

TCP e controllo di congestione

- ❑ modelli di TCP
- ❑ varianti di TCP
 - Cubic, Multipath TCP, Scalable TCP
- ❑ controllo di congestione e allocazione distribuita delle risorse
- ❑ programmazione non-lineare

Studio delle prestazioni di TCP

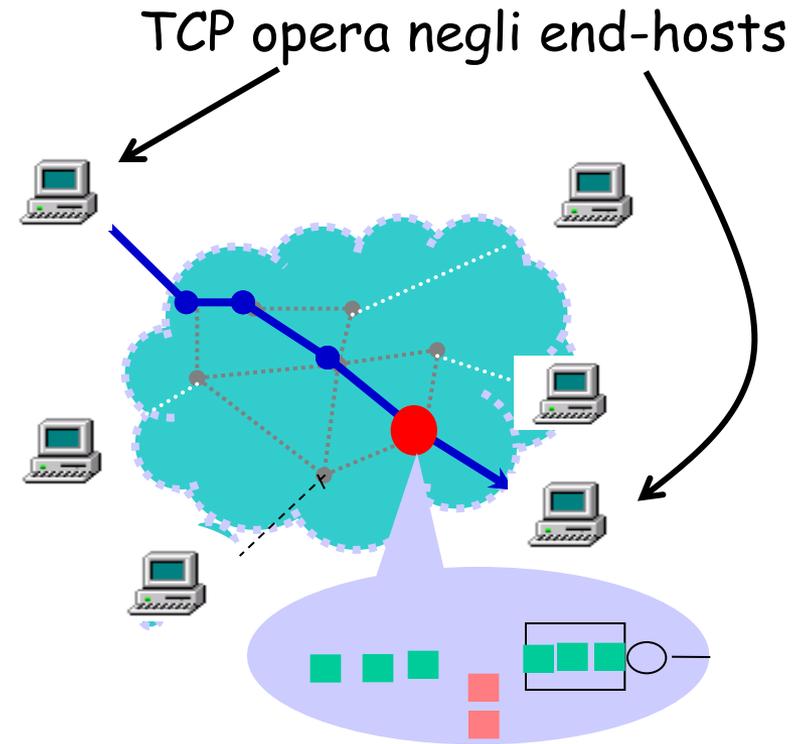
- approccio simulativo/emulativo
 - + fedeltà al comportamento vero di TCP
 - costoso in termini di tempo/sforzi
- modelli deterministici approssimati
 - + semplici, veloci
 - ignorano dettagli di TCP, assumono steady-state
- modelli fluidi/stocastici
 - + modellano anche comportamento transitorio
 - comunque approssimati (ignorano alcuni dettagli di TCP)

Studio delle prestazioni di TCP

- approccio simulativo/emulativo
 - + fedeltà al comportamento vero di TCP
 - costoso in termini di tempo/sforzi
- modelli deterministici approssimati
 - + semplici, veloci
 - ignorano dettagli di TCP, assumono steady-state
- modelli fluidi/stocastici
 - + modellano anche comportamento transitorio
 - comunque approssimati (ignorano alcuni dettagli di TCP)

Ruolo di TCP

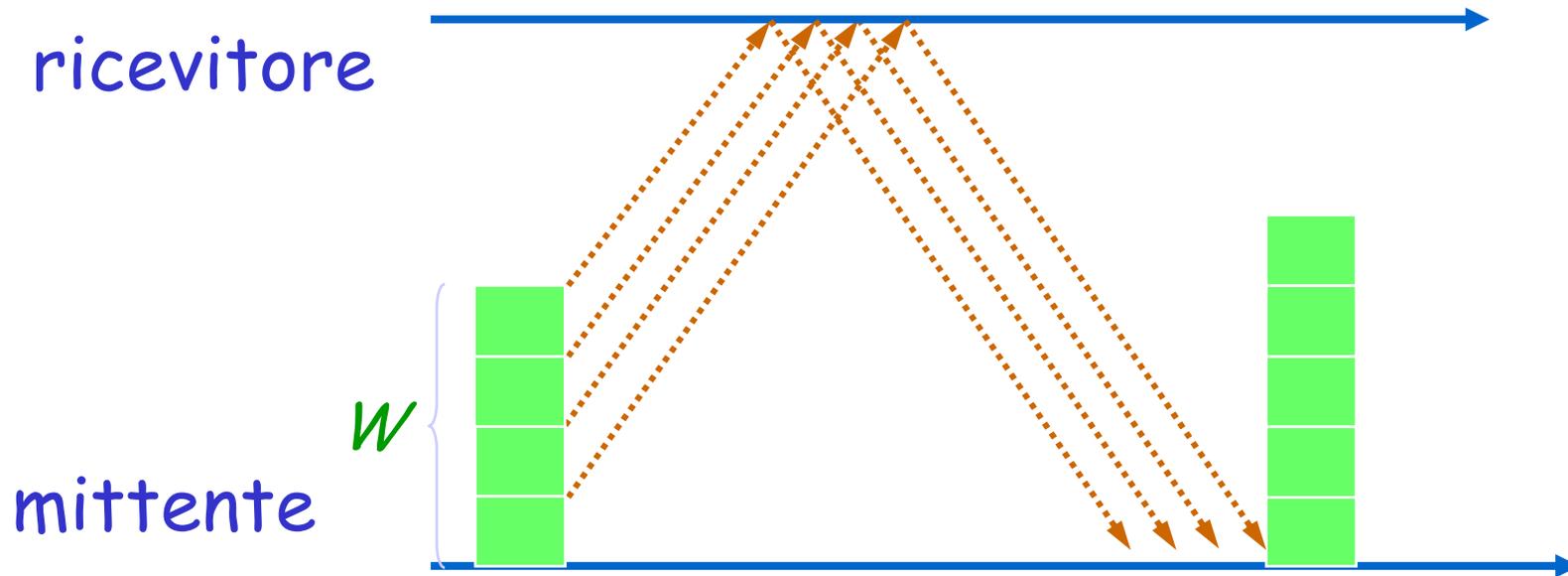
- controllo della congestione:
 - si diminuisce la velocità di trasmissione quando si rilevano perdite, la si aumenta in assenza di perdite
- router
 - scartano/marcano pacchetti in presenza di congestione
- interazione tra sistemi terminali (TCP) e router?
 - come si comprende (quantifica) questa interazione?



router congestionato scarta i pacchetti

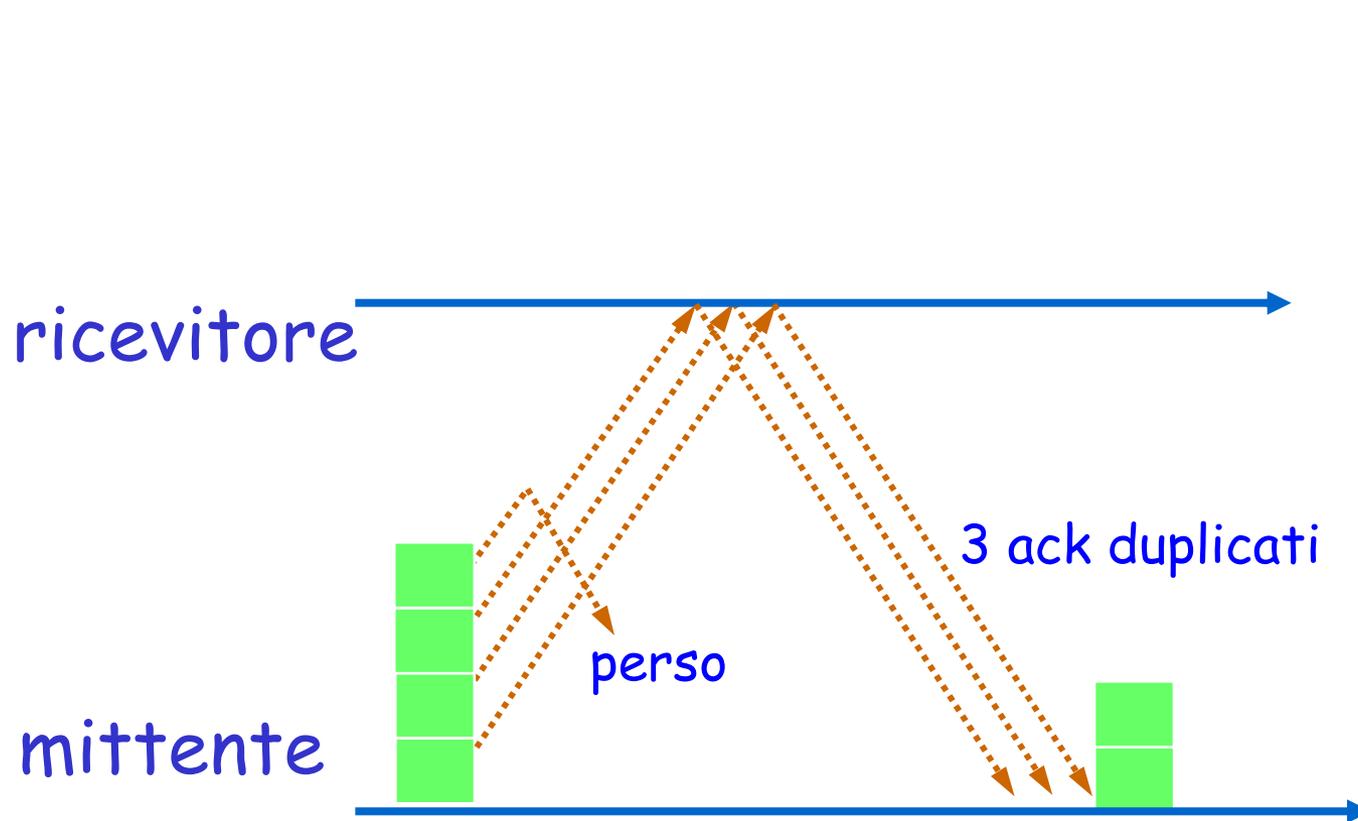
Comportamento ad alto livello di TCP

- algoritmo a finestra (*window W*)
 - al più W pacchetti "fuori nella rete"
 - ricezione di ACK permette al trasmettitore di inviare un nuovo pacchetto
 - uso di ACK cumulativi
- fase di congestion avoidance: finestra aumenta di uno ogni RTT
 - $W \leftarrow W + 1 / W$ per ACK
 - $\Rightarrow W \leftarrow W + 1$ per RTT
- ricerca della banda disponibile



Comportamento ad alto livello di TCP

- algoritmo a finestra (window W)
- aumento della finestra di 1 ogni RTT
 $W \leftarrow W + 1 / W$ per ACK
- perdite come indicazione di congestione
- dimezzamento della finestra quando si rileva una perdita (3 ACK duplicati), $W \leftarrow W/2$

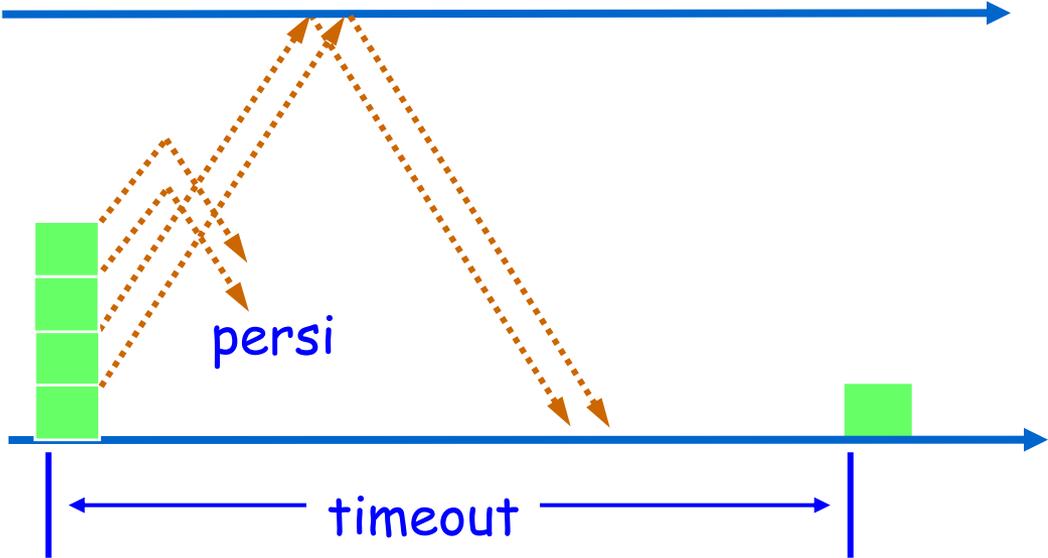


Comportamento ad alto livello di TCP

- algoritmo a finestra (window W)
- aumento della finestra di 1 ogni RTT
 $W \leftarrow W + 1/W$ per ACK
- dimezzamento finestra dopo 3 ACK duplicati,
 $W \leftarrow W/2$
- timeout in mancanza di ACK \rightarrow finestra ridotta a uno, $W \leftarrow 1$

