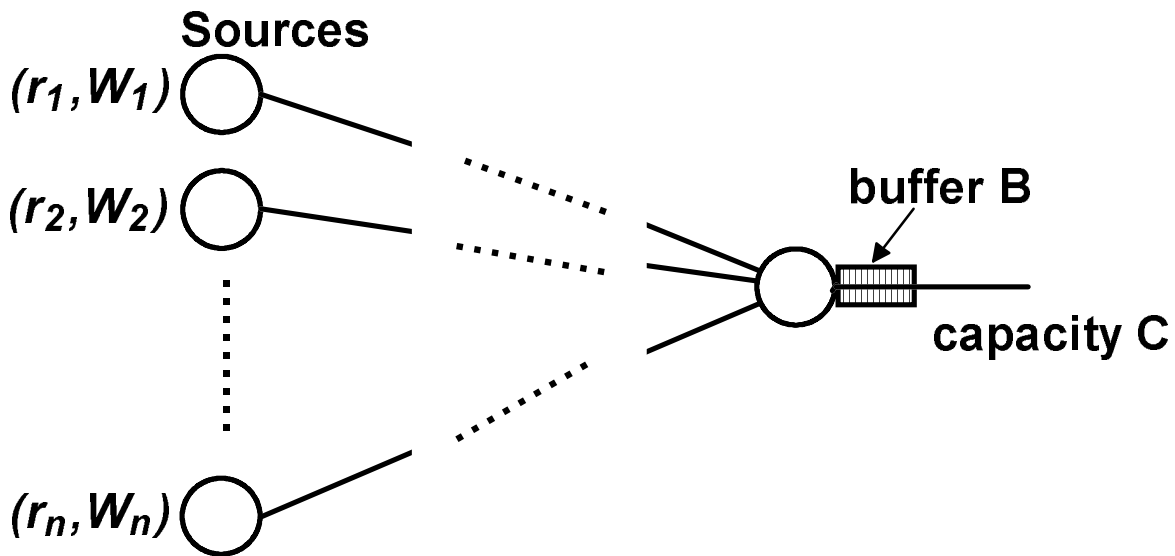


Rate-Based Flow Control

- **Integrated service networks try to provide guaranteed rate to some users, and thus cannot use window flow control.**
- **In addition, high speed sessions on high speed links require very large windows when using window flow control.**

With rate-based flow control, each session has some requested rate and some minimum rate (which has to be guaranteed)

If session rates have to be reduced, the rate reduction must be done “fairly”, while obeying the minimum rate requirements of each session.



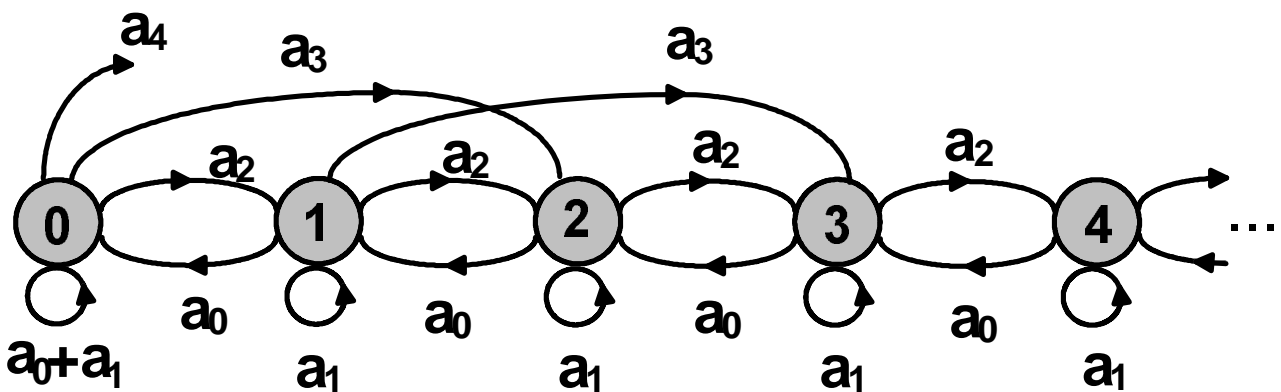
“For all practical purposes”, If $\sum_{i=1}^n r_i < C$ and $B \geq k \sum_{i=1}^n W_i$ (k : constant say 3), few packets are lost.

One can construct pathological examples in which queues are substantially larger than this.

Queueing Analysis of Leaky Bucket

View the system as a discrete time system with a state change each $1/r$ seconds.

Let states $i=0,1,2,\dots,W$ correspond to $W-i$ permits in the packet and no packets waiting. Let states $i=W+1,\dots$ correspond to an empty bucket and $i-W$ packets waiting.

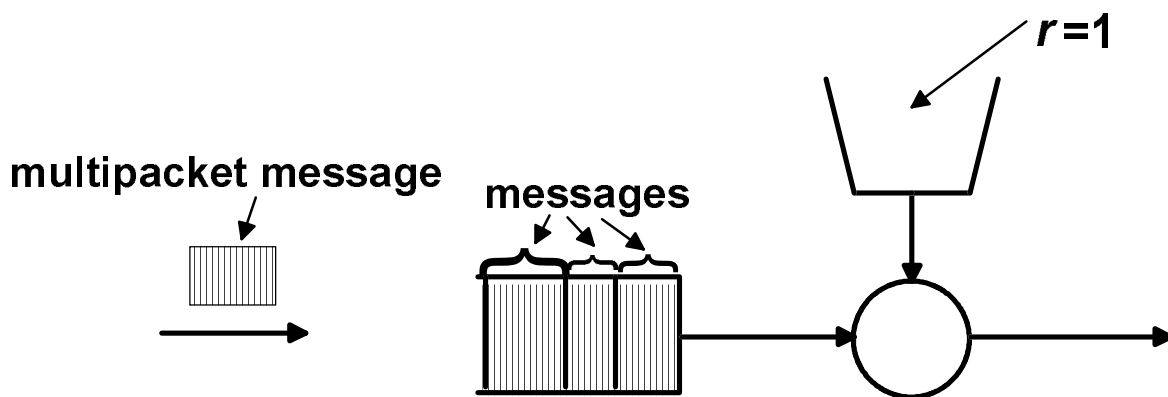


$$a_k = \frac{1}{k!} \cdot \lambda^k \cdot e^{-\lambda} \text{ (for Poisson packet arrivals)}$$

Note that this is the same as $M/D/1$ with slotted service.

Alternative model for Leaky-Bucket Scheme

A more realistic model is that of Poisson distributed message arrivals, and a geometrically distributed number of small packets per message



Let $\mu = \frac{1}{L}$, where L is the mean number of packets per message.

Thus μ is the service rate at which messages are served when the bucket is empty.

Let $Y(t) = (\# \text{ packets}) + (\# \text{ empty bucket positions})$

$Y(t)$ then decreases at rate 1, except when an arrival occurs, and then $Y(t)$ increases by the number of packets arriving (i.e. the time required to serve the arrival)

$Y(t)$ is then the unfinished work in fictitious $M/m/1$ queue!