

Performance models of TCP

□ can simulate

- + faithful to operation of TCP
- expensive, time consuming

□ deterministic approximations

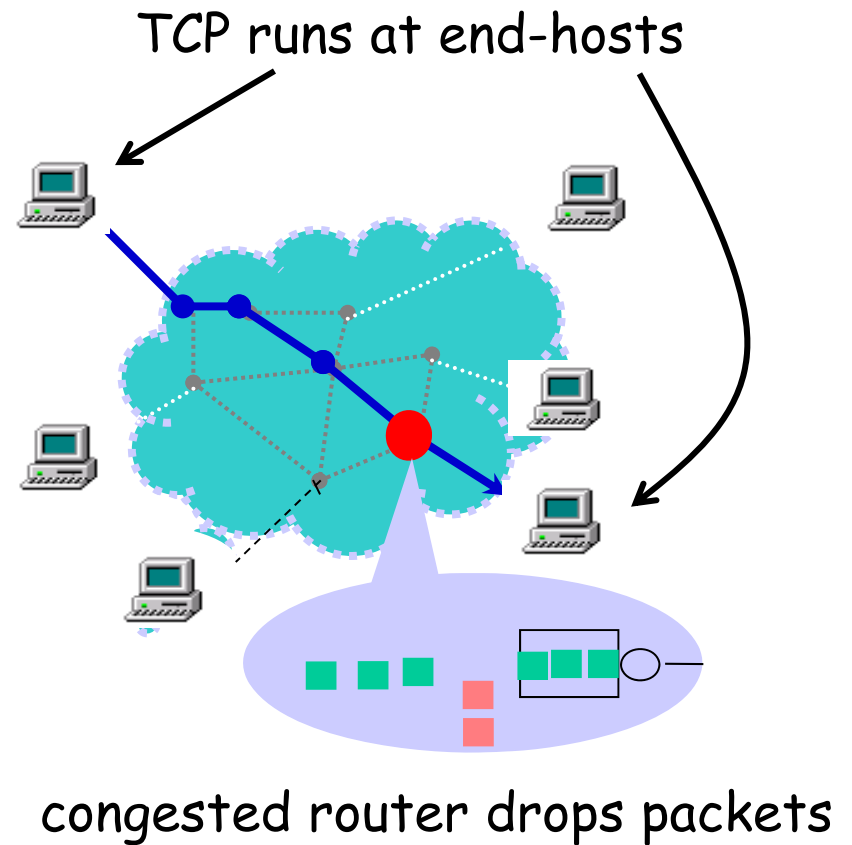
- + quick
- ignore some TCP details, steady state

□ fluid models

- + transient behavior
- ignore some TCP details

TCP behavior

- ❑ congestion control:
 - decrease sending rate when loss detected, increase when no loss
- ❑ routers
 - discard, mark packets when congestion occurs
- ❑ interaction between end systems (TCP) and routers?
 - want to understand (quantify) this interaction

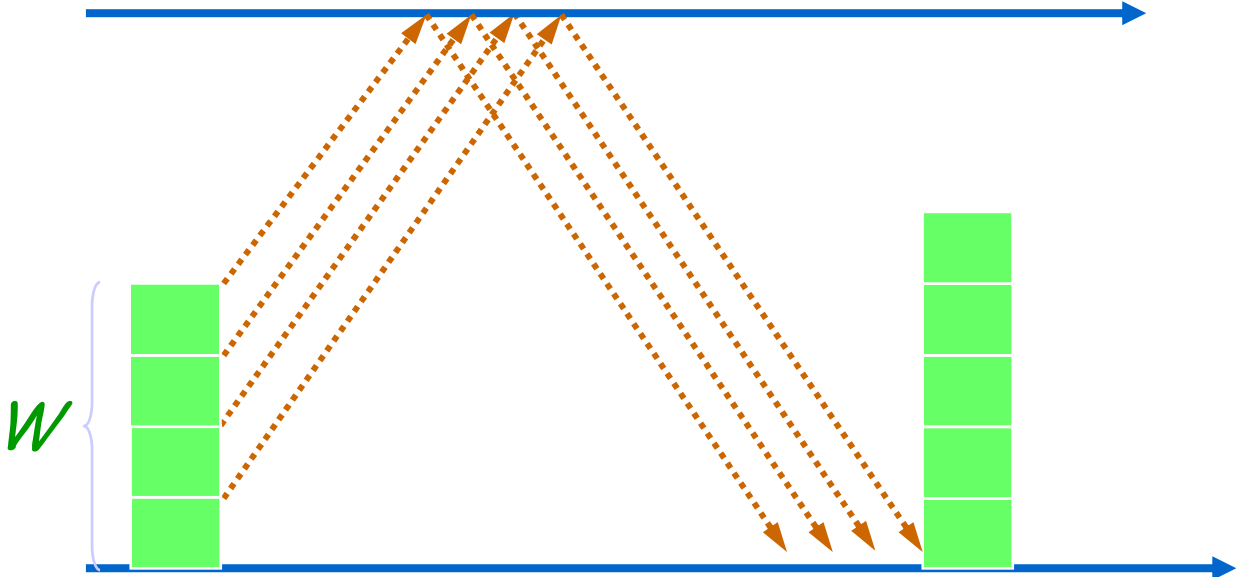


Generic TCP behavior

- window algorithm (window W)
 - up to W packets in network
 - return of ACK allows sender to send another packet
 - cumulative ACKS
- increase window by one per RTT
 - $W \leftarrow W + 1 / W$ per ACK
 - $\Rightarrow W \leftarrow W + 1$ per RTT
- seeks available network bandwidth

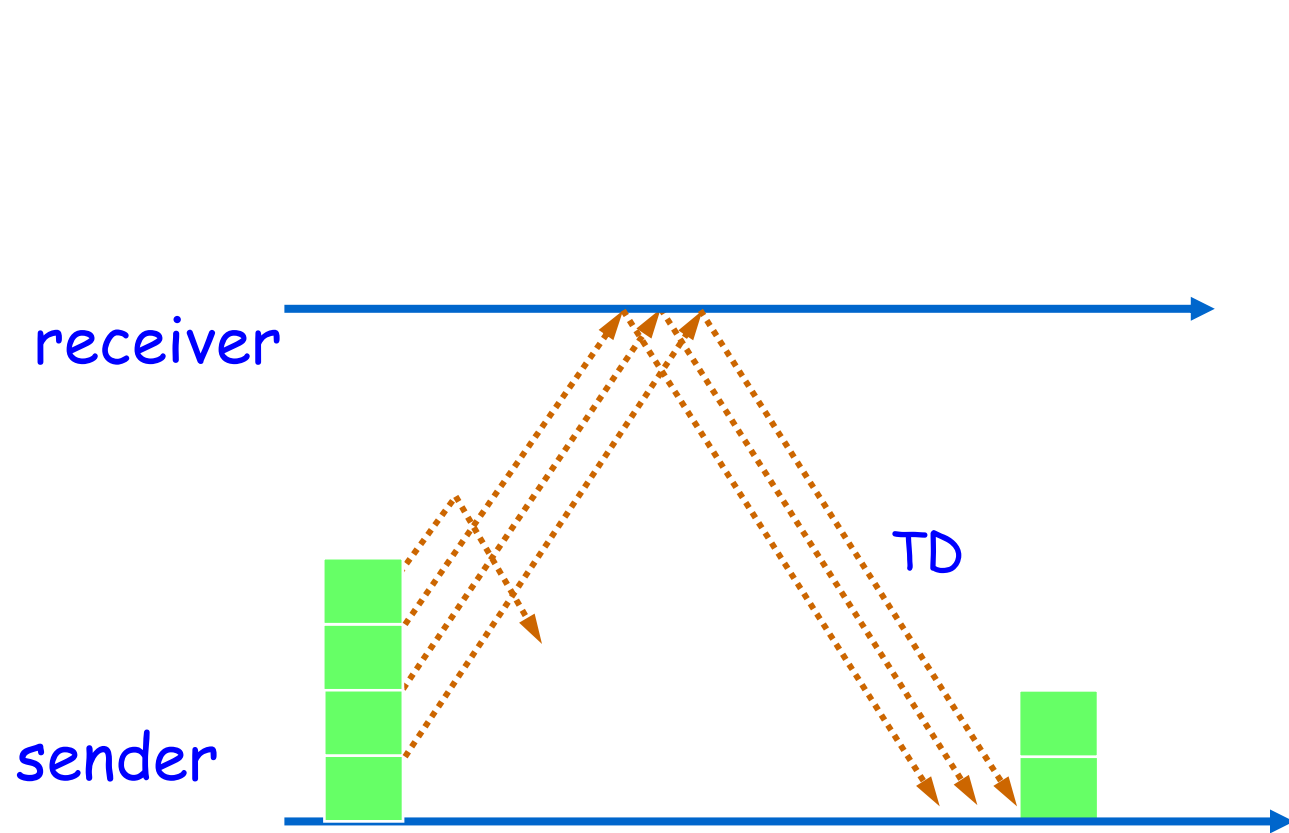
receiver

sender



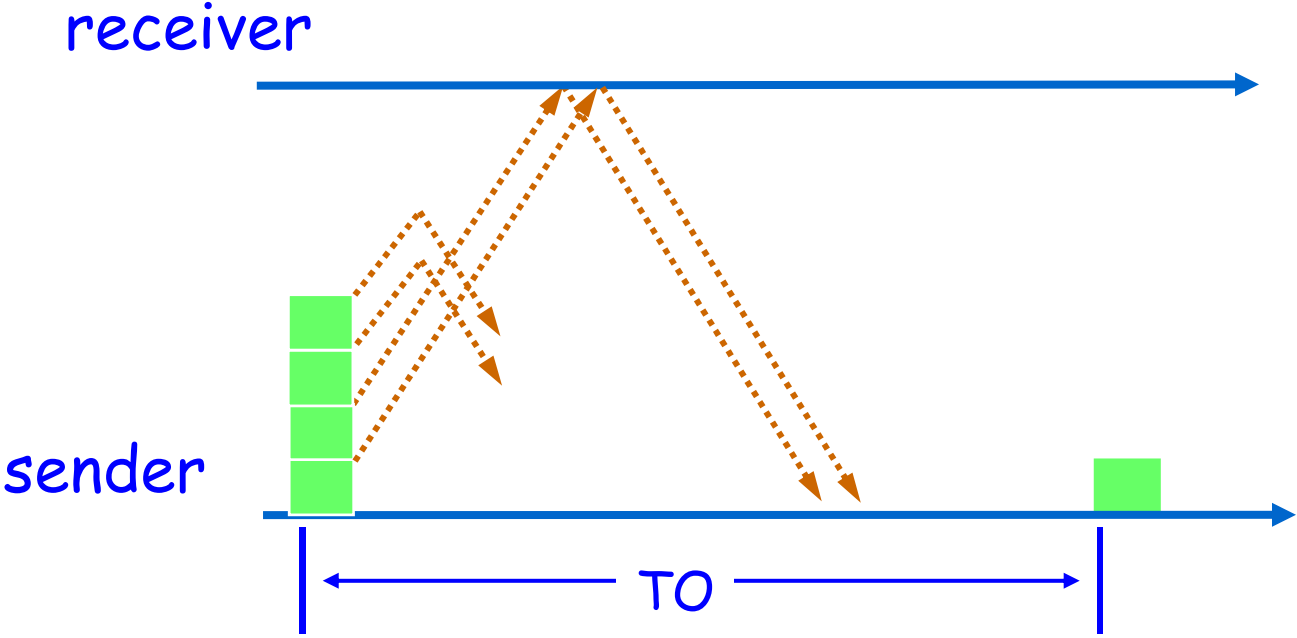
Generic TCP behavior

- window algorithm (window W)
- increase window by one per RTT
 $W \leftarrow W + 1 / W$ per ACK
- loss indication of congestion
- decrease window by half on detection of loss,
(triple duplicate ACKs), $W \leftarrow W / 2$