ricevitore

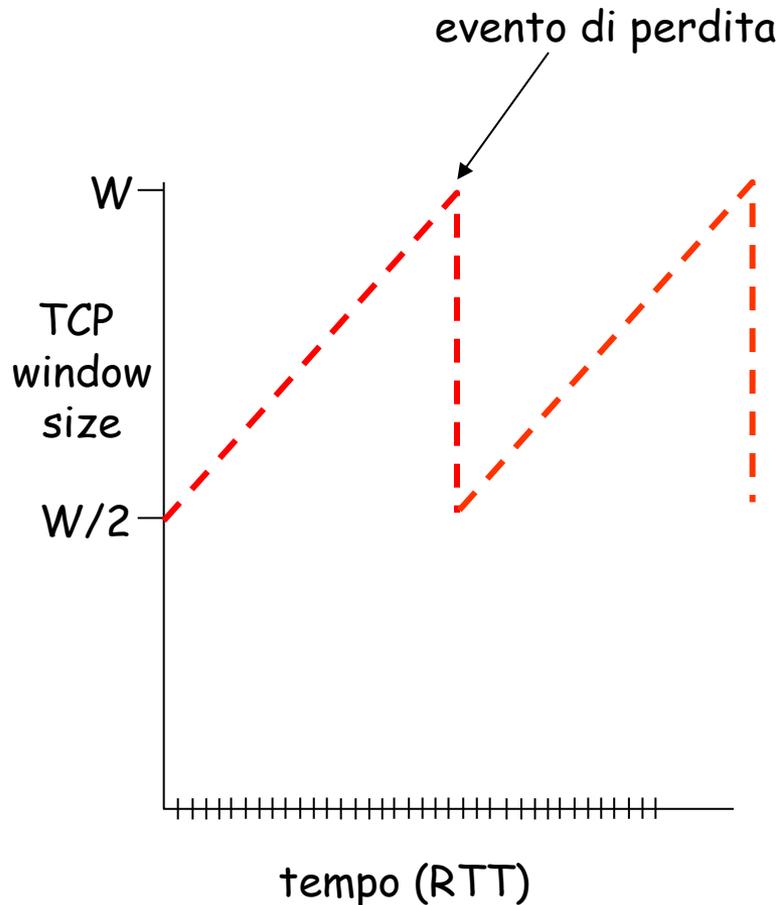
mittente



Comportamento ad alto livello di TCP

- ❑ algoritmo a finestra (window W)
- ❑ aumento della finestra di 1 ogni RTT
 $W \leftarrow W + 1/W$ per ACK
- ❑ dimezzamento finestra dopo 3 ACK duplicati,
 $W \leftarrow W/2$
- ❑ timeout in mancanza di ACK \rightarrow finestra ridotta a uno, $W \leftarrow 1$
- ❑ ulteriori timeout consecutivi raddoppiano di durata, fino a 6 volte

TCP: relazione throughput/probabilità di perdita

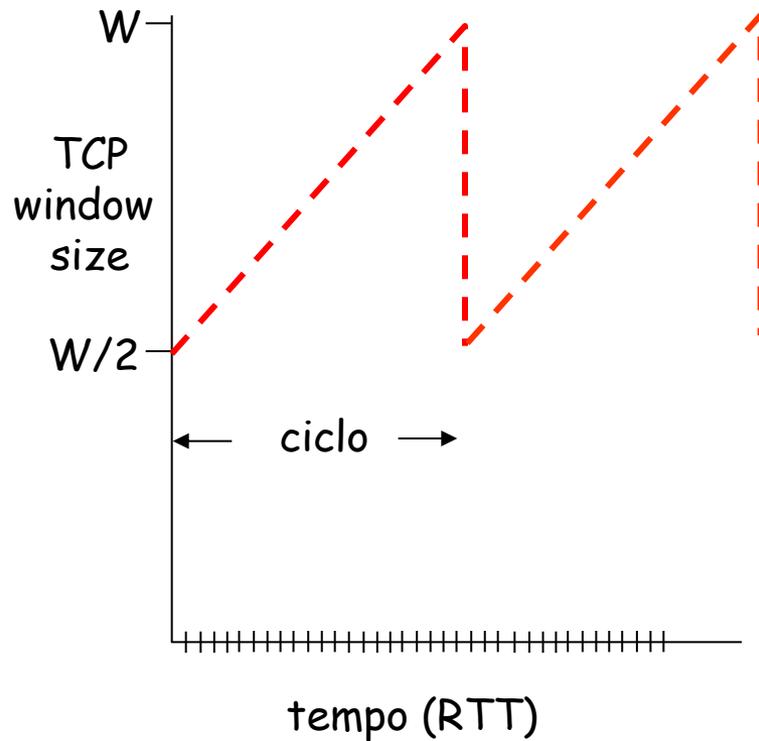


modello idealizzato:

- W dimensione massima supportabile della finestra (oltre avviene perdita)
- finestra di TCP parte da $W/2$, cresce fino a W , si dimezza, cresce di nuovo, e così via...
- una "finestrata" di pacchetti ogni RTT
- *trovare*: throughput in funzione di prob. di perdita e RTT

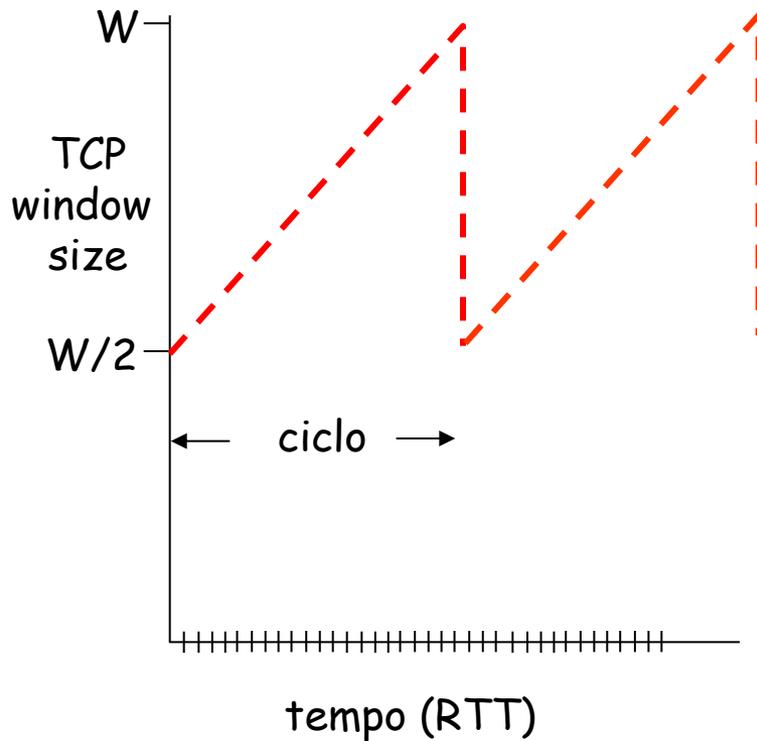
TCP: relazione throughput/probabilità di perdita

pacchetti trasmessi per "ciclo" =



TCP: relazione throughput/probabilità di perdita

pacchetti trasmessi per "ciclo" =



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

TCP: relazione throughput/probabilità di perdita

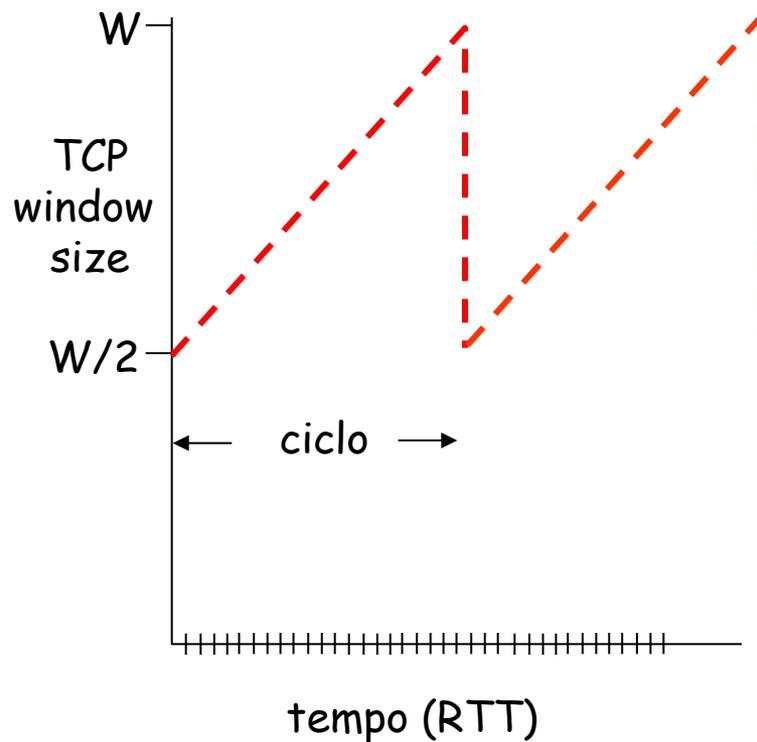
$$\# \text{ pacchetti trasmessi per "ciclo"} \approx \frac{3}{8} W^2$$

1 pacchetto perso per "ciclo" implica:

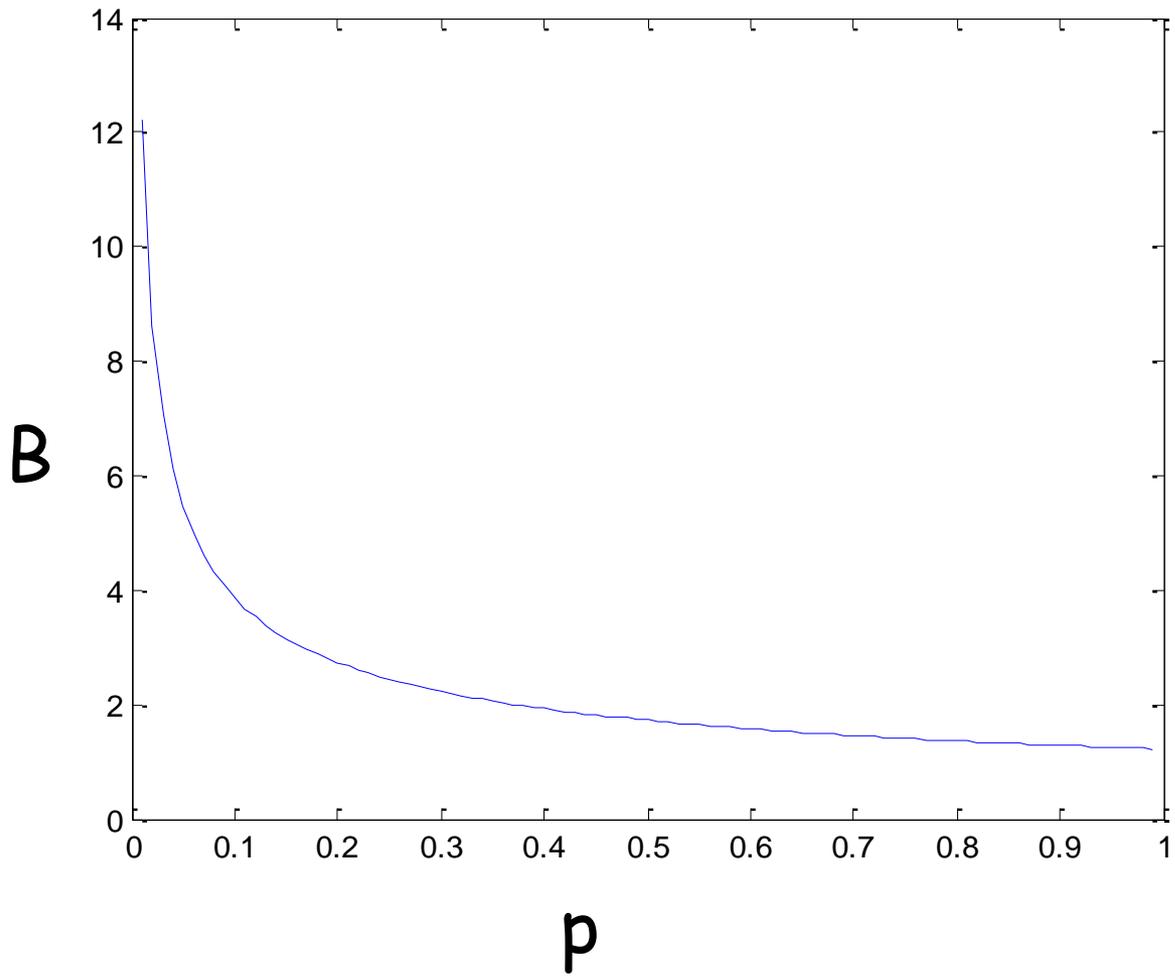
$$p_{\text{loss}} \approx \frac{8}{3W^2} \quad \text{ovvero: } W = \sqrt{\frac{8}{3p_{\text{loss}}}}$$

$$B = \text{avg_throughput} = \frac{3}{4} W \frac{\text{packets}}{\text{RTT}}$$

$$B = \text{avg_throughput} = \frac{1.22}{\sqrt{p_{\text{loss}}}} \frac{\text{packets}}{\text{RTT}}$$



Abbiamo ignorato: slow start, timeouts, dimensione max della finestra...



