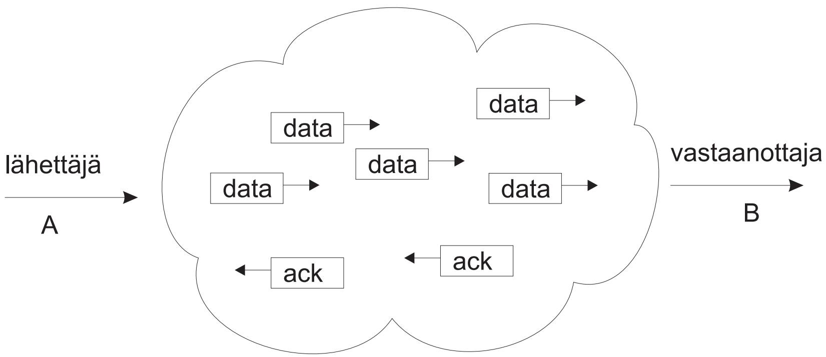
Flow control: window mechanism

- Number of unacknowledged segments (data units) $\leq W$ (window size)
- Total number of transmission permits $= W$
  - each sent segment takes one permit
  - each received acknowledgment returns one permit



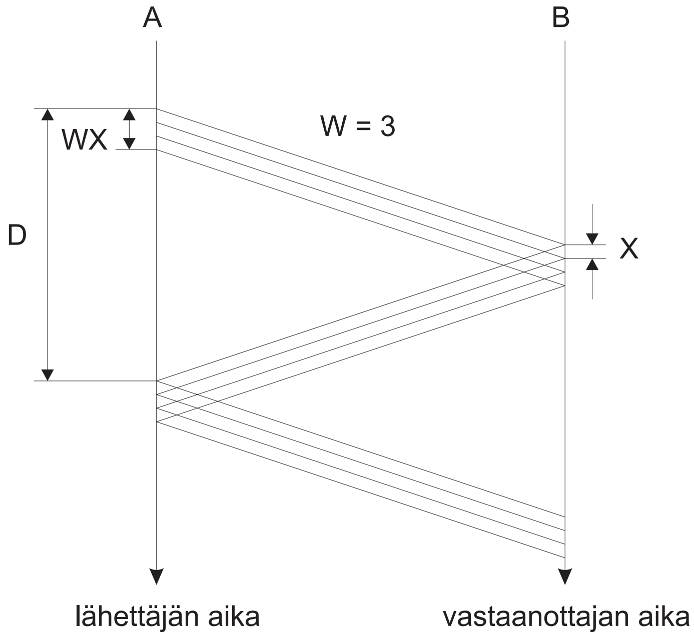
$$W = \begin{cases} \text{total number of transmission permits} \\ + \text{ number of segments on the way} \\ + \text{ number of acks on the way} \end{cases}$$

Window flow control (continued)

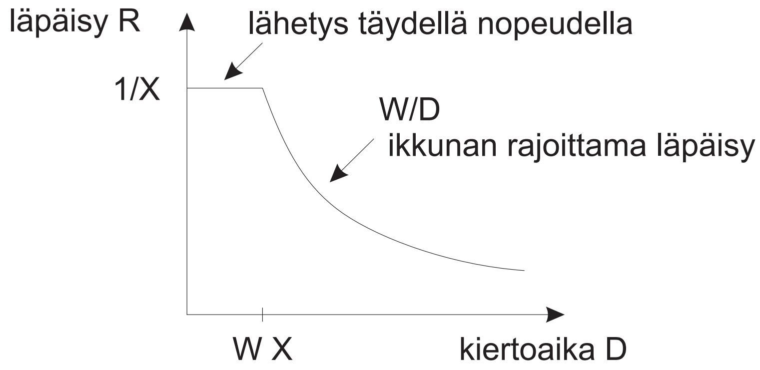
- The size of the window determines the throughput R (segments / s)