Generic TCP Behavior

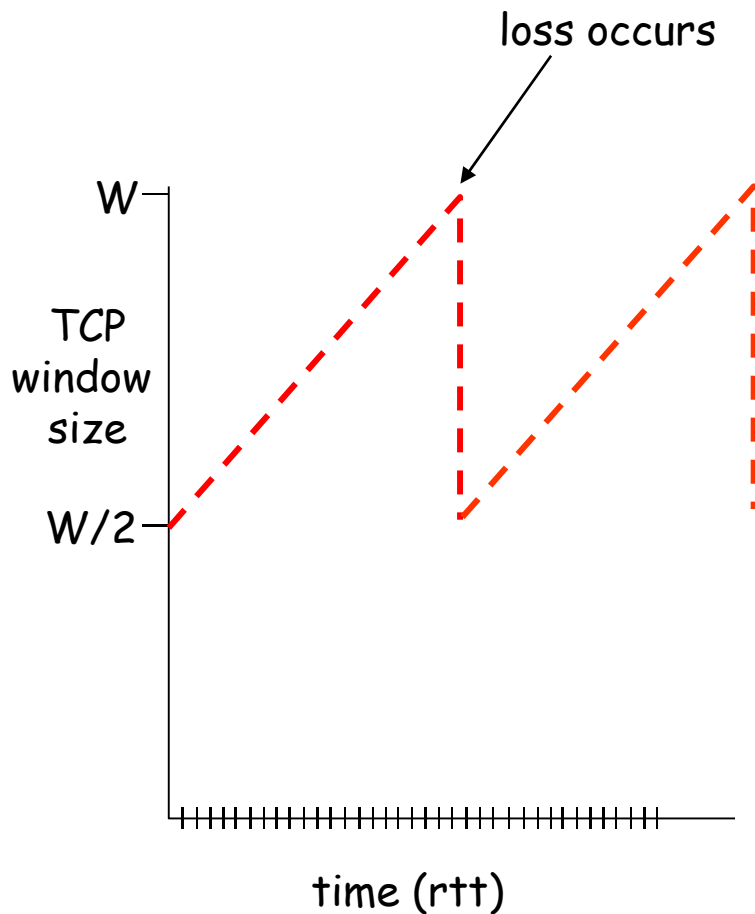
- window algorithm (window W)
- increase window by one per RTT
 $W \leftarrow W + 1 / W$ per ACK
- halve window on detection of loss, $W \leftarrow W/2$
- timeouts due to lack of ACKs \rightarrow window reduced to one, $W \leftarrow 1$



Generic TCP Behavior

- window algorithm (window W)
- increase window by one per RTT (or one over window per ACK, $W \leftarrow W+1/W$)
- halve window on detection of loss, $W \leftarrow W/2$
- timeouts due to lack of ACKs, $W \leftarrow 1$
- successive timeout intervals grow exponentially long up to six times

TCP throughput/loss relationship

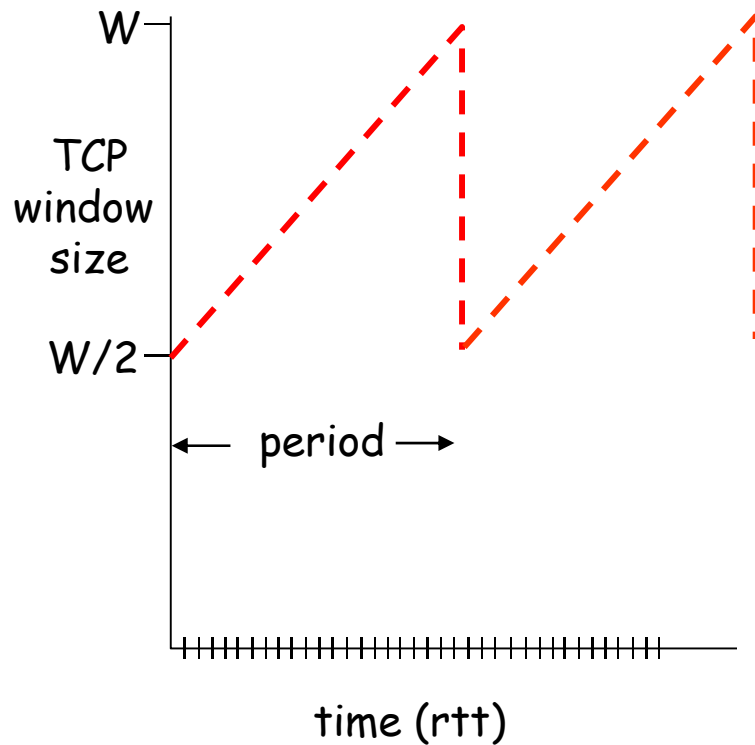


Idealized model:

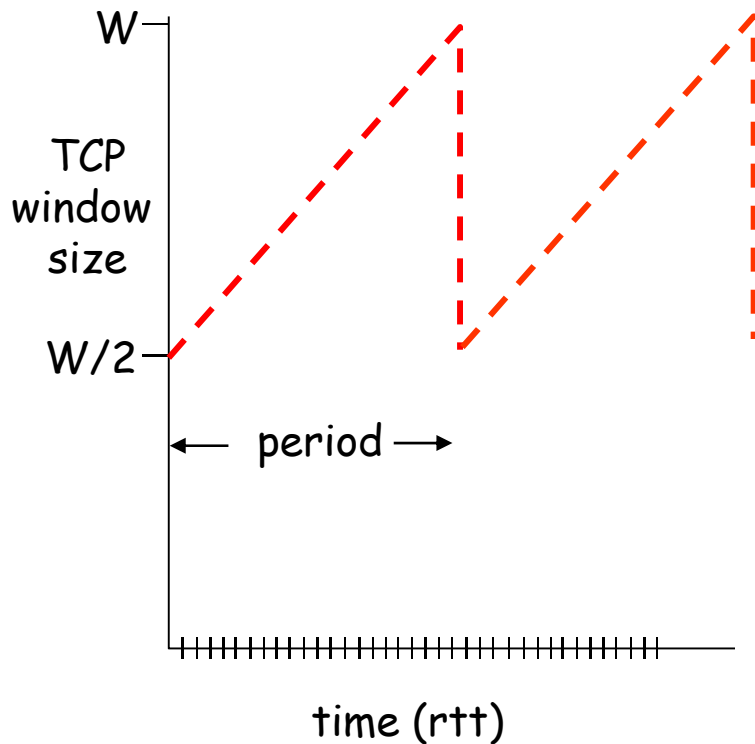
- W is maximum supportable window size (then loss occurs)
- TCP window starts at $W/2$ grows to W , then halves, then grows to W , then halves...
- one window worth of packets each RTT
- *find*: throughput as function of loss, RTT

TCP throughput/loss relationship

packets sent per "period" =



TCP throughput/loss relationship

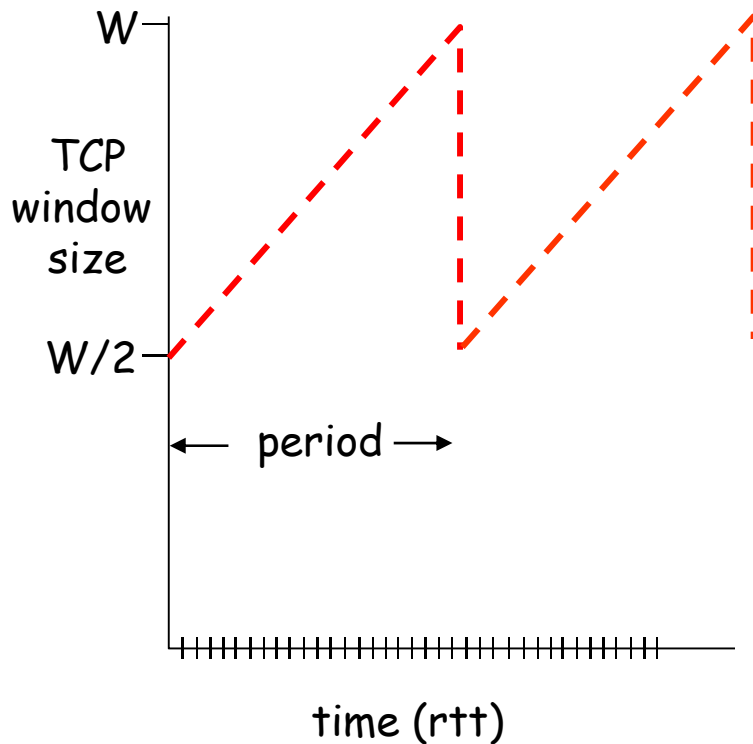


packets sent per "period" =

$$\begin{aligned}
 \frac{W}{2} + \left(\frac{W}{2} + 1\right) + \dots + W &= \sum_{n=0}^{W/2} \left(\frac{W}{2} + n\right) \\
 &= \left(\frac{W}{2} + 1\right) \frac{W}{2} + \sum_{n=0}^{W/2} n \\
 &= \left(\frac{W}{2} + 1\right) \frac{W}{2} + \frac{W/2(W/2 + 1)}{2} \\
 &= \frac{3}{8}W^2 + \frac{3}{4}W \\
 &\approx \frac{3}{8}W^2
 \end{aligned}$$