Modello più dettagliato per il throughput di TCP: formula PFTK

- J. Padhye, V. Firoiu, D. Towsley, J. Kurose, "Modeling TCP Throughput: a Simple Model and its Empirical Validation", Sigcomm 1998.
- Data la probabilità di perdita di un pacchetto p , la prob. di perdita di un burst bp , il round trip time RTT , il primo timeout T_0 , la finestra massima W_{max} , il throughput B di TCP in pacchetti/sec si può approssimare con la formula:

$$B(p) \approx \min \left(\frac{W_{max}}{RTT}, \frac{1}{RTT \sqrt{\frac{2bp}{3}} + T_0 \min \left(1, 3\sqrt{\frac{3bp}{8}} \right) p(1 + 32p^2)} \right) \quad (32)$$

Idee base formula PFTK

- Analisi di ciclo, che ora può comprendere anche timeouts (avventualmente consecutivi)

$$B = \text{avg_throughput} = \frac{\text{numero medio di pacchetti trasferiti con successo in un ciclo}}{\text{durata media di un ciclo}}$$

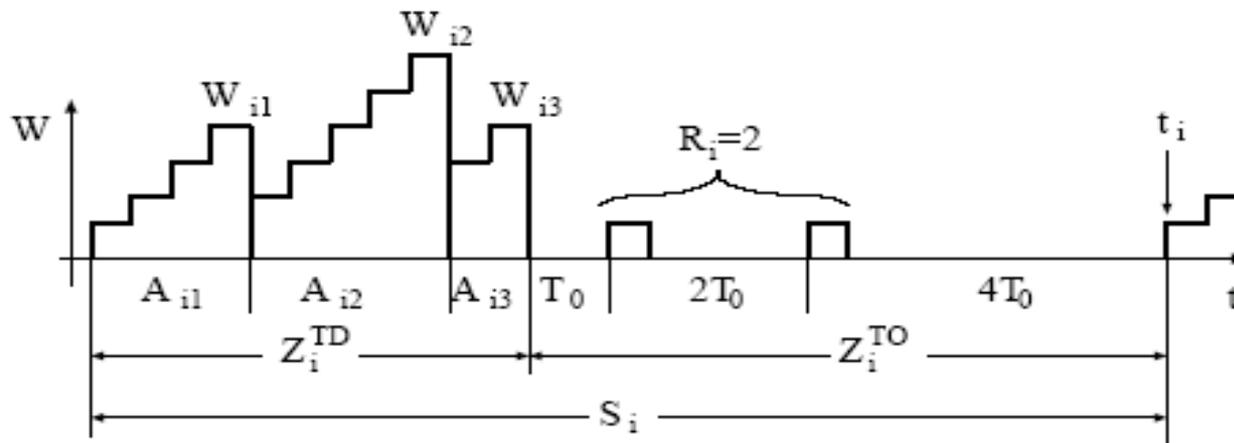


Figure 3: Evolution of window size when loss indications are triple-duplicate ACKs and time-outs

Idee base formula PFTK

- Prob. di perdita indipendente da RTT a RTT
 - perdite a burst: se un pacchetto viene perso, tutti i pacchetti consecutivi nella stessa finestra sono persi
- Timeout se si ricevono meno di tre ack duplicati

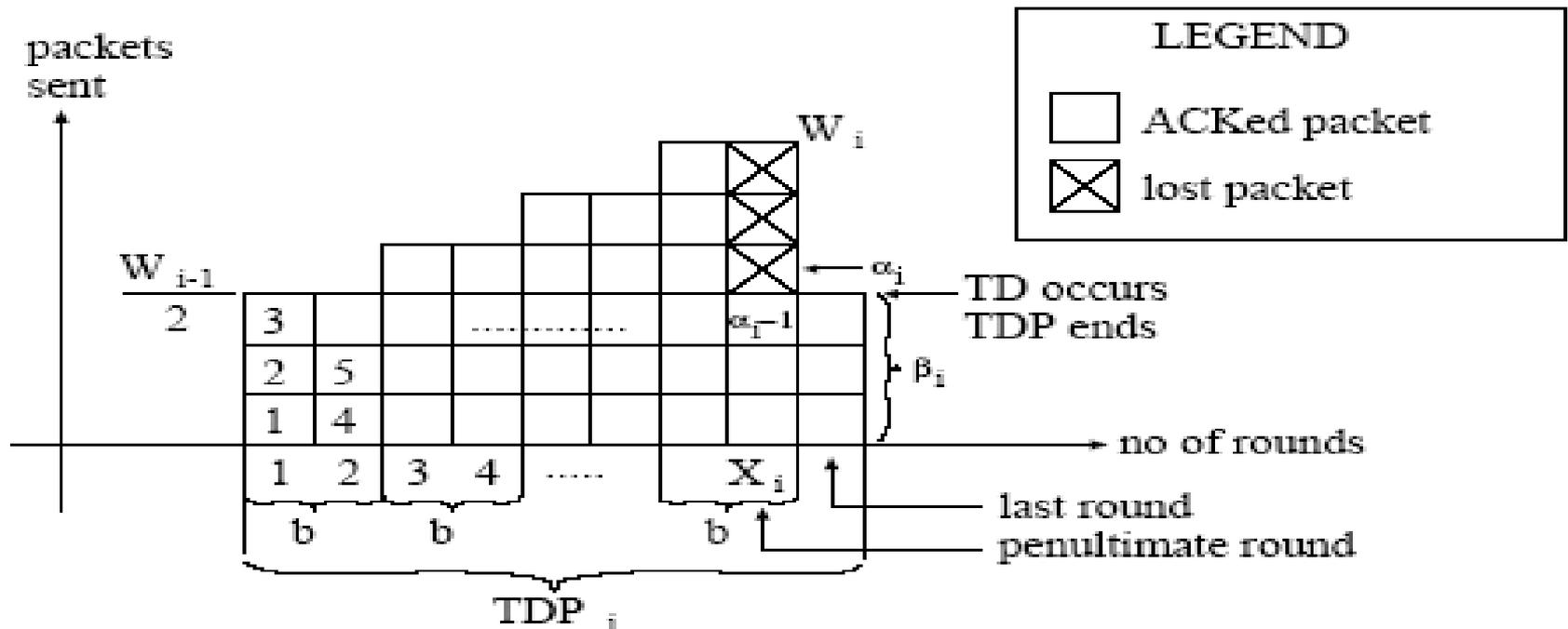


Figure 2: Packets sent during a TD period

Validazione sperimentale PFTK

$$B(p) \approx \min \left(\frac{W_{max}}{RTT}, \frac{1}{RTT \sqrt{\frac{2bp}{3}} + T_0 \min \left(1, 3 \sqrt{\frac{3bp}{8}} \right) p(1 + 32p^2)} \right) \quad (32)$$

- Throughput misurato in intervalli di 100 sec, classificati in 5 categorie:
 - TD: nessun timeout sofferto (solo triple-duplicate ack)
 - T0: sofferto almeno un timeout fondamentale (no exp backoff)
 - T1: sofferto un exp backoff (double timeout)
 - T2: sofferti due exp backoffs
 - T3 o più: sofferti tre o più exp backoff