  X = transmission time of one segment

  D = round trip time (end-to-end delay of the data + end-to-end delay of the ack)

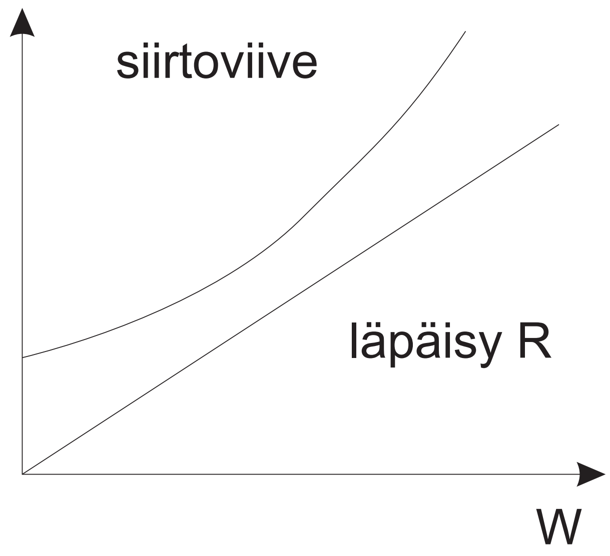


$$R = \min\{\frac{1}{X}, \frac{W}{D}\}$$



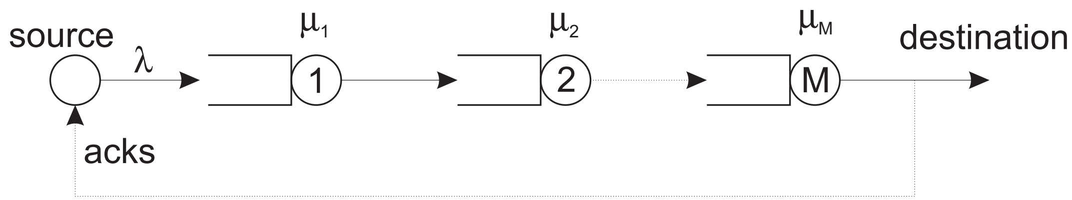
Window flow control (continued)

- With increasing window size W the throughput R increases
- But then also the queueing delays increase
- A good window size is a trade-off between throughput and delay



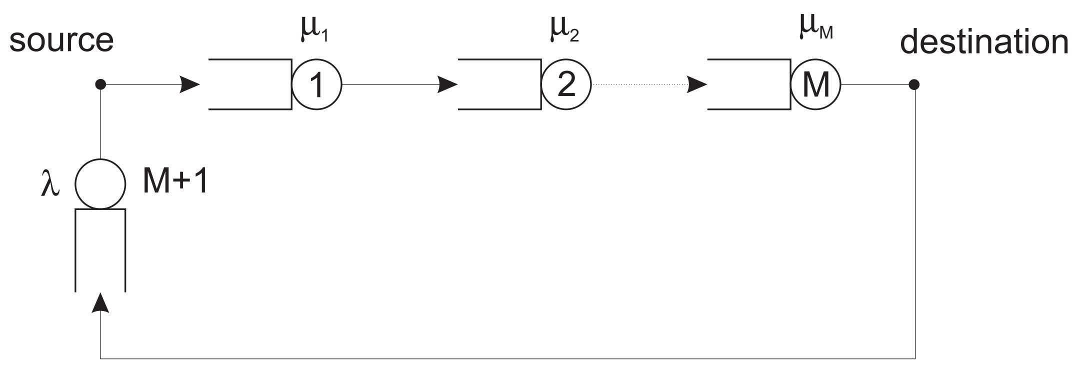
Queueing network analysis of the window flow control

- Suppose that the route of a packet flow is fixed (virtual circuit)
- When the source is permitted to send (sending permits in store), it generates packets at rate λ
- On the route, there are M nodes, with service rates $\mu_1, \ldots \mu_M$
- Propagation delays are neglected (we focus on the queueing delays)
- Acknowledgments are assumed to arrive without any delay
  - it is quite feasible to take into account also the delay of the acks