TCP throughput/loss relationship

$$\# \text{ packets sent per "period"} \approx \frac{3}{8} W^2$$



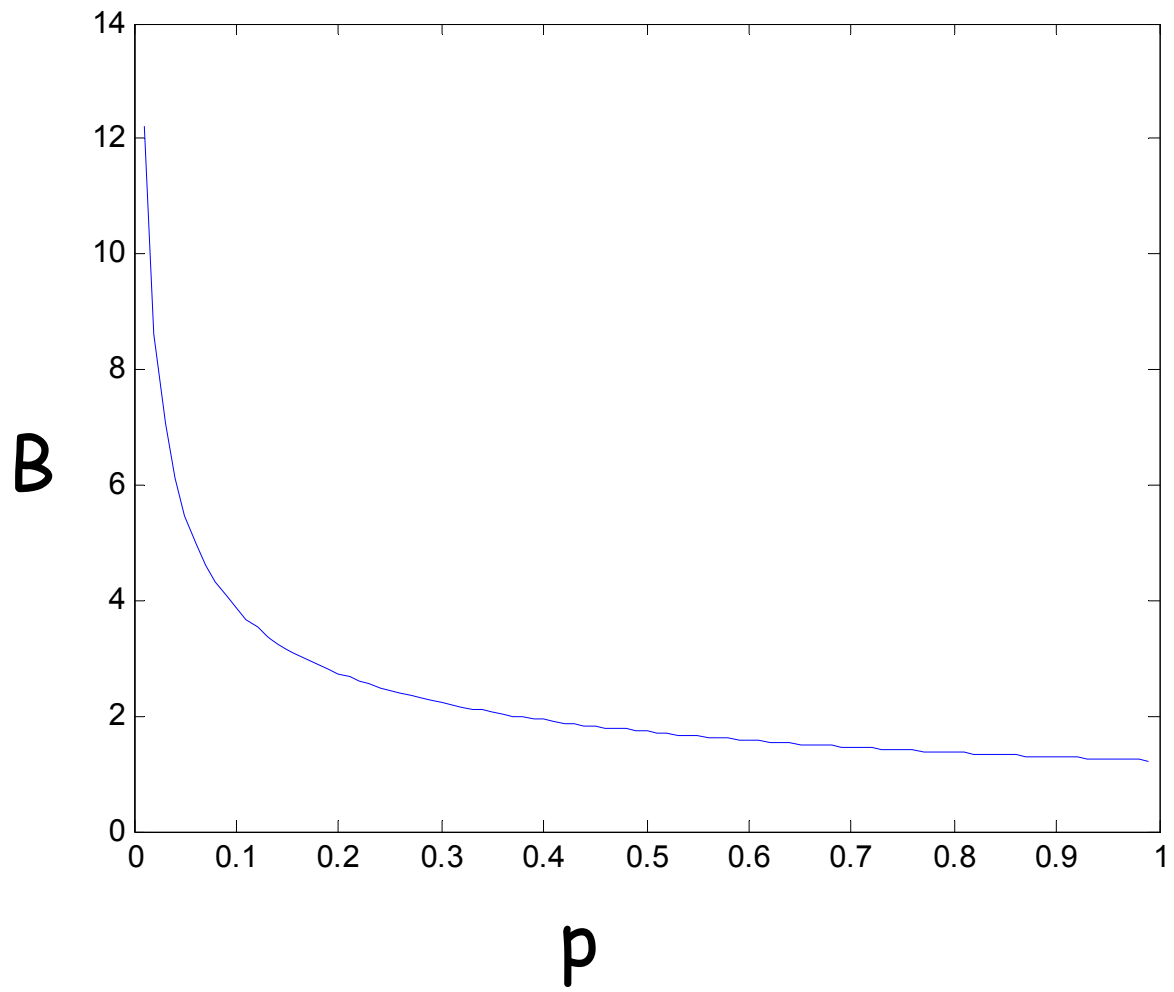
1 packet lost per "period" implies:

$$p_{\text{loss}} \approx \frac{8}{3W^2} \quad \text{or:} \quad W = \sqrt{\frac{8}{3p_{\text{loss}}}}$$

$$B = \text{avg.}_\text{thruput} = \frac{3}{4} W \frac{\text{packets}}{\text{rtt}}$$

$$B = \text{avg.}_\text{thruput} = \frac{1.22}{\sqrt{p_{\text{loss}}}} \frac{\text{packets}}{\text{rtt}}$$

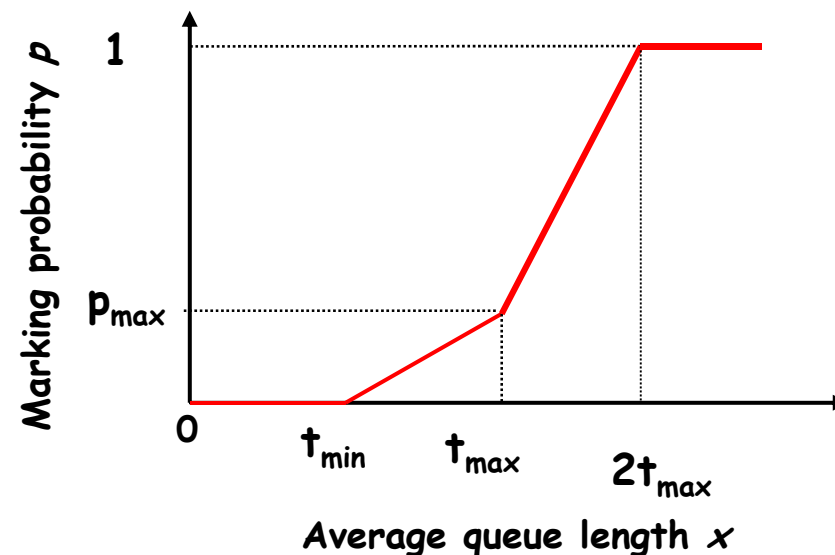
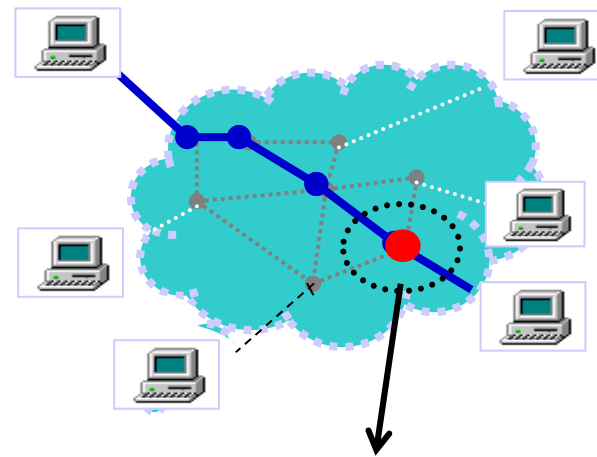
B throughput formula can be extended to model timeouts and slow start (PFTK)



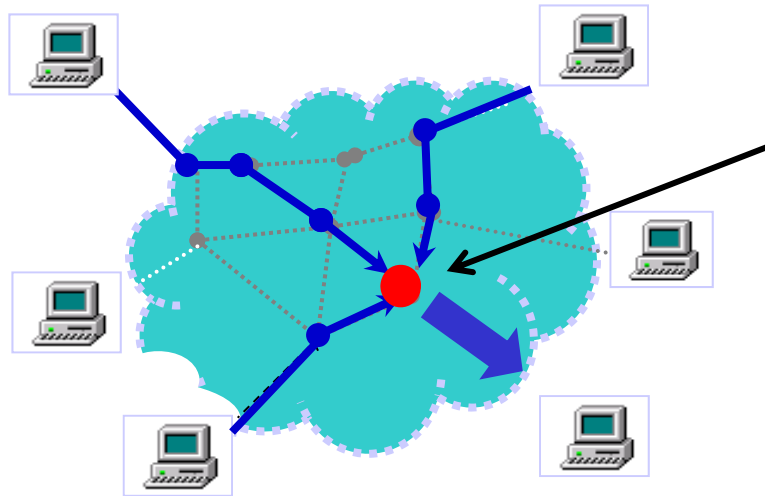
Recall RED queue management

□ dropping/marking packets depends on average queue length $\rightarrow p = p(x)$

□ more generally: active queue management (AQM)



Bottleneck behavior



bottleneck router:

- capacity fully utilized
- all interfering sessions see same loss prob.
- do all sessions see same thruput?

$$\sum_i B_i(p, RTT_i) = C$$

C - router bandwidth

B_i - throughput of flow i

Single bottleneck: infinite flows

- N infinite TCP sessions

- two way propagation delay $A_i, i = 1, \dots, N$
- throughput $B_i(p, RTT_i)$

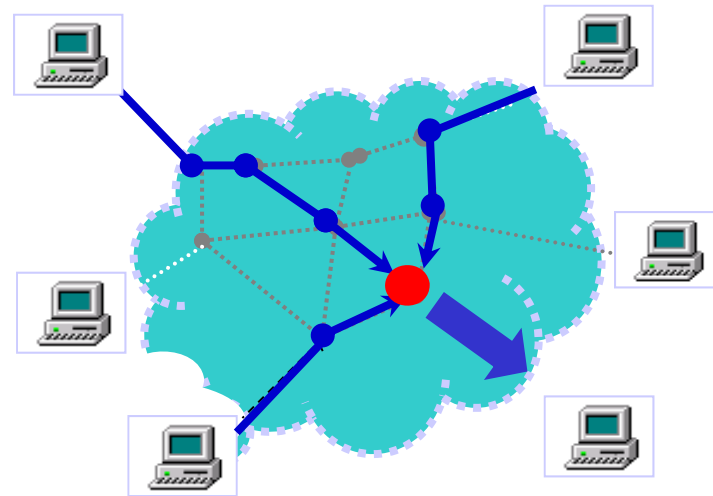
- one bottleneck router

- RED queue management

- avg. queue length x ; dropping probability $p(x)$

- to obtain

- B_i : TCP sessions' throughput,
- router behavior, e.g., drop prob. avg. queue len.



Model and solution

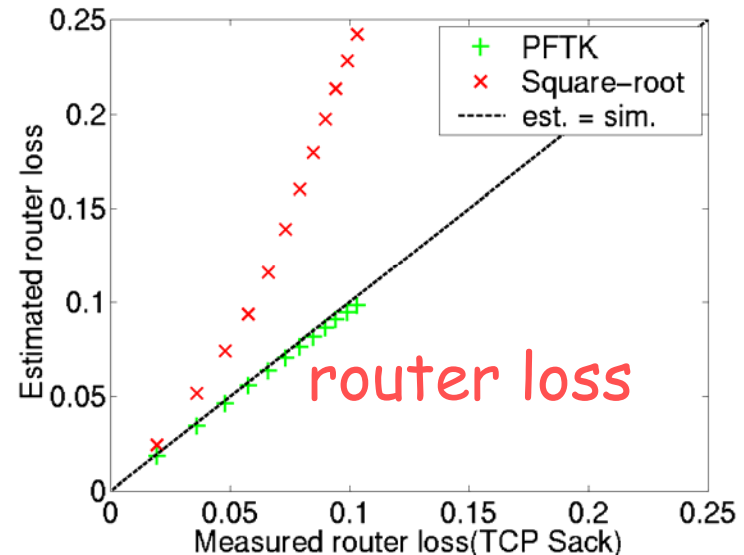
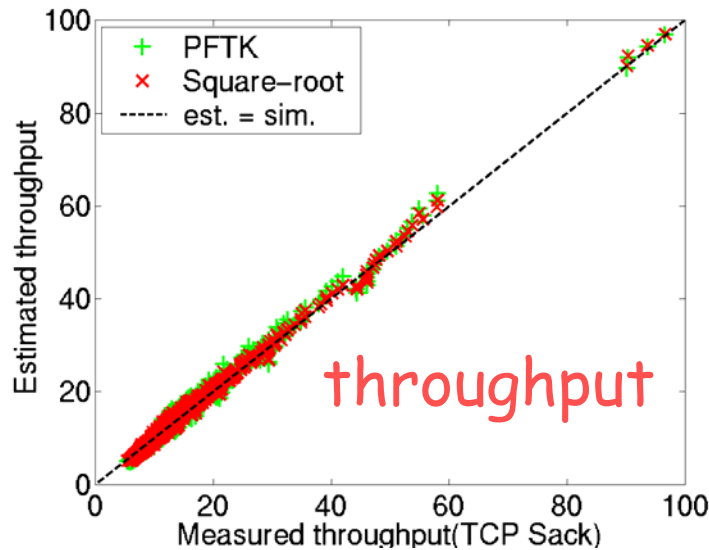
□ model

$$\sum_i B_i(x) = C, \text{ for } j=1, \dots, N$$

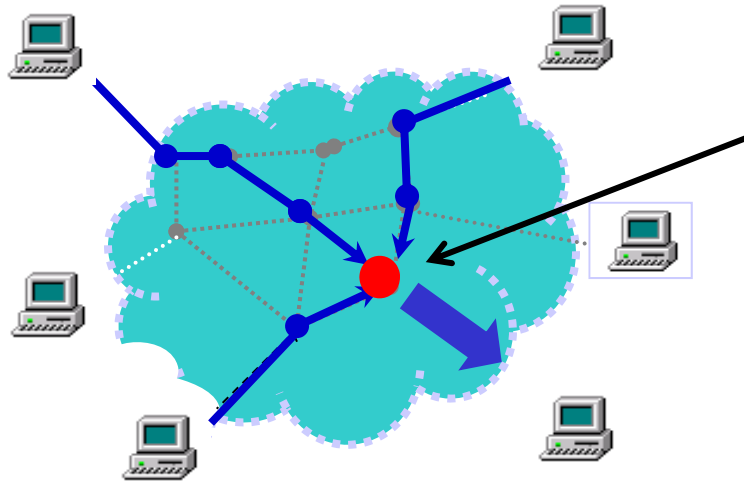
- solve a fixed point problem for x
- unique solution provided B is monotonic and continuous on x
- resulting x can be used to obtain RTT_i and p