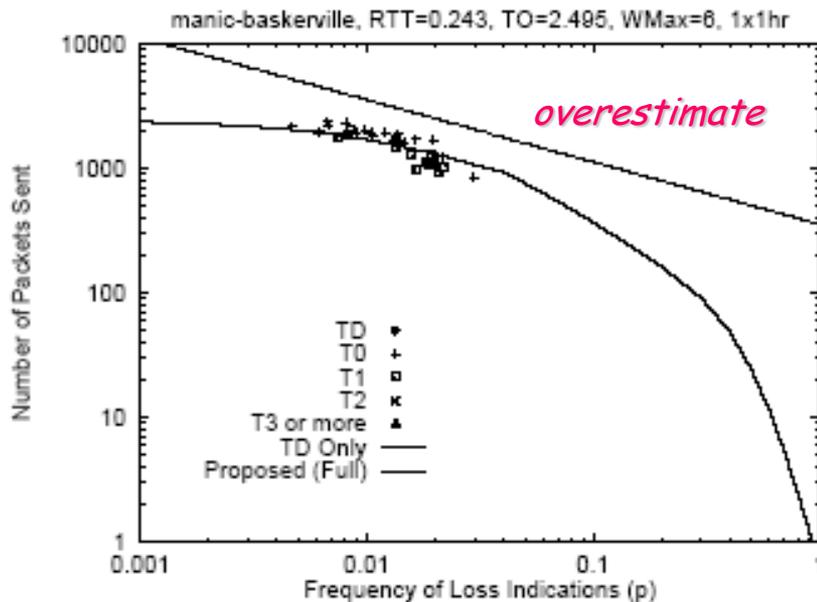


Figure 7: manic to baskerville

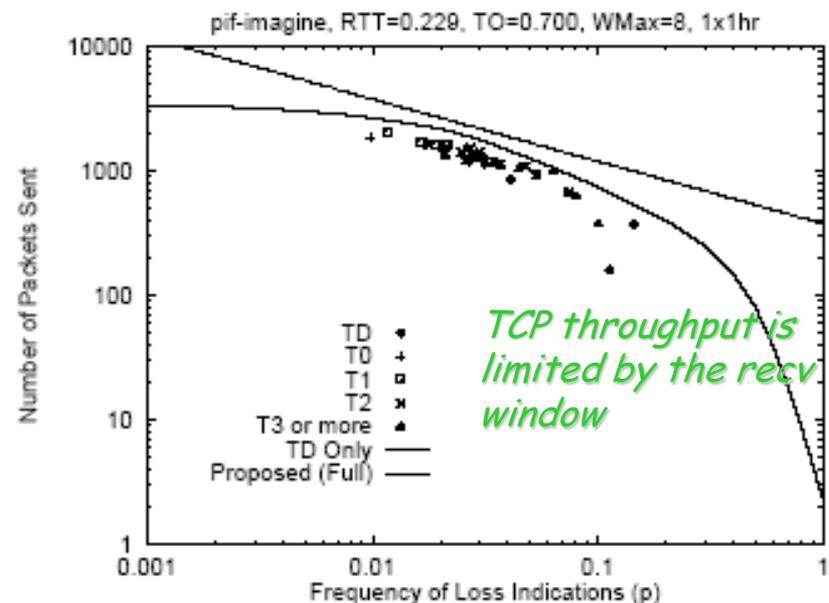


Figure 8: pif to imagine

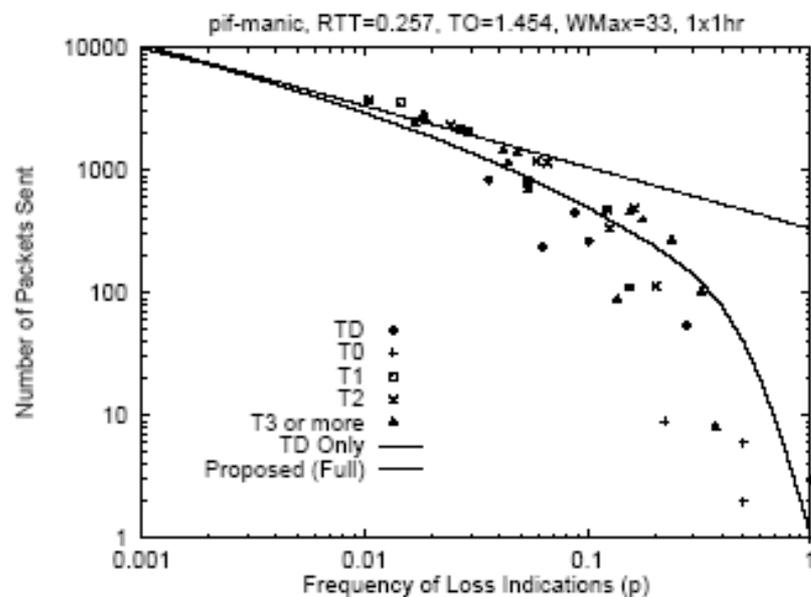


Figure 9: pif to manic

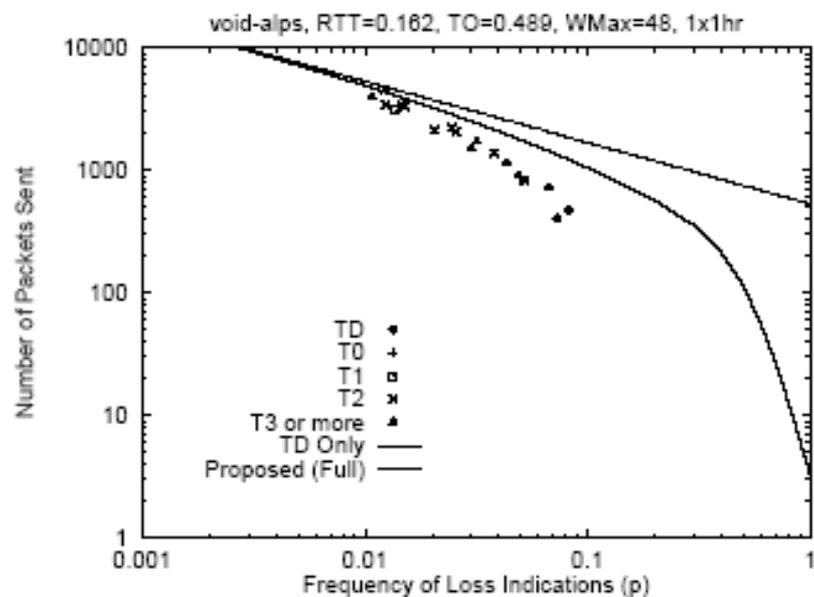


Figure 10: void to alps

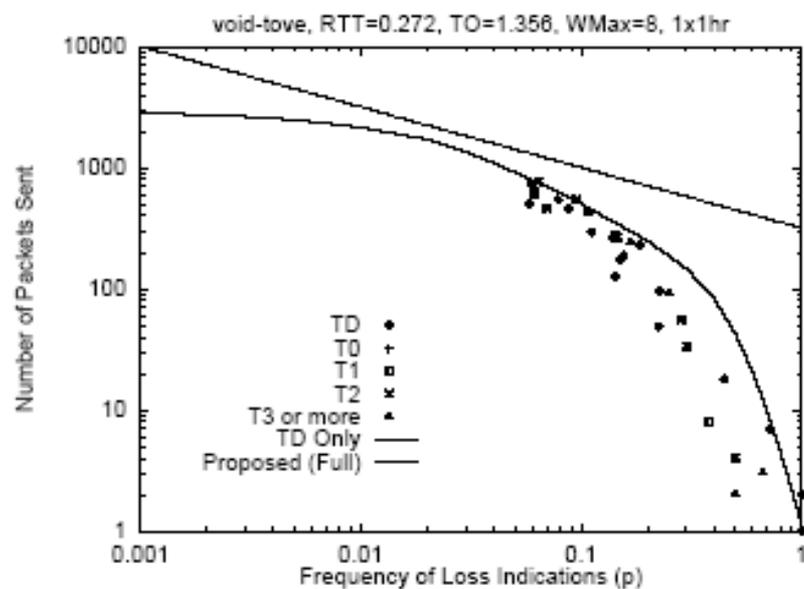


Figure 11: void to tove

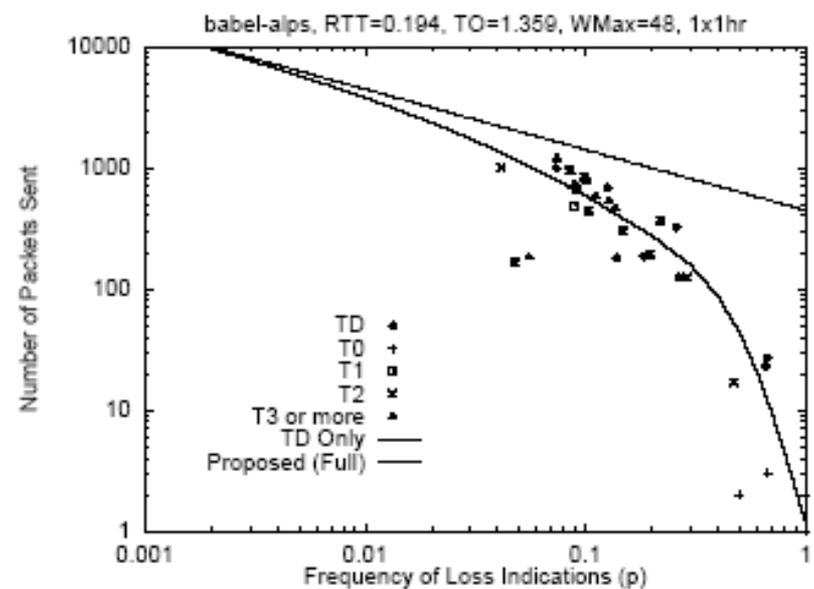


Figure 12: babel to alps

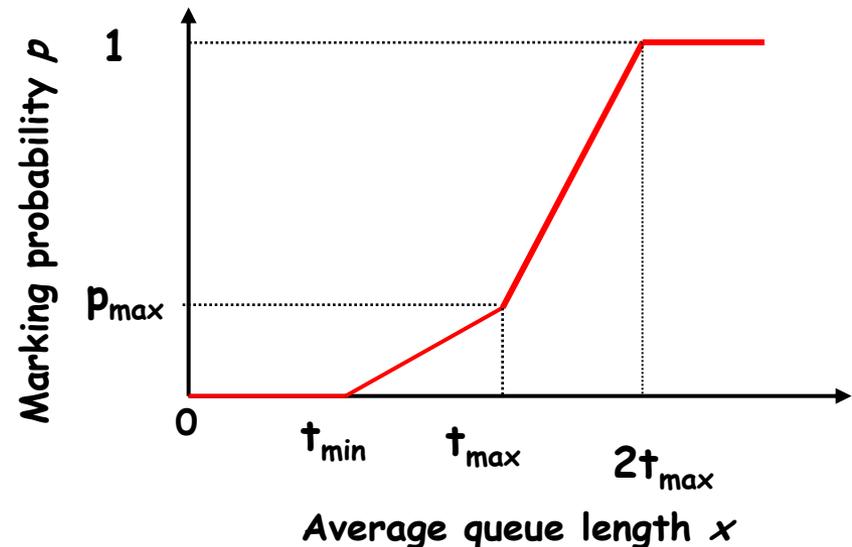
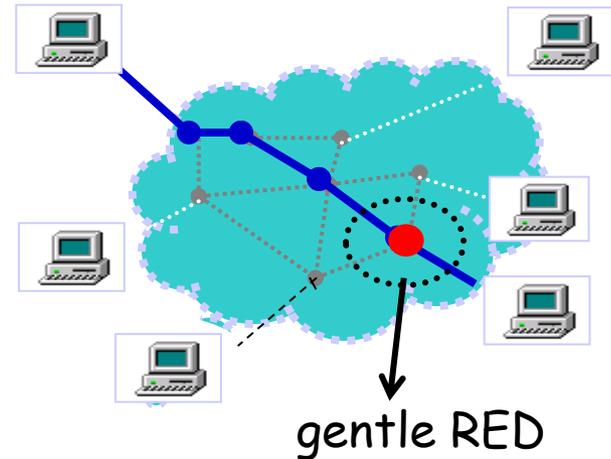
Finora: analisi di un singolo flusso TCP, supponendo nota (e indipendente) la risposta della rete (analisi ad "anello aperto")

Come si studia l'interazione di (tanti) flussi TCP con la rete?
(analisi ad "anello chiuso")

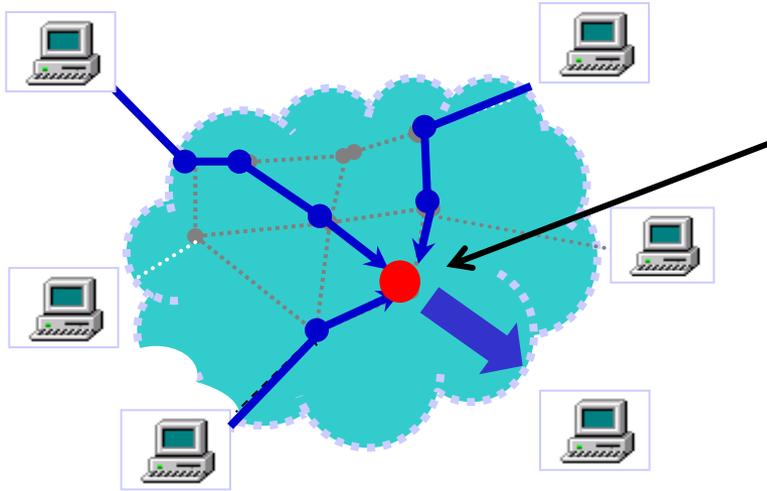
Ripasso di gestione della coda RED

- prob. p di scarto/mark dei pacchetti dipende da lunghezza media della coda x
 $\rightarrow p = p(x)$

- più in generale: active queue management (AQM)



Comportamento a singolo collo di bottiglia (bottleneck)



router di bottleneck:

- un link di uscita pienamente utilizzato
- tutti flussi TCP che lo attraversano subiscono stessa prob. di perdita
- D : i flussi ottengono lo stesso throughput?

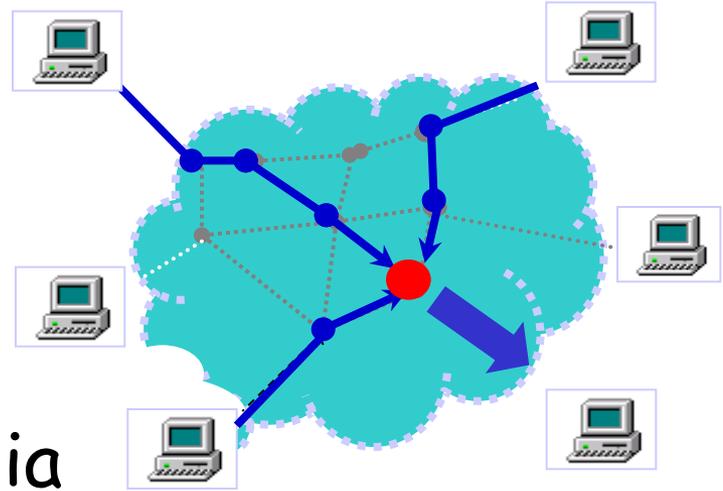
$$\sum_i B_i(p, RTT_i) = C$$

C - link bandwidth

B_i - throughput of flow i

Singolo collo di bottiglia: flussi "infiniti"

- N flussi TCP con infiniti pacchetti da trasmettere
 - Ritardo di prop. a due vie $A_i, i = 1, \dots, N$
 - throughput $B_i(p, RTT_i)$
- un singolo collo di bottiglia
 - gestione della coda RED:
 - lunghezza media della coda x ; prob. di perdita di pacchetti $p(x)$
- si vuole ottenere:
 - B_i : throughput del i -esimo flusso TCP
 - risposta del router, ovvero, x e $p(x)$



Modello e soluzione

- modello

$$p = p(x) \quad (AQM)$$

$$RTT_i = A_i + x / C \quad (\text{round trip time})$$

$$\sum_i B(p, RTT_i) = C, \quad \text{per } i=1, \dots, N$$

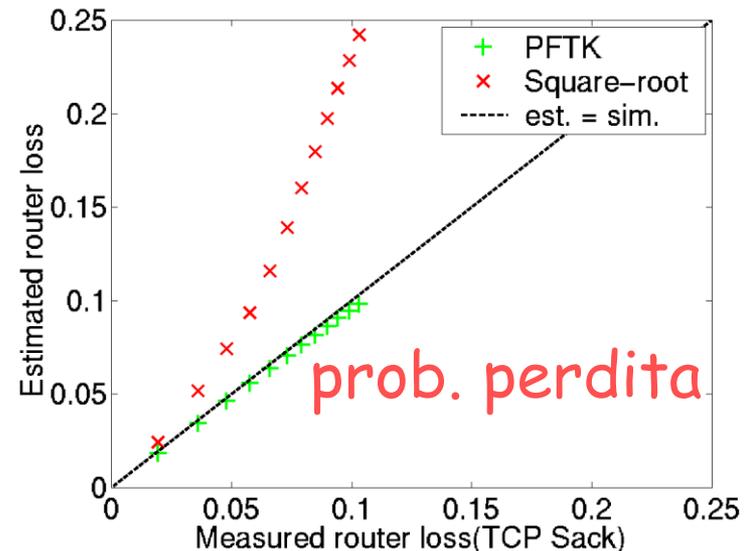
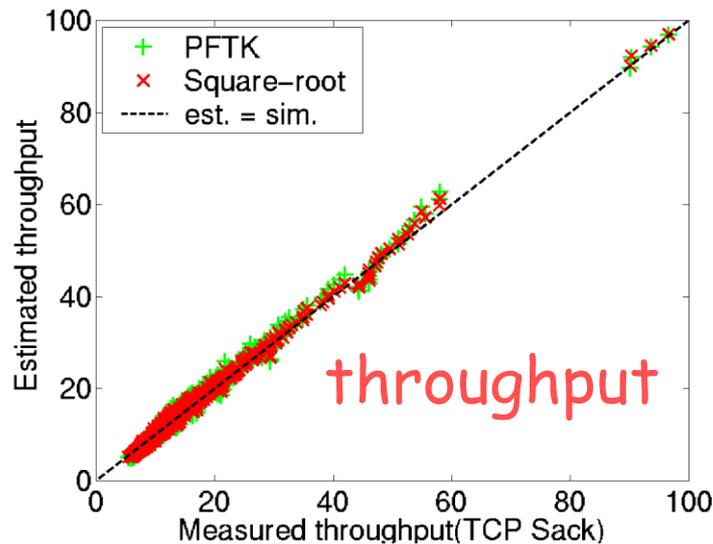
- soluzione a "punto fisso" (fixed point problem)
per x