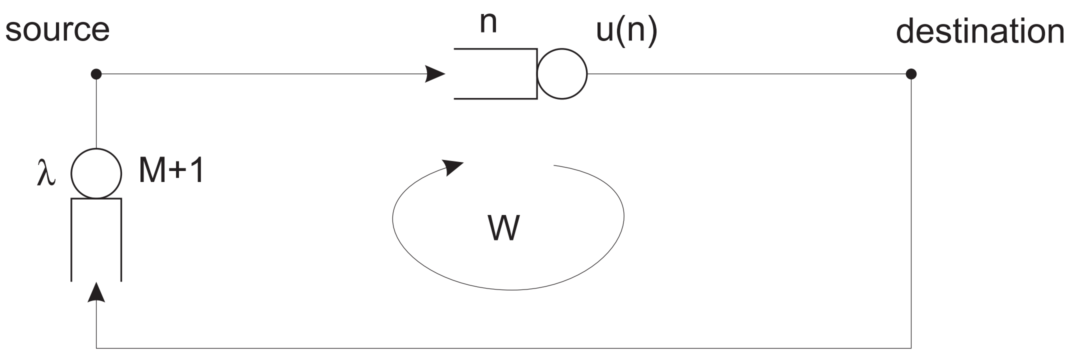
Queueing network model

- The system can be modeled by a closed queueing network, with W ‘packets’ circulating
- In fact, a packet represent a transmission permit
  - in the forward direction, each data packet binds one permit
  - the receiver returns the permit in the form of an acknowledgment
- The extra queue $M+1$ represents the store of transmission permits (collected acks)
  - when there are permits in store, the queue sends packets at rate λ
  - when the queue is empty (all permits have been consumed), there is no output from the queue
  - the output from queue $M+1$ behaves thus precisely as the real source



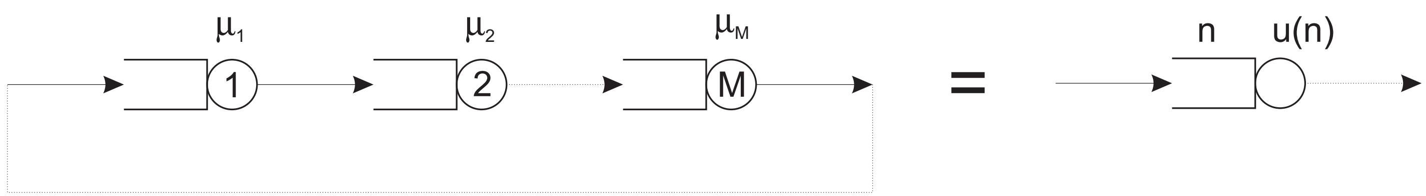
Queueing network model (model)

- By means of the queueing network model, we can find as a function of W
  - end-to-end delay of a packet
  - the throughput of the network (W/ round trip time)
- A closed queueing network can be analyzed with the aid of the mean value analysis (MA)
- The analysis can be facilitated by Norton's theorem
  - the upper branch can be replaced by a single queue with service rate $u(n)$ which depends on the total number of packets circulation



The Equivalent queue

- The service rate $u(n)$ of the equivalent queue equals the throughput in a 'short circuited' network with n packets circulating (can again be found by means of MVA)
- For simplicity assume that the queues $1, \ldots, M$ are identical and $\mu_1 = \cdots = \mu_M = \mu$
- Then the mean delay in one queue is $(1 + (n-1)/M)/\mu$
  - a customer arriving at a queue sees the situation as if he were an outside observer
  - in each queue there are on the average $(n-1)/M$ customers ahead
  - additionally, customer's own service takes on the average the time $1/\mu$
- Thus the round trip time is on the average $(M + n - 1)/\mu$
- The throughput is $u(n) = n\mu/(M + n - 1)$ (n customers circulate in a round trip time)