Model versus simulation: single bottleneck, infinite flows

- fixed router capacity 4 Mbps and RED parameters
- 10-120 TCP flows
- two-way prop. delay $20+2i$ ms, $i=1,\dots,N$



Bottleneck behavior



bottleneck router:

- capacity fully utilized
- all interfering sessions see same loss prob.

$$\sum_i B_i(RTT_i, p) = C$$

C - router bandwidth

B_i - throughput of flow i

Aside: other applications

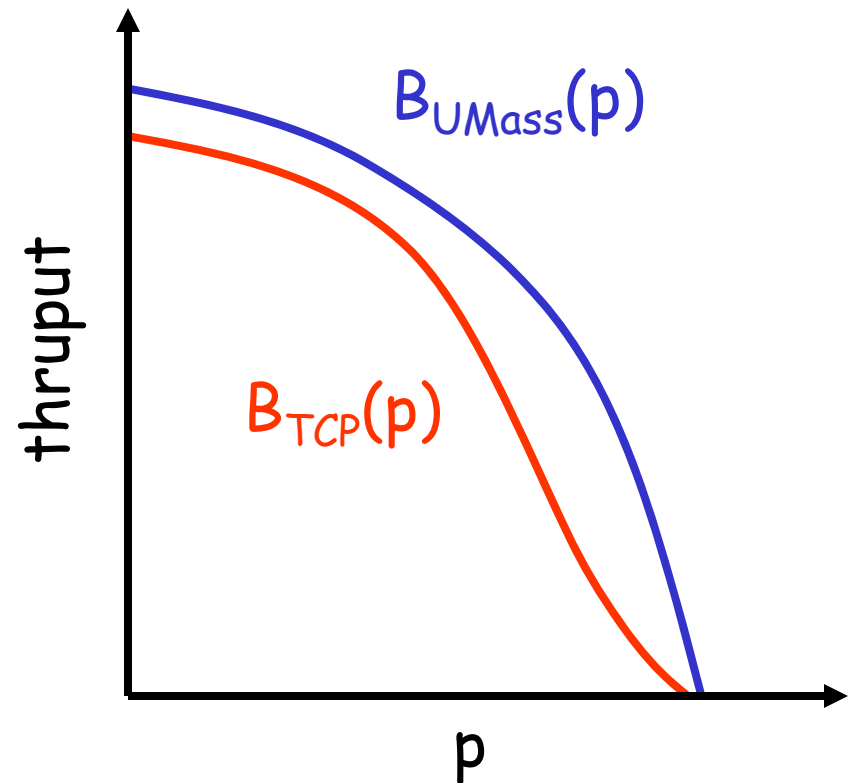
- ❑ comparing different congestion control algorithms
- ❑ is forward error correction useful?

New and improved TCP

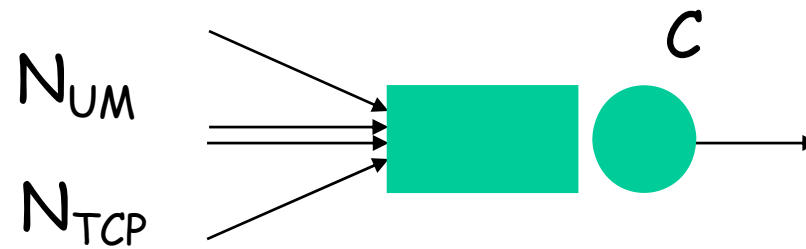
□ new/improved,
 $B_{UMass}(p)$

• TCP, $B_{TCP}(p)$

$$B_{UMass}(p) > B_{TCP}(p)$$



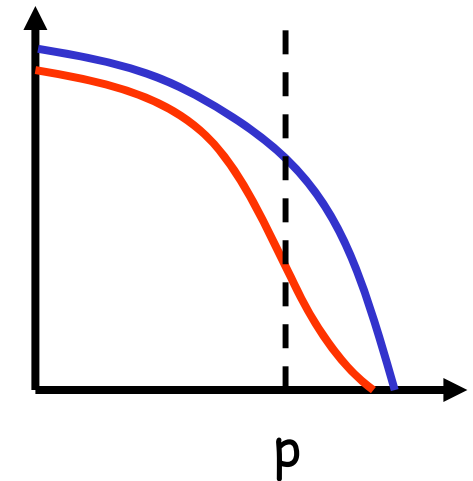
Sharing bottleneck with TCP



$$N_{UM} B_{UM}(p) + N_{TCP} B_{ni}(p) = C$$

$$\Rightarrow B_{UM}(p) > B_{TCP}(p)$$

- a win! friendly?



Replacing TCP with TCP UMass

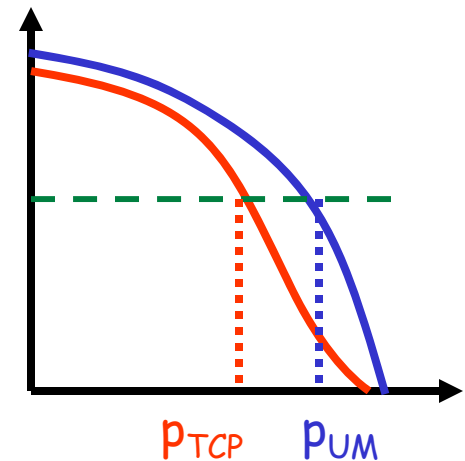
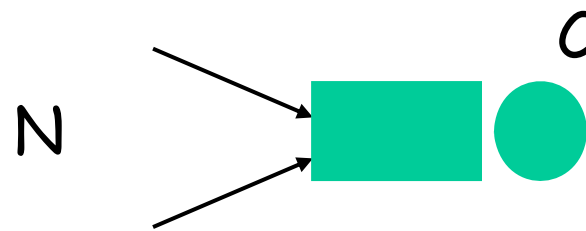
$$N B_{UM}(p_{UM}) = C$$

vs

$$N B_{TCP}(p_{TCP}) = C$$

$$\Rightarrow p_{UM} > p_{TCP}$$

- a loss!

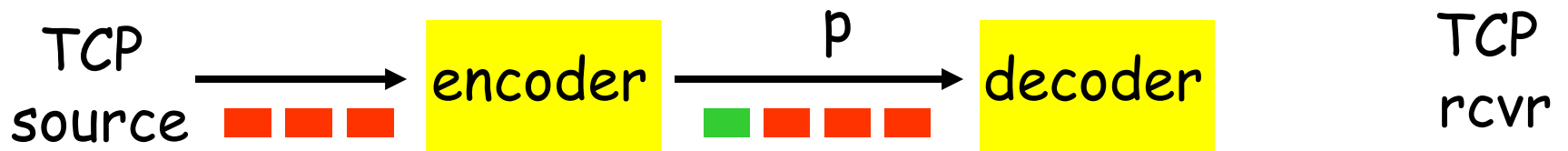
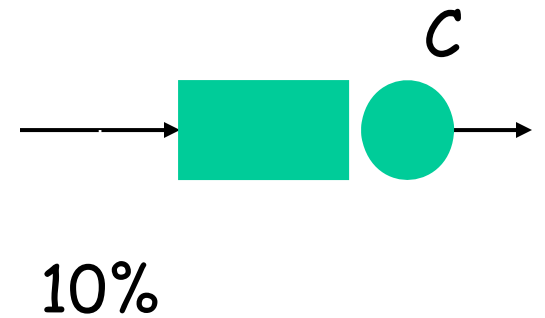


Use of FEC

❑ 10% pkt loss \Rightarrow poor thruput



use packet level FEC!

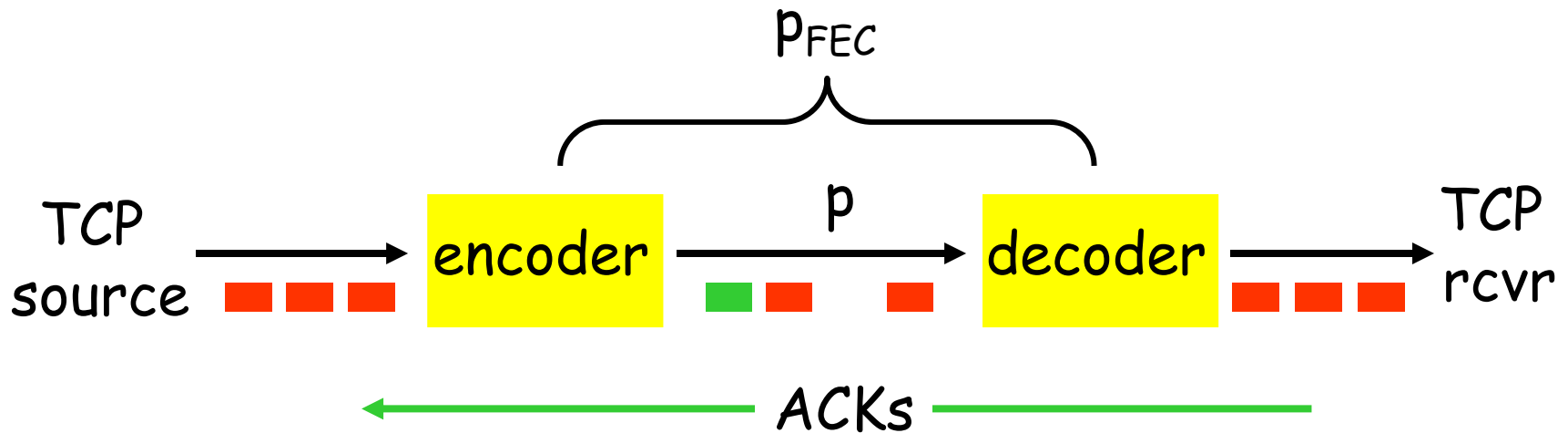
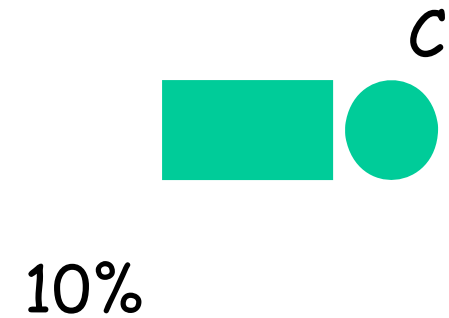


Use of FEC

10% pkt loss \Rightarrow poor thruput



use packet level FEC!