- x usata nel calcolo di RTT_i e p

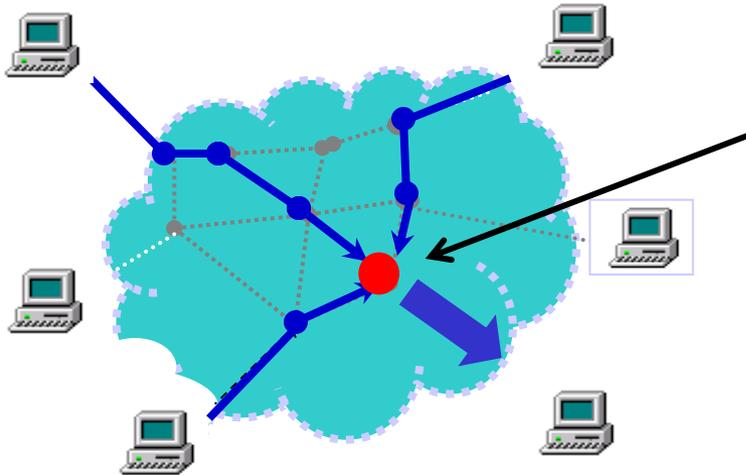
- unica soluzione purchè B monotona e continua
rispetto a x

Modello vs simulazione: singolo collo di bottiglia, flussi TCP infiniti

- capacità del link 4 Mbps e parametri RED assegnati
- 10-120 flussi TCP
- ritardi di prop. a due vie $A_i = 20+2i$ ms, $i = 1, \dots, N$



Modello a collo di bottiglia



router di bottleneck:

- capacità del link pienamente utilizzata
- tutti i flussi TCP subiscono la stessa prob. di perdita

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

C - capacità del link

B_i - throughput del flusso i

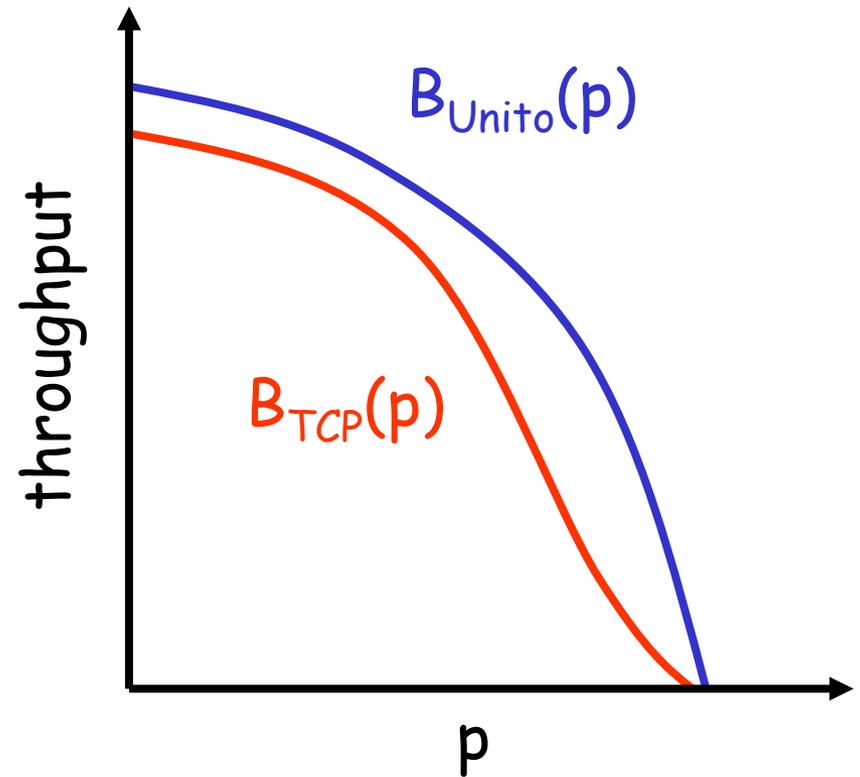
Applicazioni del modello a collo di bottiglia

- confronto di diversi algoritmi di controllo della congestione
- è utile fare forward error correction?

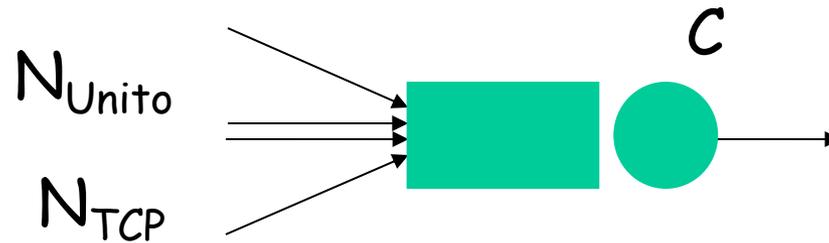
Una nuova versione (migliorata) di TCP: TCP Unito

- TCP "Unito", $B_{\text{Unito}}(p)$
- TCP, $B_{\text{TCP}}(p)$

$$B_{\text{Unito}}(p) > B_{\text{TCP}}(p)$$



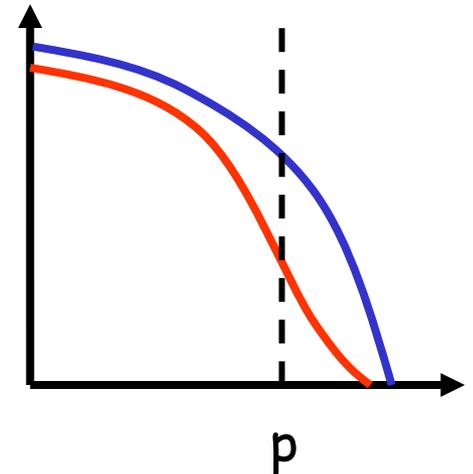
Condivisione del collo di bottiglia con versione standard di TCP



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

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

evviva!



Se si rimpiazza TCP con TCP Unito...

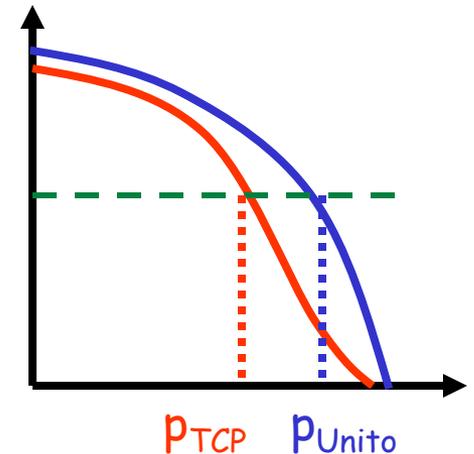
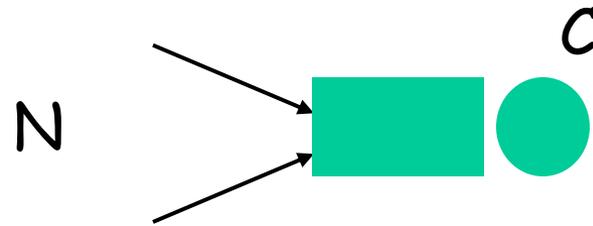
$$NB_{\text{Unito}}(p_{\text{Unito}}) = C$$

vs

$$NB_{\text{TCP}}(p_{\text{TCP}}) = C$$

$$\Rightarrow p_{\text{Unito}} > p_{\text{TCP}}$$

oh oh...

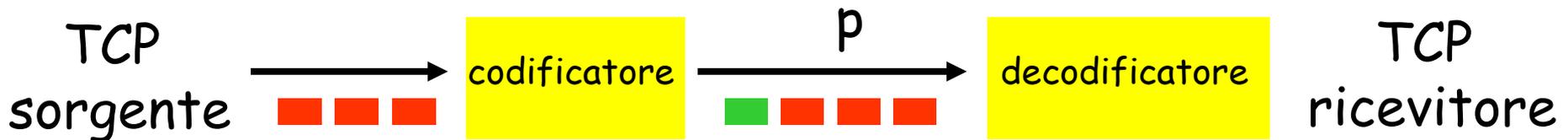
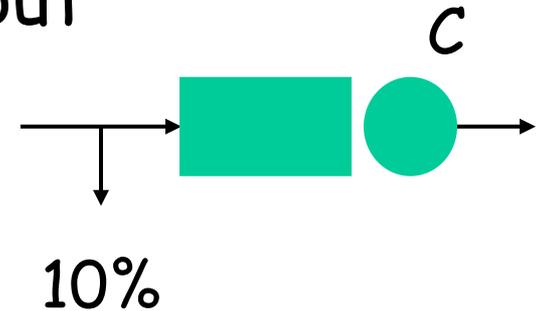


Uso di FEC

□ 10% pkt loss \Rightarrow basso throughput



allora usiamo FEC
(a livello pacchetto)!

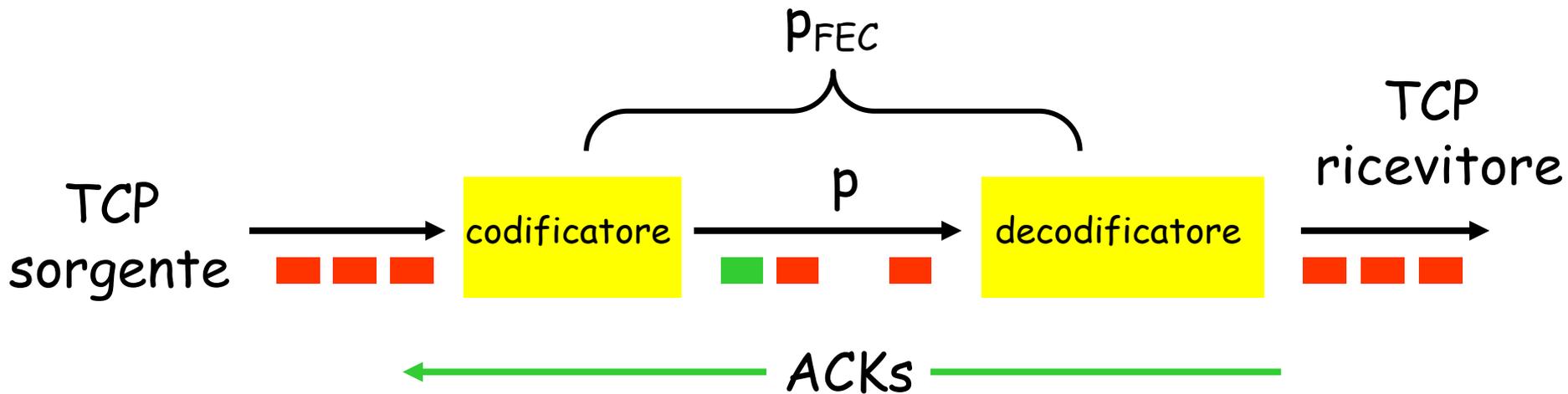
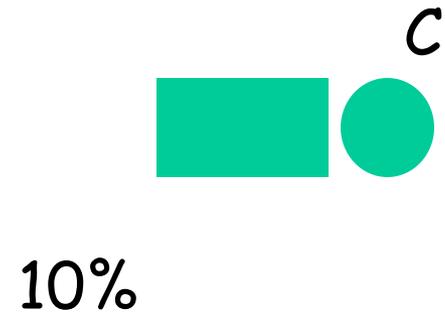


Uso di FEC

□ 10% pkt loss \Rightarrow poor thruput



use packet level FEC!



• $p_{FEC} \ll p$

Uso di FEC

□ banda disponibile, $C_{FEC} < C_{wo}$

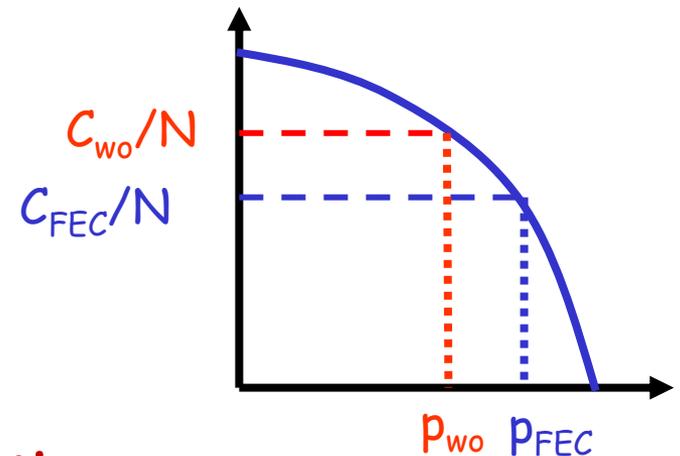
□ $B_{FEC}() = B_{wo}() = B_{TCP}()$

□ $N B_{wo}(p_{wo}) = C_{wo}$

□ $N B_{FEC}(p_{FEC}) = C_{FEC}$

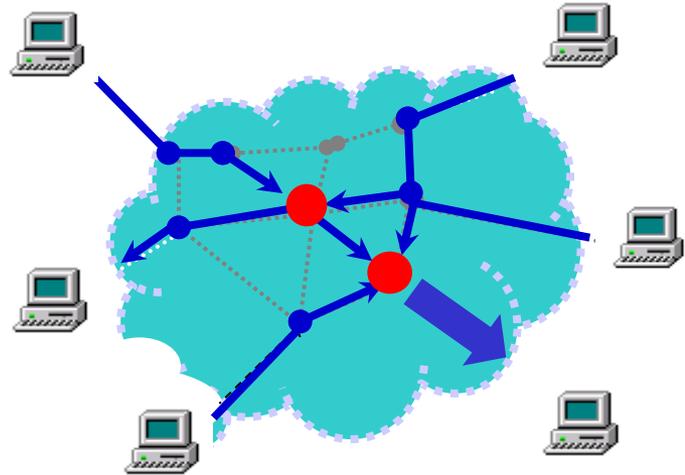
$$\Rightarrow p_{FEC} > p_{wo}$$

□ **FEC riduce le prestazioni!**



Colli di bottiglia multipli: flussi infiniti

- N flussi TCP
 - throughputs $B = \{B_i(R_i, p_i)\}$
- V router AQM congestionati
 - capacità $C = \{C_v\}$
 - code medie $x = \{x_v\}$
 - prob. perdita $p = \{p_v(x_v)\}$