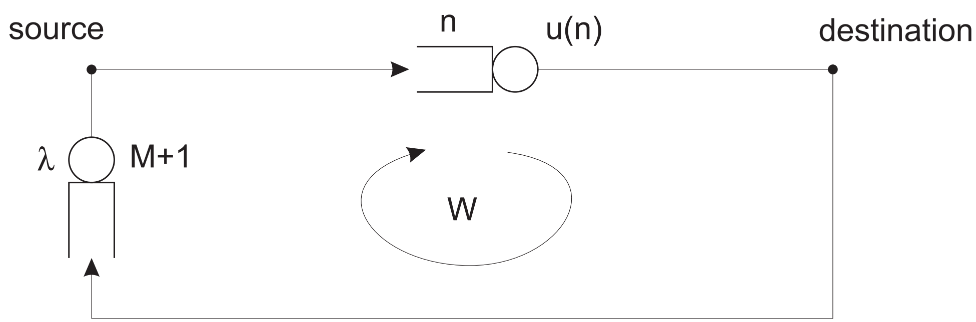


Solution of the queueing network model

- A two queue system with W circulating packets can be solved
- Let the number of packets in the upper queue be n
  - arrival rate to the queue is λ when $n < W$ and 0 when $n = W$ (all the packets in the upper queue)
- The upper queue constitutes a birth-death process

$$p_n = p_0 \frac{\lambda^n}{\Pi_{i=0}^{n} u(i)} = p_0 \frac{\lambda^n}{\mu^n \, \Pi_{i=0}^{n} \frac{i}{i+M-1}} = p_0 \rho^n \frac{(n+M-1)!}{n!(M-1)!}$$

$$\sum_{n=0}^{W} p_n = 1 \quad \Rightarrow \quad p_0 = \left(\sum_{n=0}^{W} \rho^n \frac{(n+M-1)!}{n!(M-1)!} \right)^{-1}$$



Throughput and delay (in the forward direction)

- The throughput γ (packet rate) can be calculated in two different ways:

$$\gamma = \mathrm{E}[u(n)] = \sum_{n=0}^{W} p_n u(n)$$

$$\gamma = (1 - p_W)\lambda$$

- The mean delay T in the forward direction is now obtained by Little's result

$$\mathrm{E}[T] = \frac{\mathrm{E}[n]}{\gamma} = \frac{\sum_{n=0}^{W} p_n n}{\sum_{n=0}^{W} p_n u(n)}$$

Throughput and delay (a special case)

- Assume first that $\lambda = \infty$ (saturated / ‘greedy’ source)
- Then the throughput and delay in the forward direction are as the throughput and round trip time in a ‘short circuited’ network

$$\gamma = u(W) = \frac{W\mu}{W+M-1}, \qquad \mathrm{E}[T] = (W+M-1)\frac{1}{\mu}$$

- The second case $\lambda = \mu$ is a more ‘typical’ one
- With regard to the throughput this differs from the previous one only in that now the number of identical queues is $M+1$; the mean delay in the forward direction is the fraction $M/(M+1)$ of the round trip time

$$\gamma = u(W) = \frac{W\mu}{W+M}, \qquad \mathrm{E}[T] = \frac{M}{M+1}(W+M)\frac{1}{\mu}$$

Choosing the window size

- One wishes great γ but small $\mathrm{E}[T]$
- One has to make a trade-off between these
- Often one takes $\gamma/\mathrm{E}[T]$ as the quantity to be maximized
- In the case $\lambda = \mu$ the maximum is achieved when $W = M$
- In the case $\lambda = \infty$ the maximum is achieved when $W = M - 1$
- As a rule of thumb, the window size should be equal to the number of (bottleneck) nodes
  - then none of the queues is generally empty (whence service capacity would be wasted)
  - on the other hand, there are no long queues and the delay times are reasonable