$p_{FEC} \ll p$

Use of FEC

□ available bandwidth, $C_{\text{FEC}} < C_{\text{wo}}$

□ $B_{\text{FEC}}() = B_{\text{wo}}() = B_{\text{TCP}}()$

□ $N B_{\text{wo}}(p_{\text{wo}}) = C_{\text{wo}}$

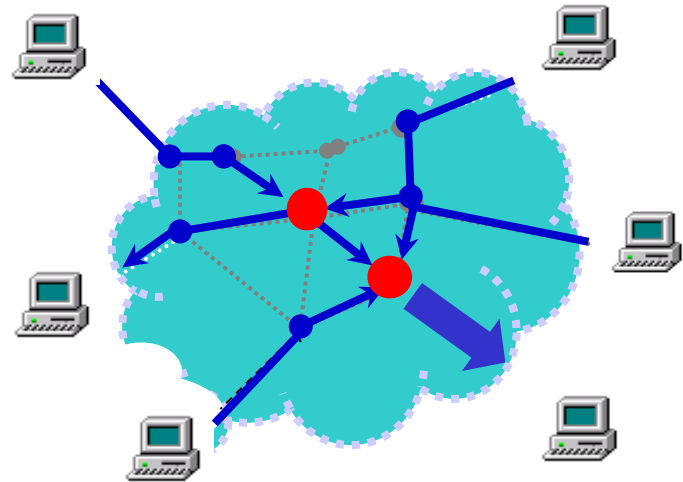
□ $N B_{\text{FEC}}(p_{\text{FEC}}) = C_{\text{FEC}}$

$$\Rightarrow p_{\text{FEC}} > p_{\text{wo}}$$

□ **FEC reduces performance!**

Multiple Bottleneck: infinite flows

- N TCP flows
 - throughputs $B = \{B_i(R_i, p_i)\}$
- V congested AQM routers
 - capacities $C = \{C_v\}$
 - avg. queue lengths $x = \{x_v\}$
 - discard prob. $p = \{p_v(x_v)\}$



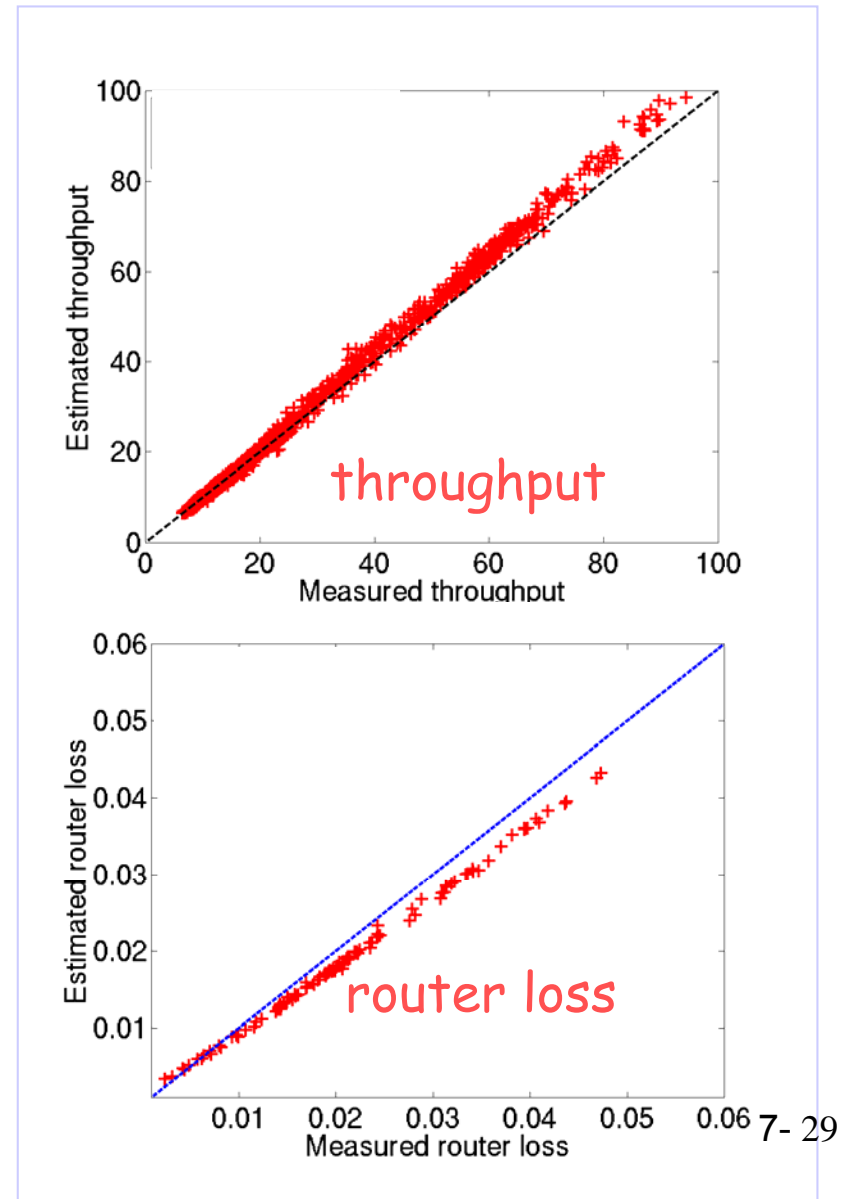
bottleneck router model

$$\sum_i B_i(x_v) = C_v, \quad v=1, \dots, V$$

V equations, V unknowns

Results: multiple bottleneck, infinite flows

- tandem network core, 5 - 10 routers
- 2-way propagation delay 20-120 ms
- bandwidth, 2-6 Mbps
- PFTK model error
 - throughput < 10%
 - loss rate < 10%
 - avg. queue length < 15%
- similar results for cyclic networks



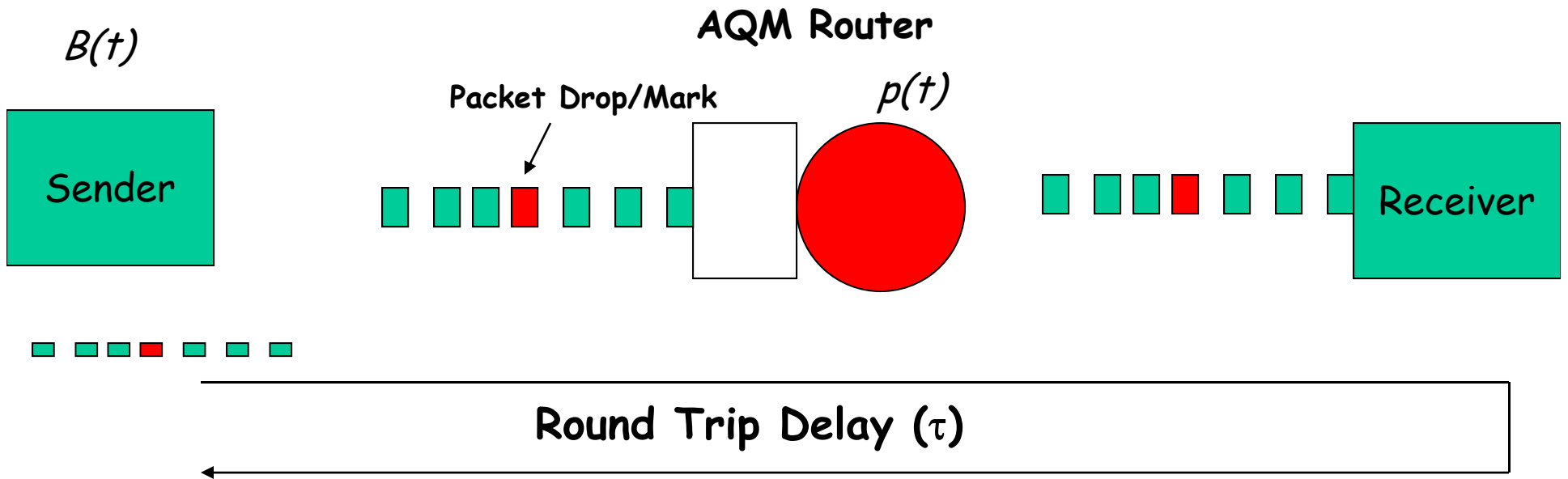
Comments

- ❑ what about UDP / non-TCP flows?
 - if there are "non-responsive" flows, just decrease bottleneck capacity by non-responsive flow rate
- ❑ what about short lived flows?
 - hard (some work in sigcomm 2001 - Massoulie)
- ❑ note: throughout, assumption that time to send packets in window is less than RTT

Dynamic (transient) analysis of TCP fluids

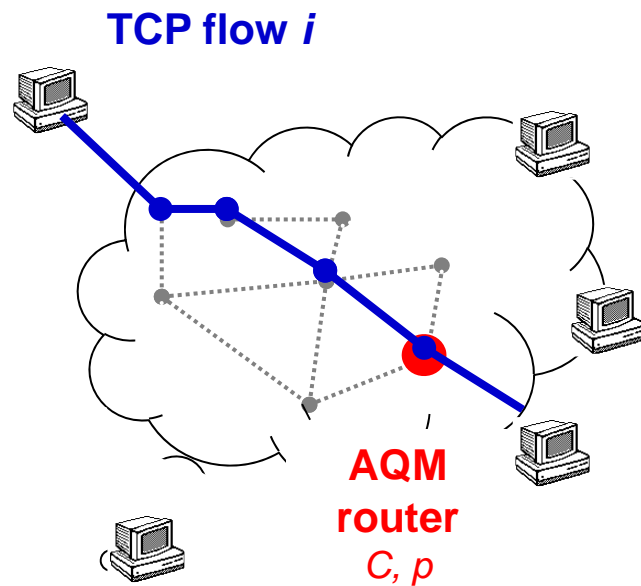
- model TCP traffic as **fluid**
- describe behavior of flows and queues using **Ordinary Differential Equations**
- solve resulting ODEs **numerically**

Loss Model



Loss Rate seen by Sender: $\lambda(t) = B(t-\tau)*p(t-\tau)$

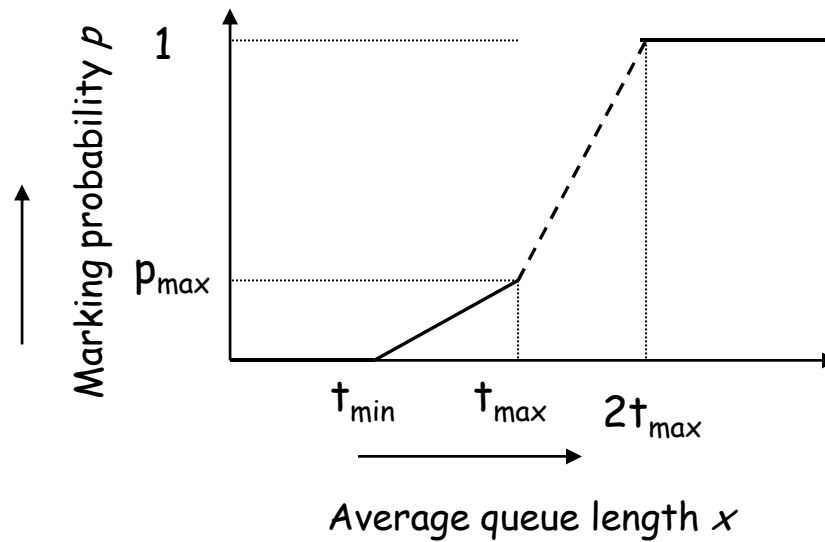
A Single Congested Router



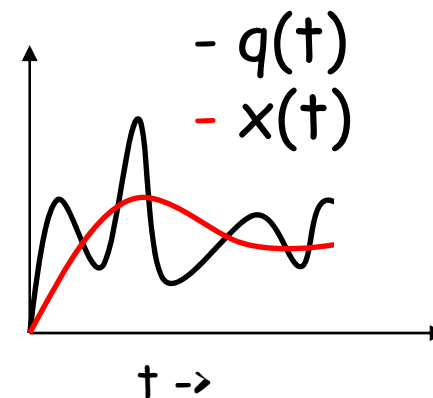
- focus on *single bottlenecked* router
 - capacity $\{C \text{ (packets/sec)}\}$
 - queue length $q(t)$
 - discard prob. $p(t)$
- N TCP flows thru router
 - window sizes $W_i(t)$
 - round trip time
$$R_i(t) = A_i + q(t)/C$$
 - throughputs
$$B_i(t) = W_i(t)/R_i(t)$$

Adding RED to the model

RED: Marking/dropping based on *average* queue length $x(t)$



$x(t)$: smoothed, time averaged $q(t)$



System of Differential Equations

Timeouts and slow start ignored

Window Size:
$$\frac{d\bar{W}_k}{dt} = \frac{\bar{W}_k(t - \tau_k)}{\bar{R}_k(t - \tau_k)} \left(\frac{1}{\bar{W}_k(t)} (1 - p(\bar{x}(t - \tau_k))) - \frac{\bar{W}_k(t)}{2} p(\bar{x}(t - \tau_k)) \right)$$

Throughput
Additive increase
Mult. decrease

Queue length:
$$\frac{d\bar{q}}{dt} = -C 1\{\bar{q} > 0\} + \sum_k \frac{\bar{W}_k(t)}{\bar{R}_k(t)}$$

Outgoing traffic
Incoming traffic

System of Differential Equations

if we ignore feedback delay, $\tau \rightarrow 0$
(not $R(t) = 0$)

$$\frac{d\bar{W}_k}{dt} = \frac{(1 - p(\bar{x}(t)))}{\bar{R}_k(t)} - \frac{\bar{W}_k(t)}{2} \frac{\bar{W}_k(t)}{\bar{R}_k(t)} p(\bar{x}(t))$$

System of Differential Equations (cont.)