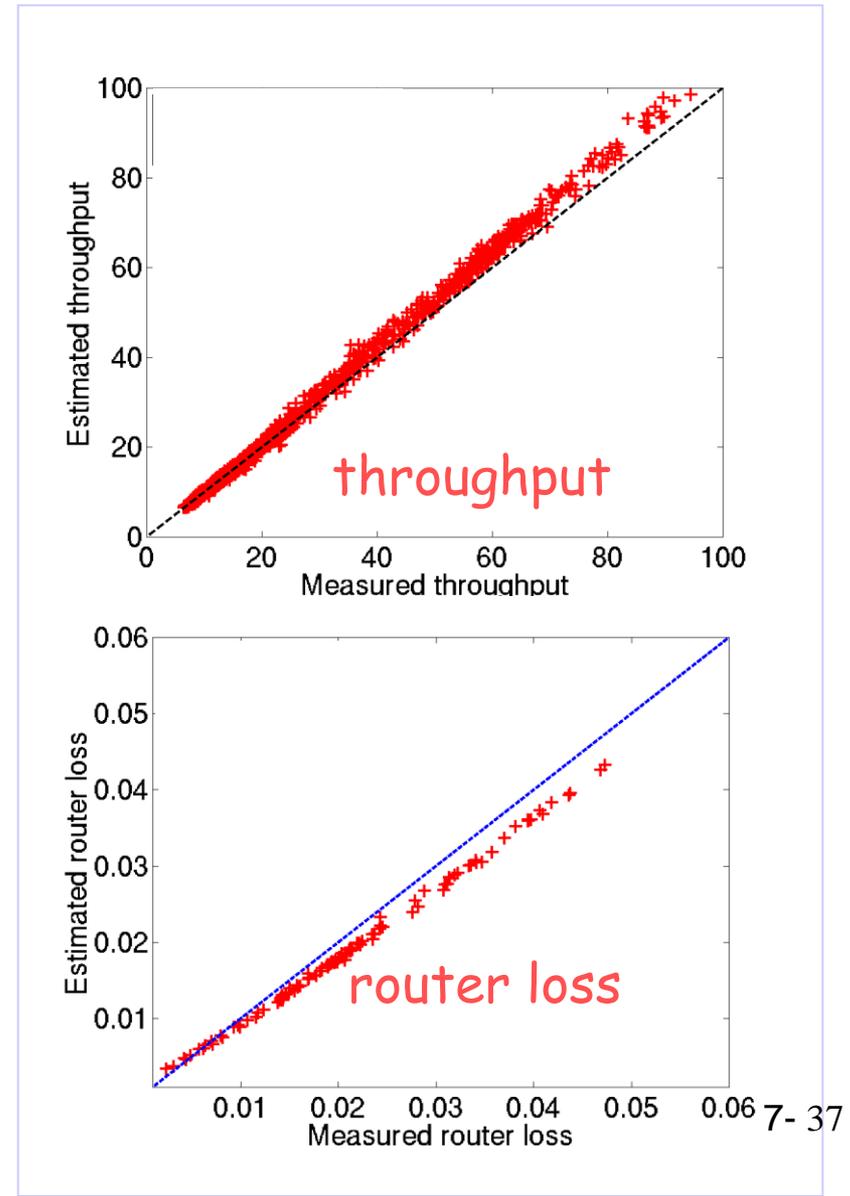
per ogni collo di bottiglia:

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

V equazioni, V incognite (x_v)

Risultati: colli di bottiglia multipli, flussi infiniti

- colli di bottiglia in cascata, 5 -10 router
- ritardi di prop. a 2-vie 20-120 ms
- bande, 2-6 Mbps
- errori con modello PFTK
 - throughput $< 10\%$
 - prob. perdita $< 10\%$
 - code medie $< 15\%$
- risultati simili per reti magliate/con cicli



Osservazioni

- ❑ se si aggiungono flussi UDP / non-TCP ?
 - nel caso di flussi "non-responsive" (che non fanno controllo di congestione), basta sottrarre il loro rate dalla capacità del collo di bottiglia
- ❑ se si aggiungono flussi TCP finiti?
 - più complesso
 - trattabili come un "rumore"
- ❑ nota: nel calcolo del throughput, abbiamo assunto che il tempo di trasmissione di una finestra è minore del RTT (non vero con RTT estremamente piccoli - flussi in locale)

Analisi dinamica (in transitorio) di popolazioni di flussi TCP

- si modella il traffico TCP come un **fluido**
- si descrive il comportamento di flussi e code usando **Equazioni Differenziali Ordinarie (Ordinary Differential Equations, ODE)**
- si risolvono le ODEs **numericamente**

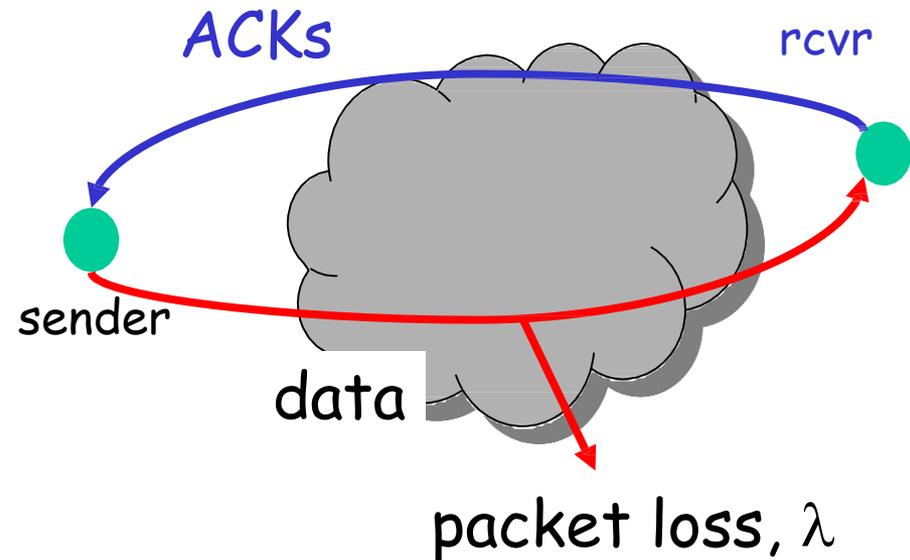
La rete come una scatola nera

- La rete introduce
 - ❖ round trip time R
 - ❖ perdite, Poisson(λ)
- Ipotesi Poisson verificata da misure sperimentali
- \bar{W} – dimensione media della finestra

$$\frac{d\bar{W}}{dt} = \frac{1}{R} - \frac{\bar{W}}{2} \times \lambda$$

as $t \rightarrow \infty$

$$\bar{W} = \frac{2}{\lambda R}$$



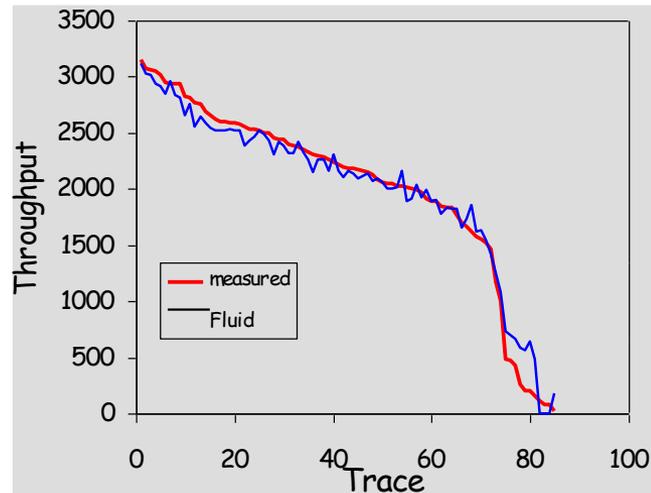
La rete come una scatola nera

□ Throughput TCP

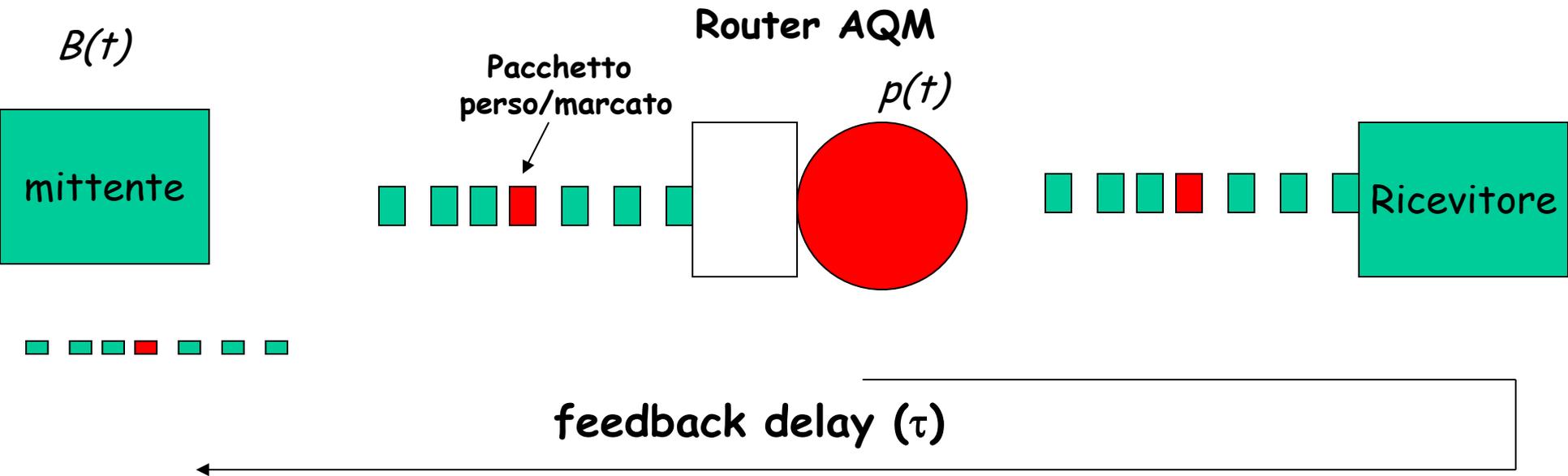
$$T_{put} = \frac{\overline{W}}{R}$$

□ validazione tramite misure

- ❖ 85 tracce di rete della durata di 100 sec. (scenari wide area)
- ❖ λ , R stimate dalle tracce



Modello a sistema di controllo in tempo continuo

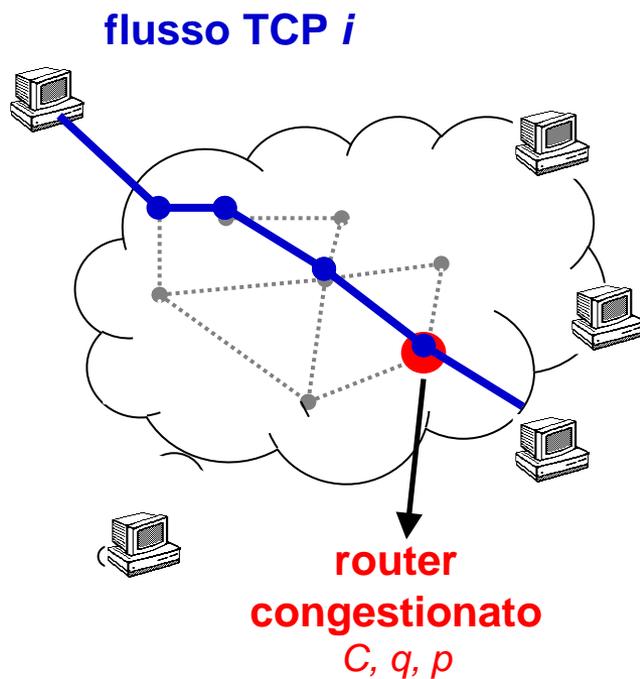


rate di ricezione di acks: $\lambda_{ack}(t) = B(t-\tau) * (1-p(t-\tau))$

rate di ricezione di

"segnali di congestione": $\lambda_{loss}(t) = B(t-\tau) * p(t-\tau)$

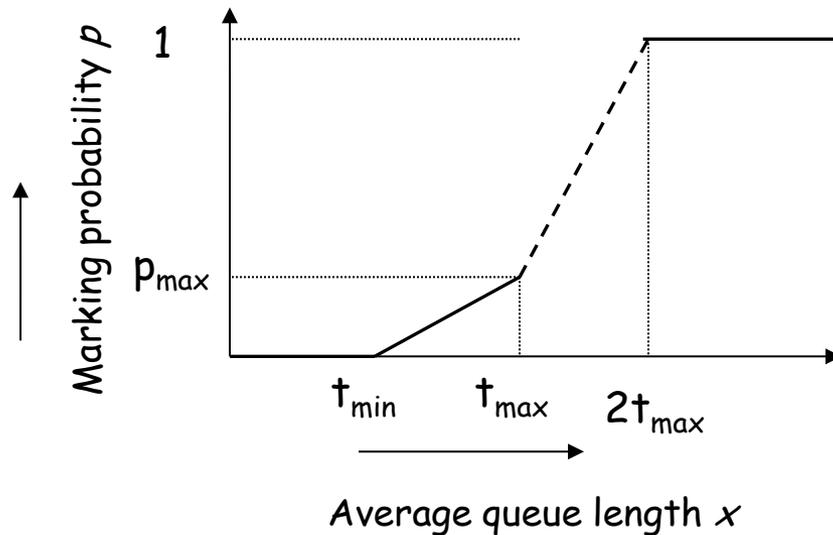
Singolo collo di bottiglia



- un solo canale congestionato
 - capacità C (*pacchetti/sec*)
 - lunghezza della coda $q(t)$
 - prob. di perdita $p(t)$
- N flussi TCP infiniti
 - window sizes $W_i(t)$
 - round trip times
 - $R_i(t) = A_i + q(t)/C$
 - throughputs
 - $B_i(t) = W_i(t)/R_i(t)$

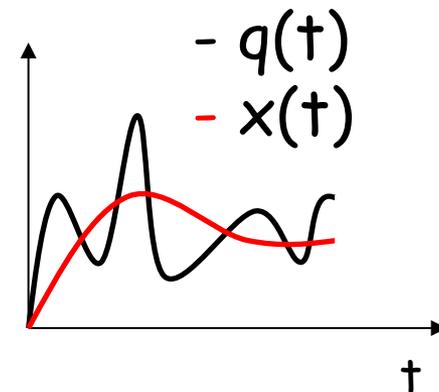
Il caso di router RED

RED: scarto/marcatura di pacchetti in base a lunghezza della coda $x(t)$ calcolata con media mobile esponenziale



versione "gentle RED"

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



Sistema di equazioni differenziali

(timeouts e slow start ignorati)

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 Multiplicative decrease

lunghezza
istantanea
della coda

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

traffico in uscita traffico in ingresso

Sistema di Equazioni Differenziali

se ignoriamo il feedback delay, $\tau \rightarrow 0$
(non i round trip times!)

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

Sistema di equazioni differenziali (cont.)

Media mobile esponenziale di RED

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

Dove:

α = parametro di RED

$t_{k+1} - t_k = \delta$ = intervallo di campionamento $\sim 1/C$

Sistema di equazioni differenziali (cont.)

evoluzione della
media mobile
esponenziale:

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

dove

α = parametro di RED

δ = intervallo di campionamento $\sim 1/C$

evoluzione della
probabilità di perdita:

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

Dove $\frac{dp}{d\bar{x}}$ dipende dal profilo di RED

N+2 equazioni accoppiate

N flussi

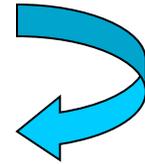
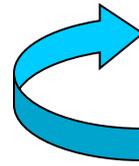
$W_i(t)$ = window size
del flusso i

$R_i(t)$ = RTT del flusso i

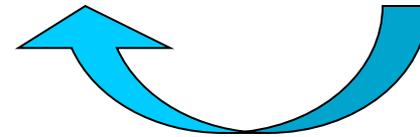
$p(t)$ = prob. perdita

$q(t)$ = lunghezza della
coda

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



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



Equazioni risolte numericamente (es: MATLAB)

Comportamento a regime

□ per $t \rightarrow \infty$

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

□ si ottiene:

□ il throughput viene

Comportamento a regime

□ per $t \rightarrow \infty$

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

□ si ottiene

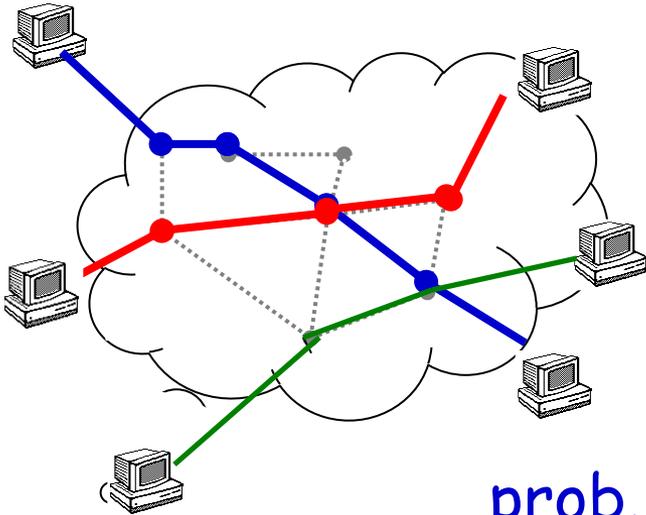
$$0 = \frac{1-p}{\bar{R}_k} - \frac{\bar{W}_k}{2} \frac{\bar{W}_k}{\bar{R}_k} p \quad \text{ovvero} \quad \bar{W}_k = \frac{\sqrt{2(1-p)}}{\sqrt{p}}$$

□ il throughput viene

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

Una coda non è una rete...

Rete - insieme di router AQM, V
sequenza V_i di router
attraversati dal flusso i



Round trip time - ritardo totale

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

prob. di perdita/marketing - (end-to-end)

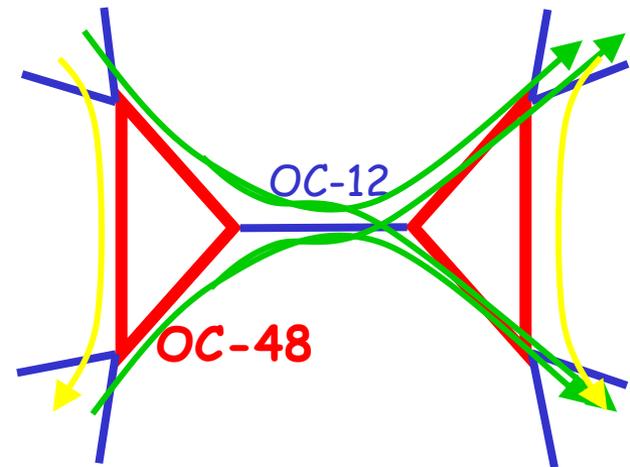
$$p_i(t) = 1 - \prod_{v \in V_i} (1 - p_v(q_v(t)))$$

vincoli di banda dei canali...

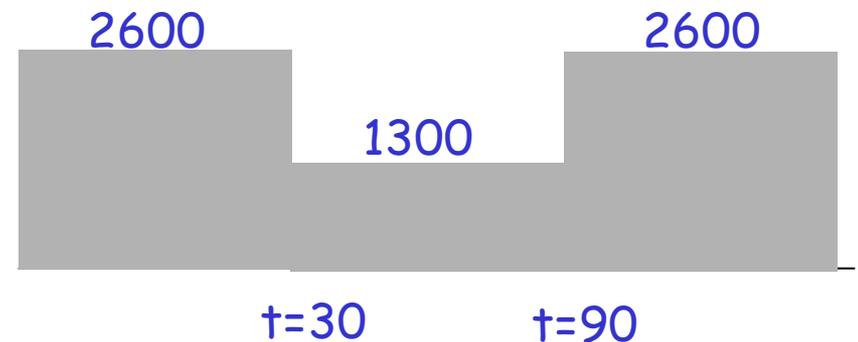
equazioni delle code...

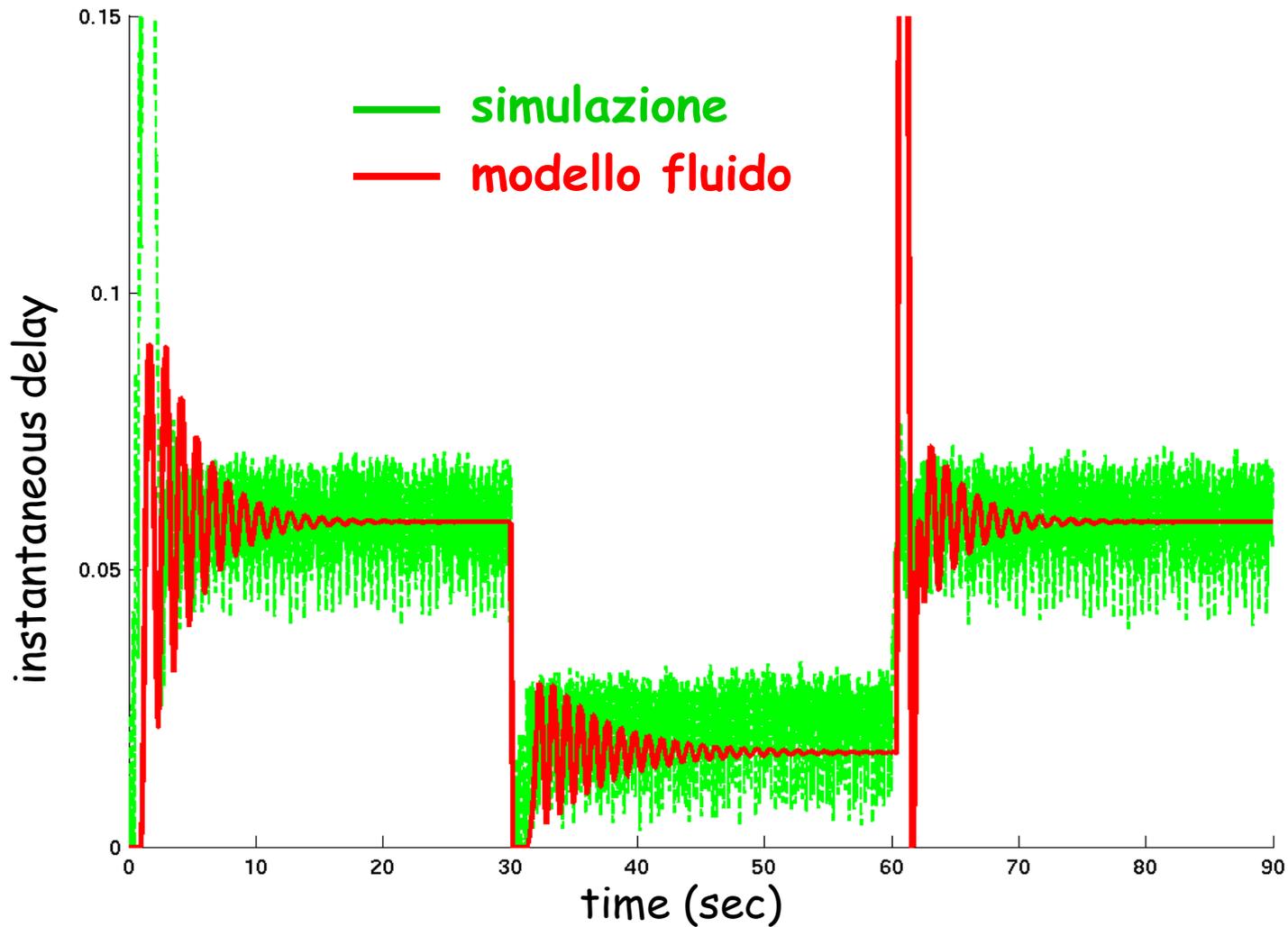
Accuratezza del modello?