Red queue length smoothing

$$x(t_{k+1}) = (1 - \alpha)x(t_k) + \alpha q(t_k)$$

Where

α = averaging parameter of RED(w_{th})

$t_{k+1} - t_k = \delta$ = sampling interval $\sim 1/C$

System of Differential Equations (cont.)

Average smoothed
queue length:

$$\frac{d\bar{x}}{dt} = \frac{\ln(1-\alpha)}{\delta} \bar{x}(t) - \frac{\ln \alpha}{\delta} \bar{q}(t)$$

Where

α = averaging parameter of RED(w_{th})

δ = sampling interval $\sim 1/C$

Loss probability:

$$\frac{dp}{dt} = \frac{dp}{d\bar{x}} \frac{d\bar{x}}{dt}$$

Where $\frac{dp}{d\bar{x}}$ is obtained from the marking profile

N+2 coupled equations

N flows

$$d\bar{W}_i/dt = f_1(p, \bar{R}_i, \bar{W}_i), \quad i = 1, \dots, N$$

$W_i(t)$ = Window size
of flow i

$R_i(t)$ = RTT of flow i

$p(t)$ = Drop probability

$q(t)$ = queue length

$$dp/dt = f_3(\bar{q}) \quad d\bar{q}/dt = f_2(\bar{W}_i)$$

Equations solved numerically using MATLAB

Steady state behavior

□ let $t \rightarrow \infty$

$$\frac{d\bar{W}_k}{dt} \rightarrow 0, \quad p(t) \rightarrow p, \quad \bar{W}(t) \rightarrow \bar{W}, \quad \bar{R}_k(t) \rightarrow \bar{R}_k$$

□ this yields

□ the throughput is

Steady state behavior

□ let $t \rightarrow \infty$

$$\frac{d\bar{W}_k}{dt} \rightarrow 0, \quad p(t) \rightarrow p, \quad \bar{W}(t) \rightarrow \bar{W}, \quad \bar{R}_k(t) \rightarrow \bar{R}_k$$

□ this yields

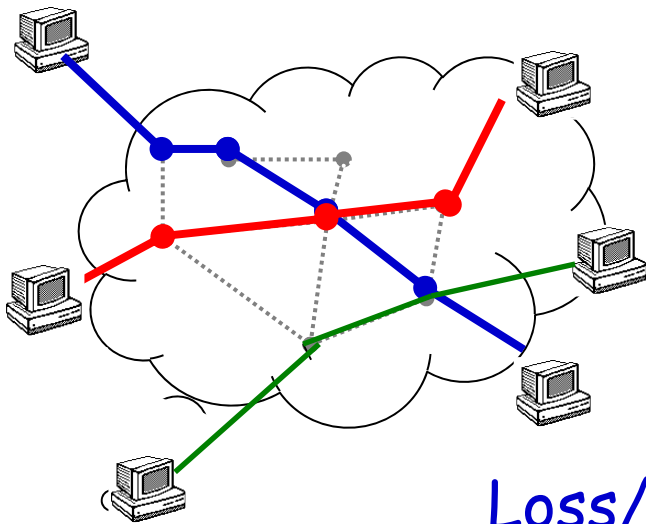
$$0 = \frac{1-p}{\bar{R}_k} - \frac{\bar{W}_k}{2} \frac{\bar{W}_k}{\bar{R}_k} p \quad \text{or} \quad \bar{W}_k = \frac{\sqrt{2(1-p)}}{\sqrt{p}}$$

□ the throughput is

$$B_k = \frac{\sqrt{2(1-p)}}{\bar{R}_k \sqrt{p}} \approx \frac{\sqrt{2}}{\bar{R}_k \sqrt{p}} \quad \text{for small } p$$

A queue is not a Network

Network - set of AQM routers, V
sequence V_i for session i



Round trip time - aggregate delay

$$R_i(t) = A_i + \sum_{v \in V_i} q_v(t) / C_v$$

Loss/marking probability - cumulative prob

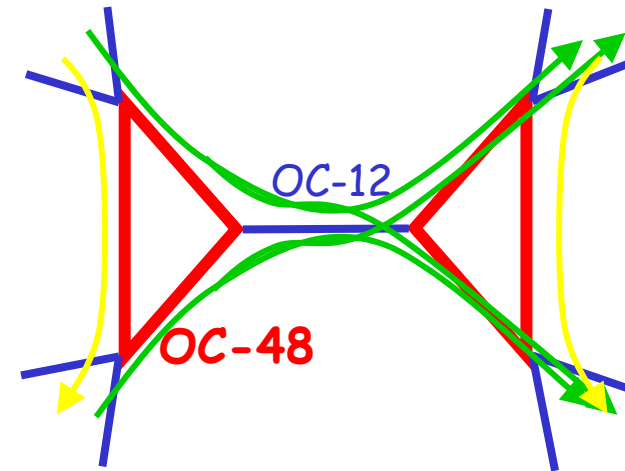
$$p_i(t) = 1 - \prod_{v \in V_i} (1 - p_v(q_v(t)))$$

Link bandwidth constraints

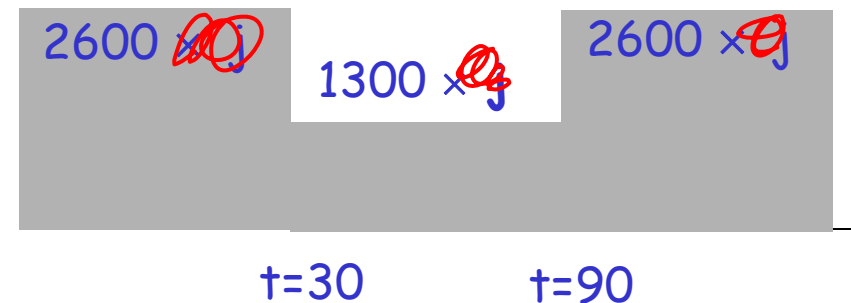
Queue equations

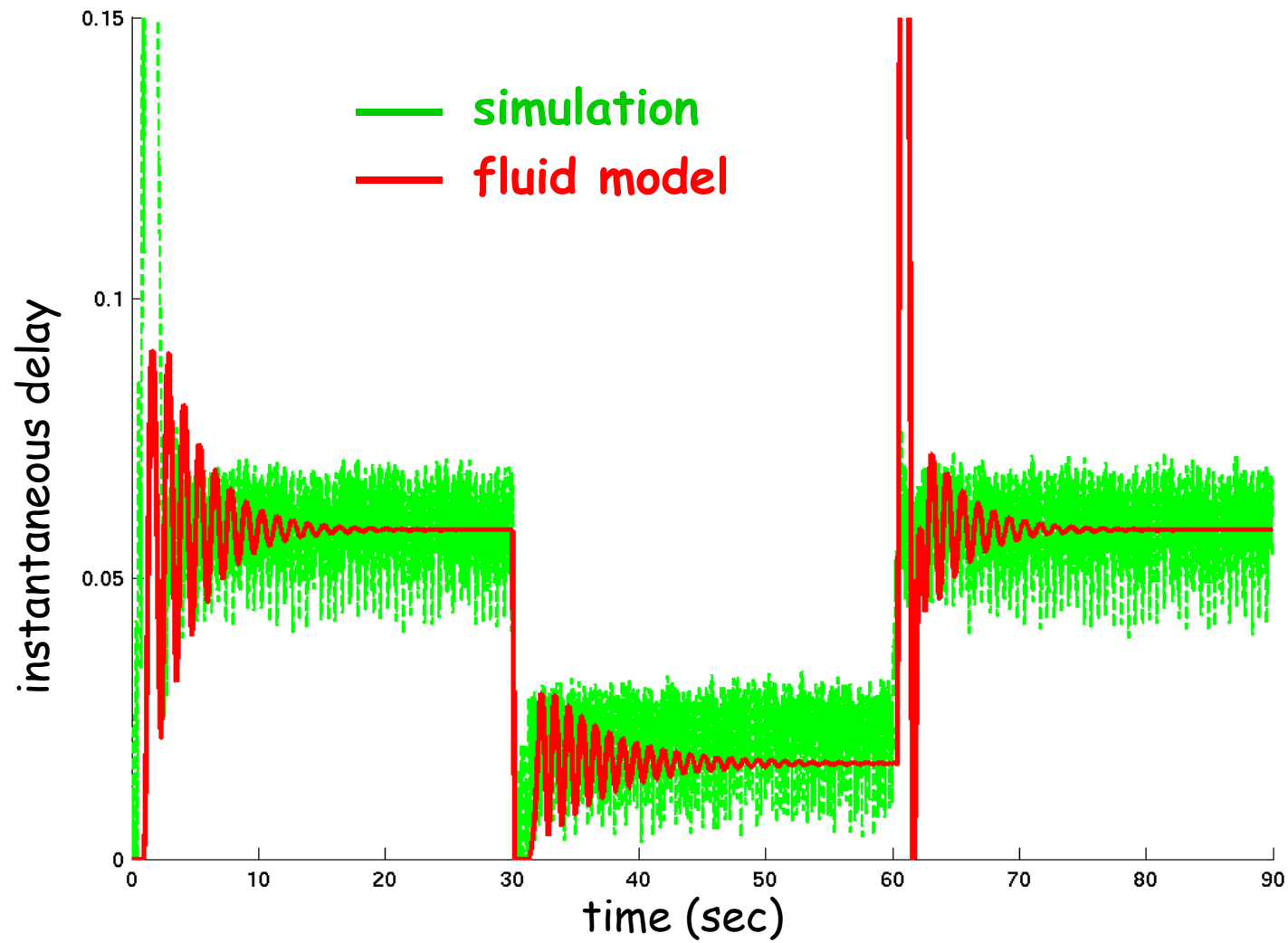
How well does it work?

- ❑ OC-12 - OC-48 links
- ❑ RED with target delay 5msec
- ❑ 2600 TCP flows

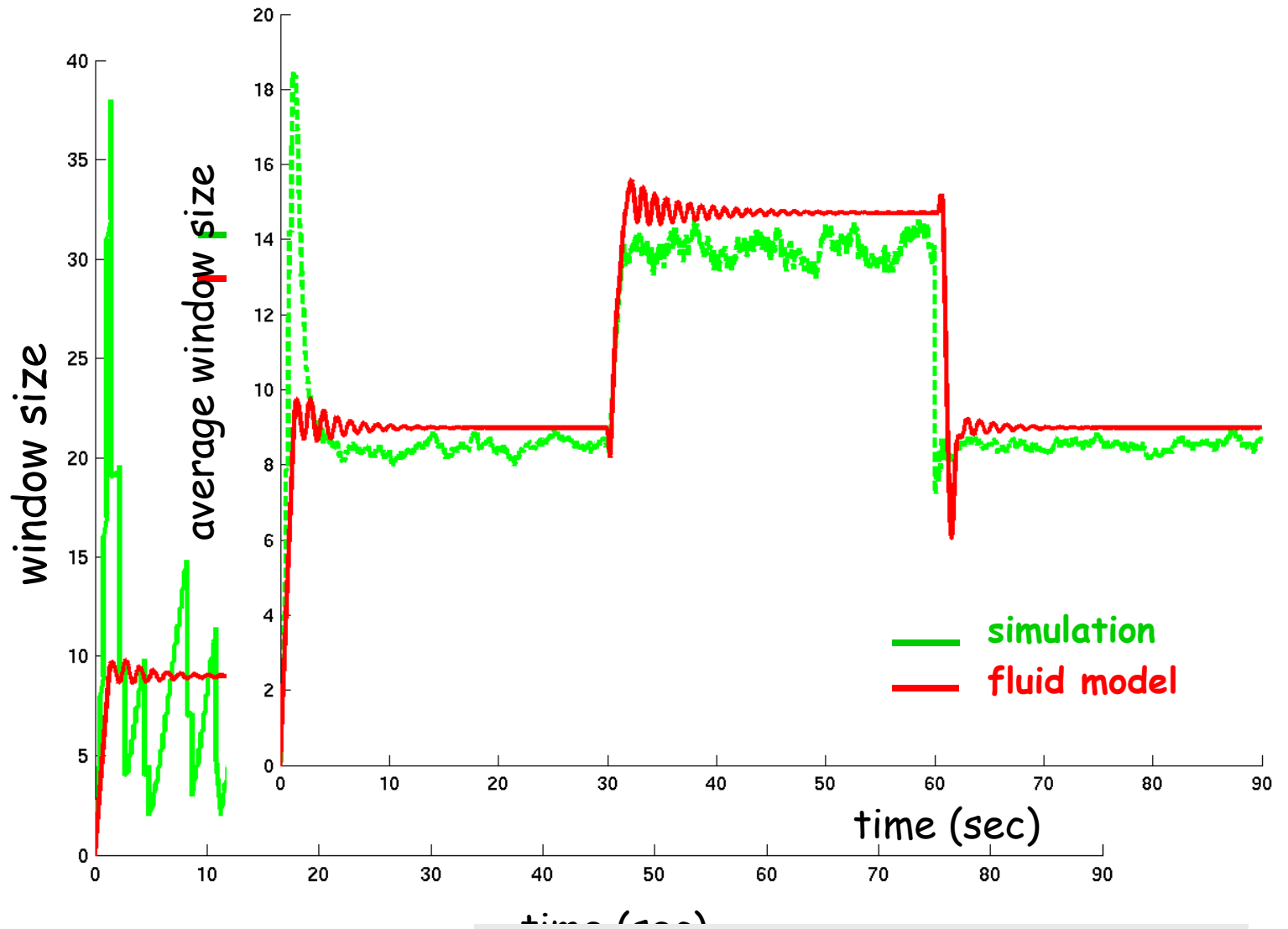


- ❑ decrease to 1300 at 30 sec.
- ❑ increase to 2600 at 90 sec.



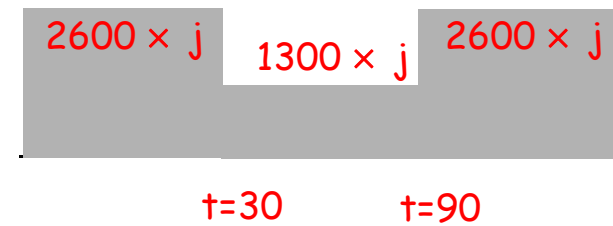
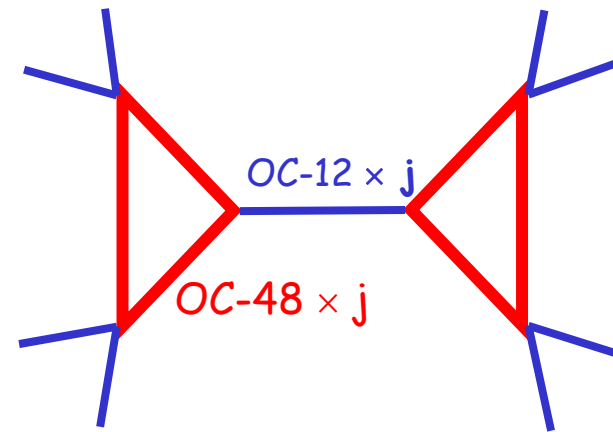
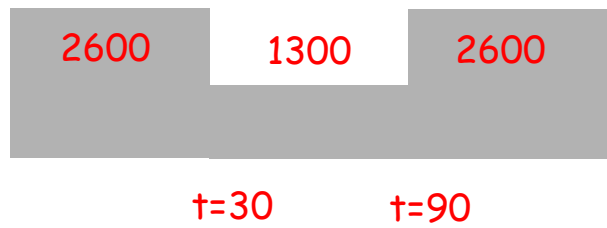
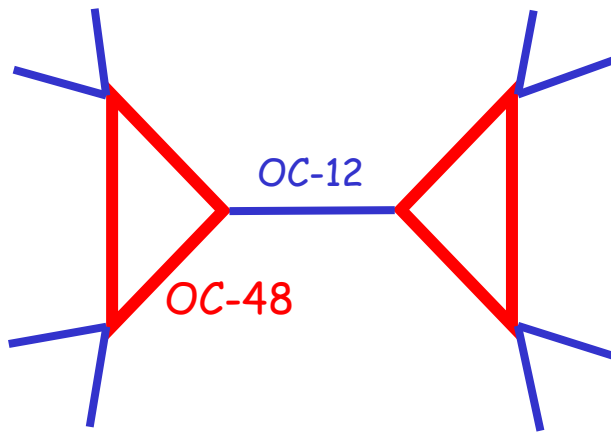


Good queue length match



matches *average* window size

Scaling Properties



$$W_k(t) = W_k^j(t)$$

$$q_v(t) = q_v^j(t) / 100^j$$

Summary: TCP flows as fluids

What have we seen?

- model TCP as constant rate fluid flows
- rate sensitive to congestion via:
 - capacities $C = \langle C_v \rangle$
 - avg. queue lengths $x = \langle x_v \rangle$
 - discard prob. $p = \langle p_v(x_v) \rangle$
- dynamic (transient) behavior of TCP modeled as system of differential equations

ability to predict performance of system of TCP flows using fluid models

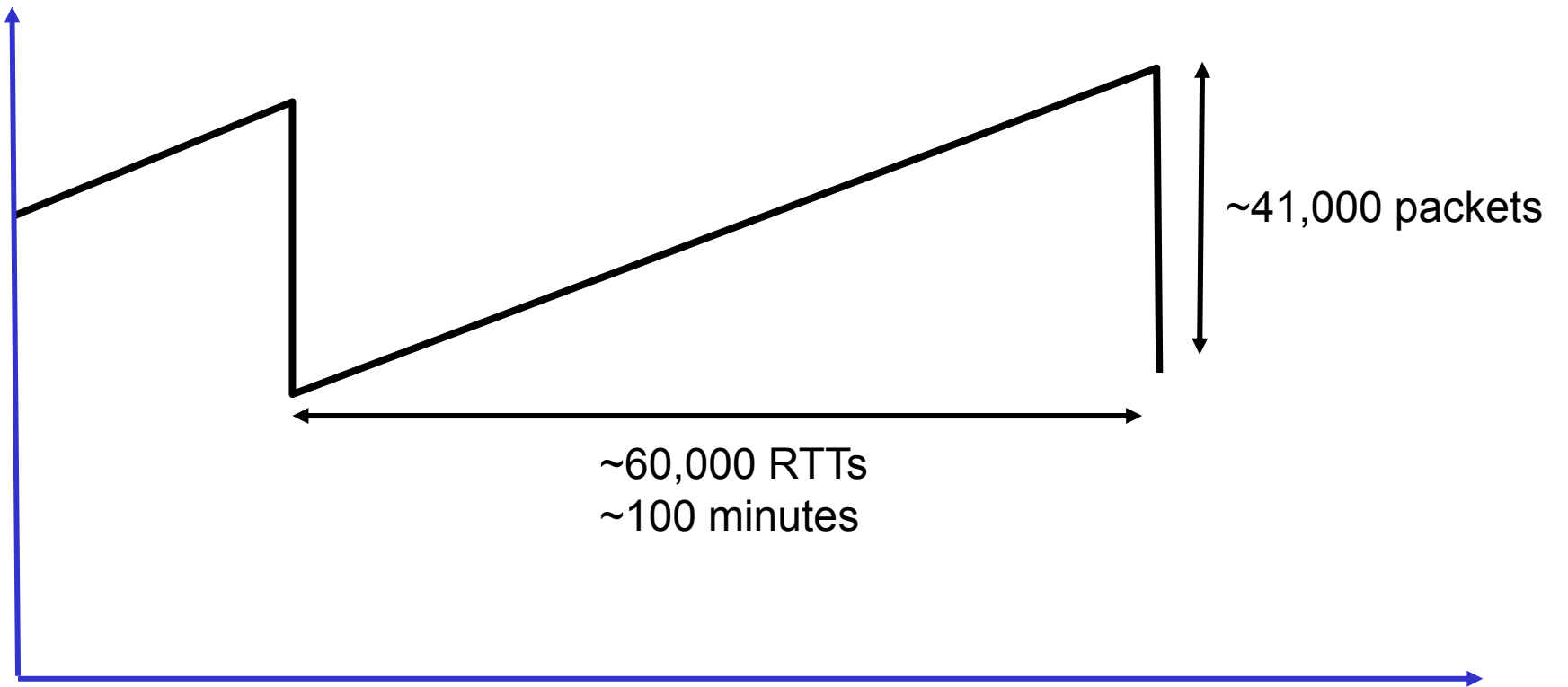
Other applications

- ❑ droptail queue
- ❑ other congestion control algorithms

High performance setting

Standard TCP connection with:

- ❑ 1500-byte packets;
- ❑ 100 ms round-trip time;
- ❑ steady-state throughput of 10 Gbps;
- ❑ requires average congestion window of 83,333 segments;
- ❑ at most one drop (mark) every 5,000,000,000 packets equivalently, one drop every $1 \frac{2}{3}$ hours).



In practice, users do one of:

- open N parallel TCP connections
- use MulTCP (roughly an aggregate of N virtual TCP connections).

Can one do better?

At one end of spectrum:

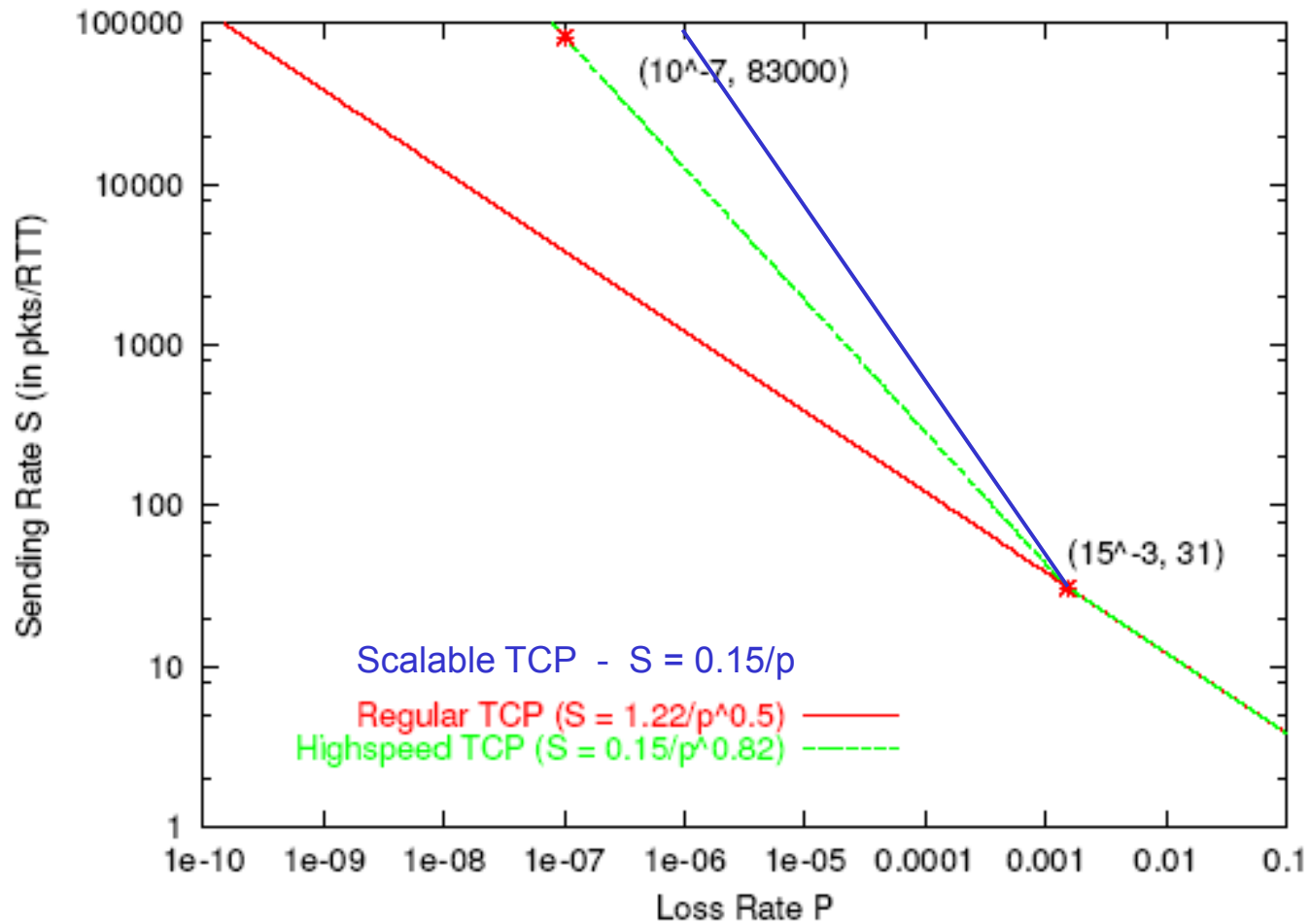
- simple, incremental, and easily-deployable changes to current protocols
 - HighSpeed TCP (TCP with modified parameters);
 - QuickStart (IP option to allow high initial congestion windows.)

At other end of spectrum:

- new transport protocol, more explicit feedback from routers

High speed TCP

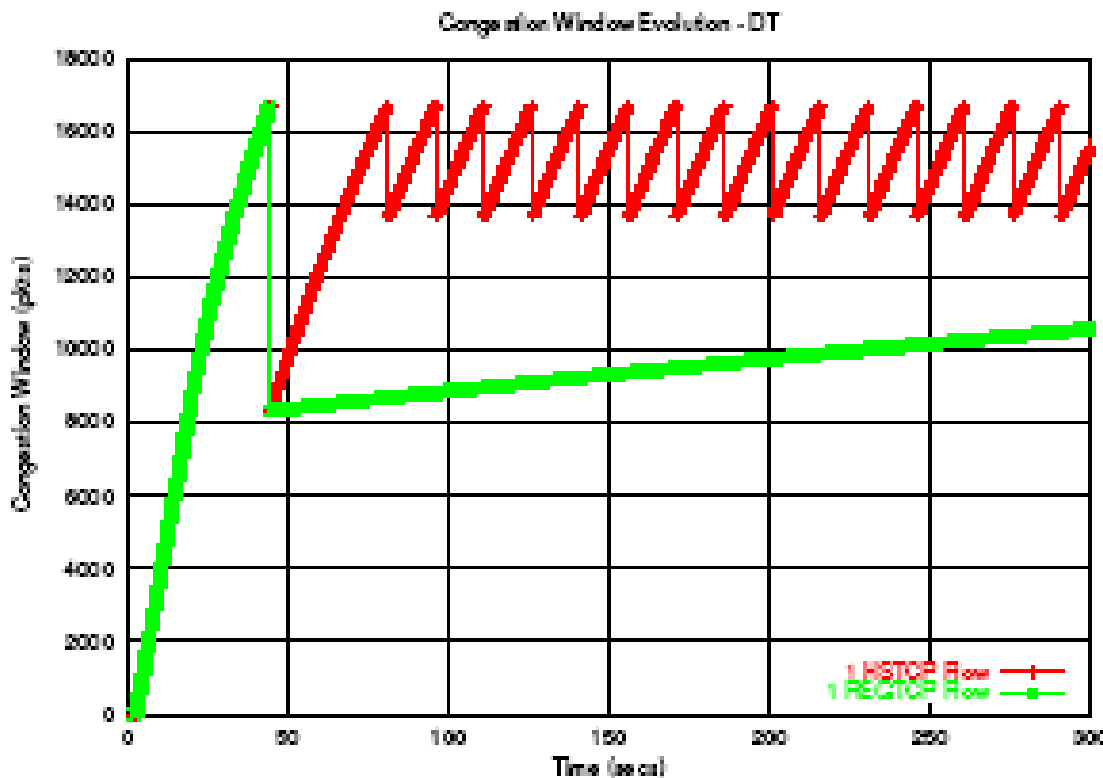
HighSpeed TCP: the modified response function.



High speed TCP

- additive increase, multiplicative decrease
- increments, decrements depend on window size

w	a(w)	b(w)
38	1	0.50
118	2	0.44
221	3	0.41
347	4	0.38
495	5	0.37
663	6	0.35
851	7	0.34
1058	8	0.33
1284	9	0.32
1529	10	0.31
1793	11	0.30
2076	12	0.29
2378	13	0.28
...		
84035	71	0.10



Scalable TCP (STCP)

- multiplicative increase, multiplicative decrease

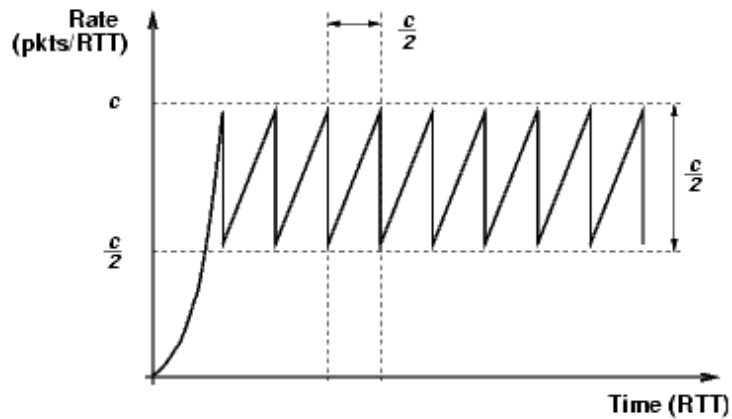
$$W \leftarrow W + a$$

per ACK

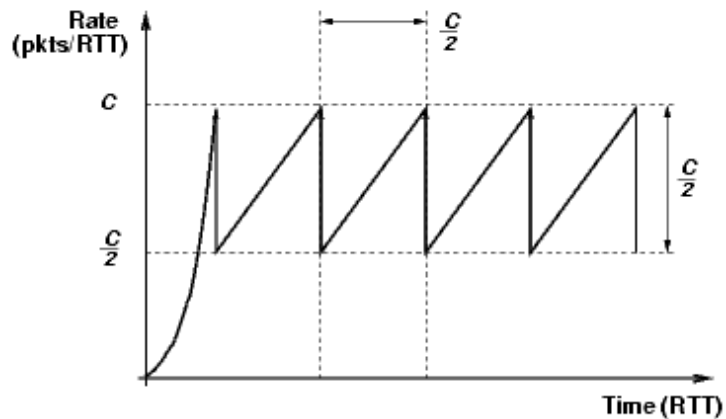
$$W \leftarrow W - b W$$

per window experiencing loss

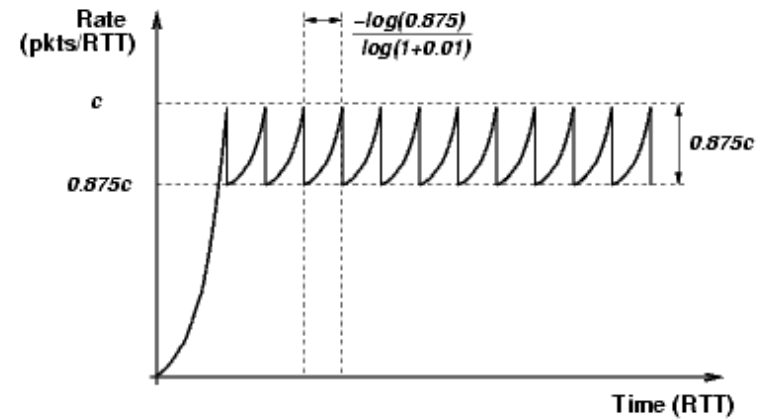
STCP in images



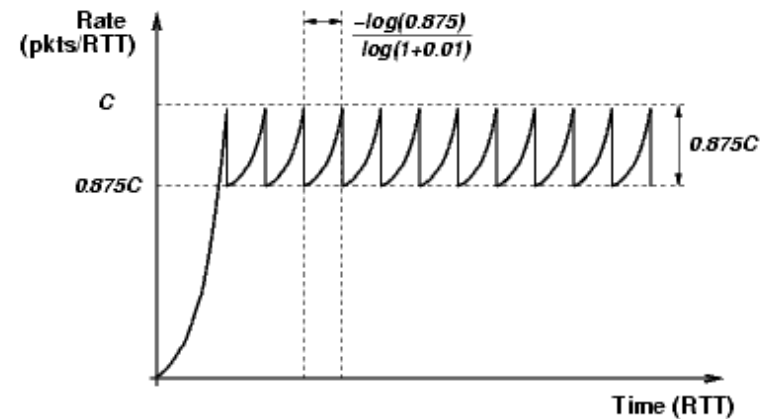
Traditional TCP with small capacity



Traditional TCP with large capacity



Scalable TCP with small capacity



Scalable TCP with large capacity

From 1st PFLDnet Workshop, Tom Kelly