- ❑ canali OC-12 - OC-48
- ❑ RED con ritardo target di 5 msec
- ❑ 2600 flussi TCP

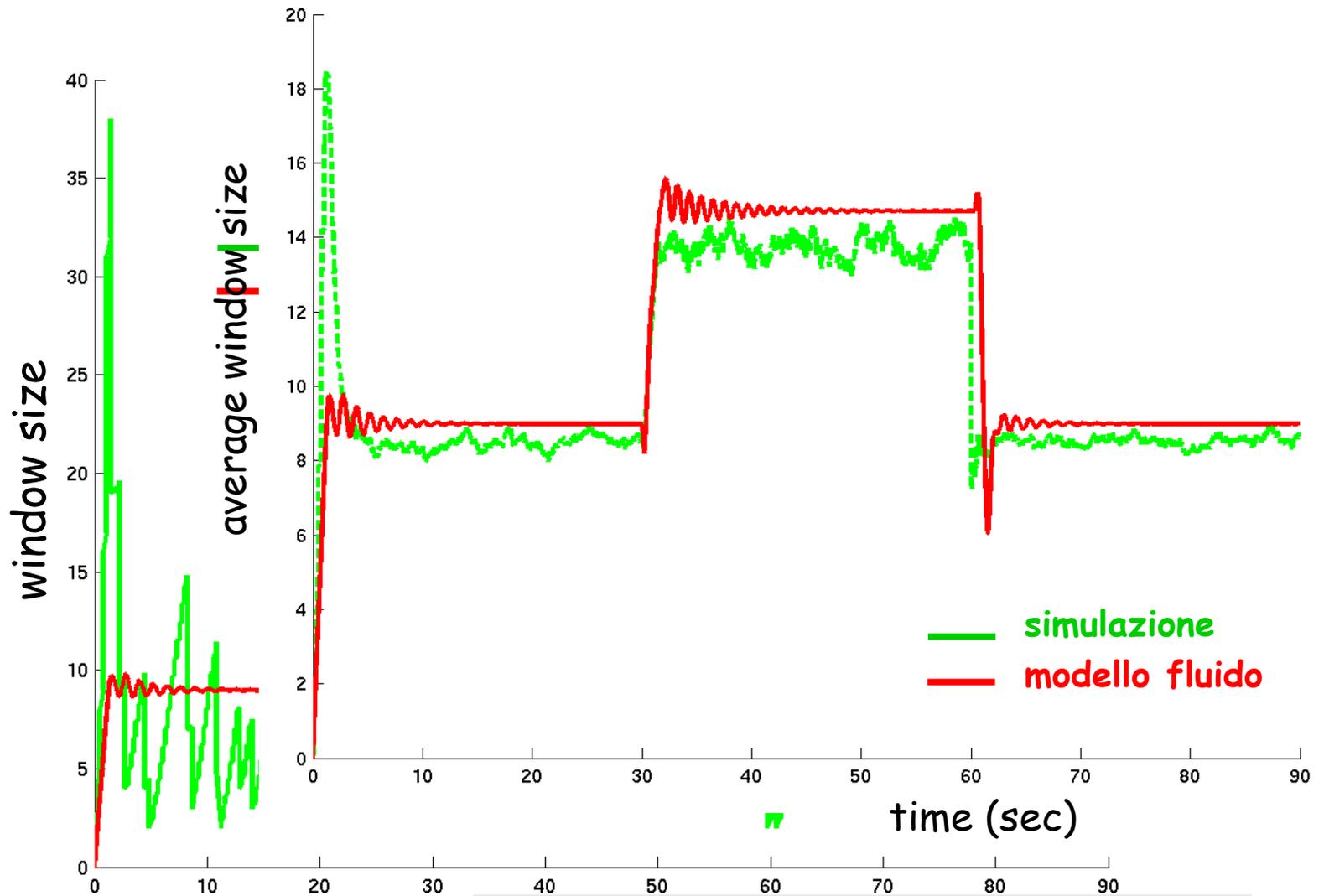


- ❑ diminuiscono a 1300 al tempo $t=30$ sec
- ❑ tornano a 2600 al tempo $t=90$ sec



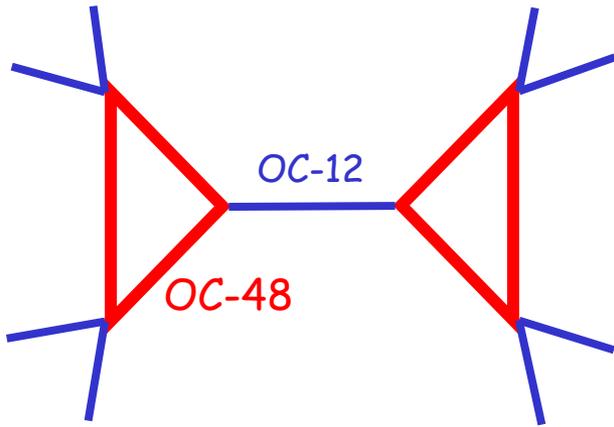


Buon accordo sulla lunghezza delle code

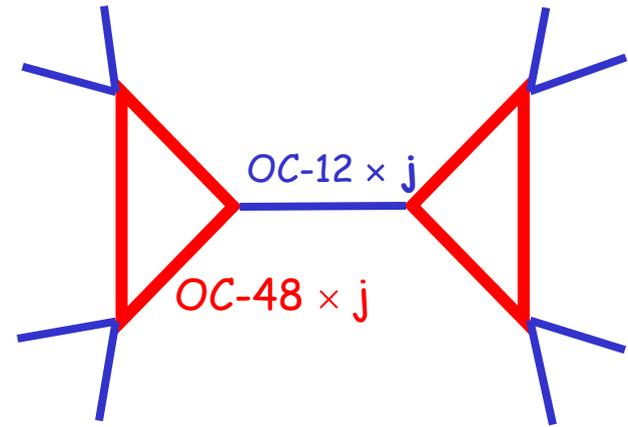


modello cattura bene dimensione
media della finestra di una
popolazione omogenea di flussi ⁷⁻⁵⁵

Proprietà di scalamento



$t=30$ $t=90$



$t=30$ $t=90$

$$W_k(t) = W^j_k(t)$$

$$q_v(t) = q^j_v(t)/j$$

Sommario: flussi TCP modellati come fluidi

Cosa abbiamo visto?

- connessioni TCP viste come flussi fluidi a tasso variabile
- tasso di emissione dipende da feedback ricevuto dalla rete:
 - capacità $C = \langle C_v \rangle$
 - lung. medie code $x = \langle x_v \rangle$
 - prob. perdita $p = \langle p_v(x_v) \rangle$
- comportamento transitorio modellato attraverso sistema di equazioni differenziali ordinarie

possibilità di predire il comportamento di (molti) flussi TCP usando modelli fluidi

Altre estensioni e applicazioni

□ code drop-tail

- occorre usare modello a coda (basato su teoria delle code)

- semplice modello $M/M/1/K$ sufficiente in regime di carico elevato
- in regime di carico medio/basso occorre tenere conto della burstiness del traffico TCP
- importante distinguere tra probabilità di perdita media e probabilità di perdita di un burst

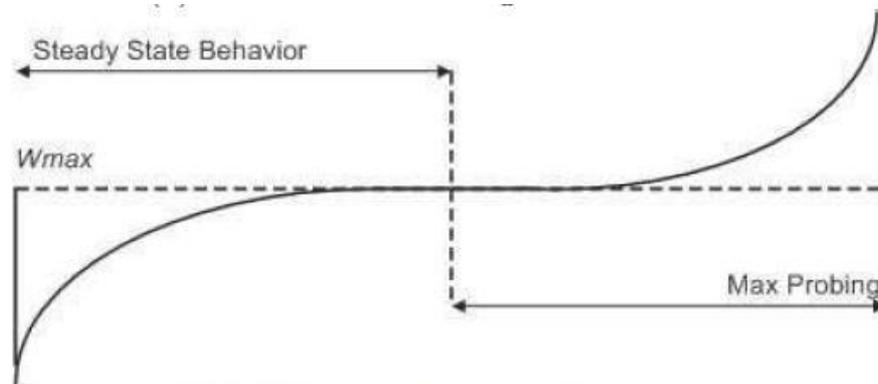
□ ottimizzazione/stabilità di RED

□ studio di altri algoritmi di controllo della congestione

- usando teoria dei controlli

TCP CUBIC

- ❑ implementazione di default in Linux (kernels ≥ 2.6)
- ❑ $W = C(t - K)^3 + W_{max}$
- ❑ t : tempo misurato dall'ultima perdita
- ❑ $K = \sqrt[3]{\frac{W_{max}\beta}{C}}$
- ❑ C : parametro
- ❑ β : fattore di decrescita moltiplicativo



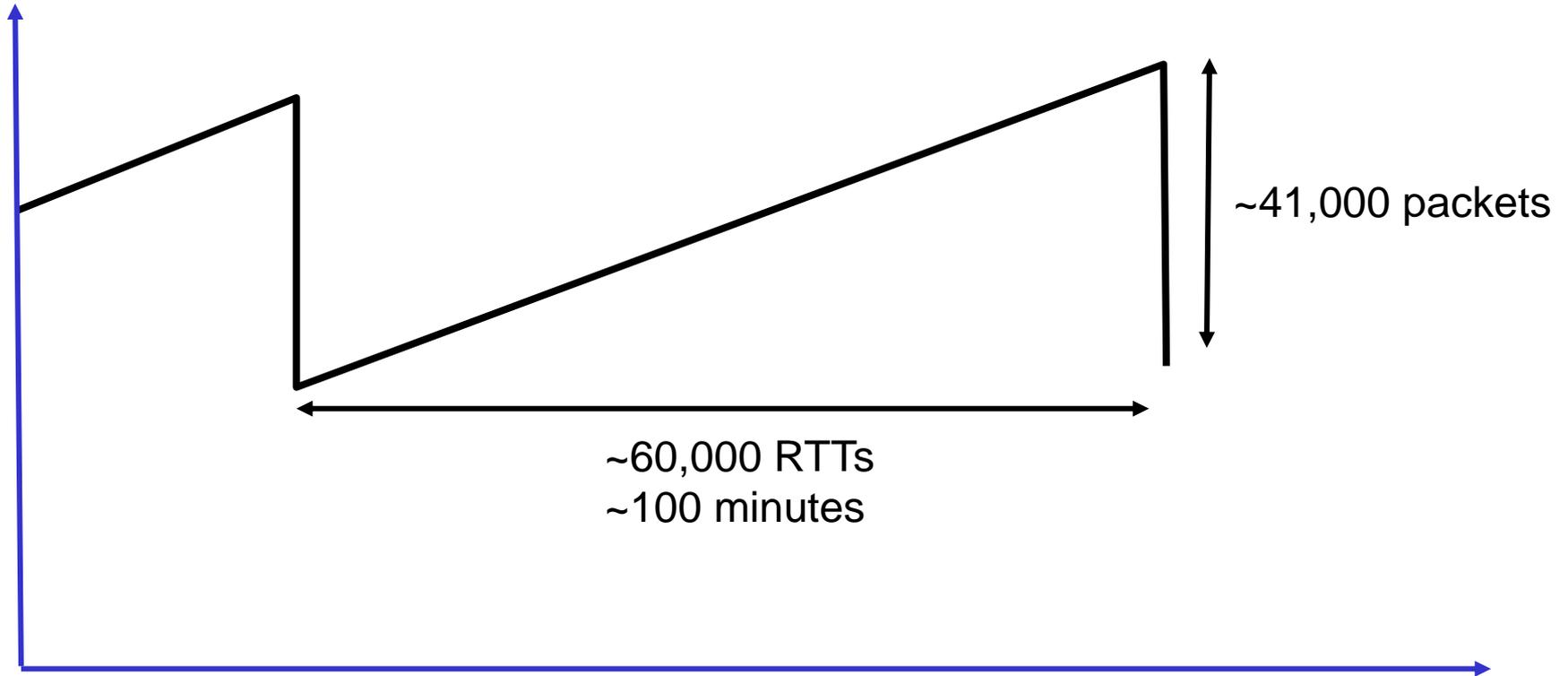
(b) CUBIC window growth function.

Scenario per alte prestazioni

Consideriamo una singola connessione TCP standard con:

- ❑ pacchetti da 1500-byte;
- ❑ 100 ms round-trip time;
- ❑ throughput atteso di 10 Gbps;
- ❑ richiede una finestra media di congestione di 83,333 segmenti!
- ❑ al più una perdita (o mark) ogni 5,000,000,000 pacchetti, o, equivalentemente, ogni 1 e 2/3 ore!

Comportamento TCP standard



□ Tempo T tra due segnali di congestione:

$$B = \frac{c}{R \sqrt{p}} \quad \longrightarrow \quad T = \frac{B R^2}{c^2} \quad \text{dinamiche troppo lente}$$

p - drop probability

$$B = \frac{c}{R\sqrt{p}} \rightarrow \frac{1}{p} = \frac{B^2 R^2}{c^2}$$

Q - given B , what is tie between drops.

$T = ?$



$$T = \frac{1}{\text{loss rate}} = \frac{1}{B} \times \frac{1}{p} = \frac{1}{B} \times \frac{B^2 R^2}{c^2} = \frac{BR^2}{c^2}$$

$$\text{loss rate} = B \times p$$

Possibili soluzioni al problema

In pratica, utenti fanno:

- apertura di N connessioni TCP in parallelo
 - richiede middleware di divisione/riassemblaggio dei dati
- uso di MulTCP (singola connessione che ottiene comportamento equivalente a N connessioni TCP virtuali)
 - decrescita di $b \cdot W$, con $b < \frac{1}{2}$ (conti prox slide)

Si può fare di meglio?

N connections

$$\beta = \frac{Nc}{R\sqrt{p}} \quad ; \quad p - \text{loss prob.}$$

MultiTCP - increase rule $\frac{1}{2}w$, decrease rule

@ loss - decrease by $\frac{w}{2M}$

$$\begin{aligned} \frac{dw}{dt} &= \frac{w}{R} \left(\frac{(1-p)}{w} - \frac{w}{2M} p \right) \\ &= \frac{(1-p)}{R} - \frac{w^2}{2MR} p \end{aligned}$$

$$0 = \frac{(1-p)}{R} - \frac{w^2}{2MR} p \rightarrow 0 = (1-p) - \frac{w^2}{2M} p$$

$$\beta \approx \frac{\sqrt{2M}}{R\sqrt{p}} \cdot \text{set } M = N^2$$

$$w = \frac{\sqrt{2M(1-p)}}{\sqrt{p}} \approx \frac{\sqrt{2M}}{\sqrt{p}}$$

Spettro delle possibili soluzioni

Da un lato dello spettro:

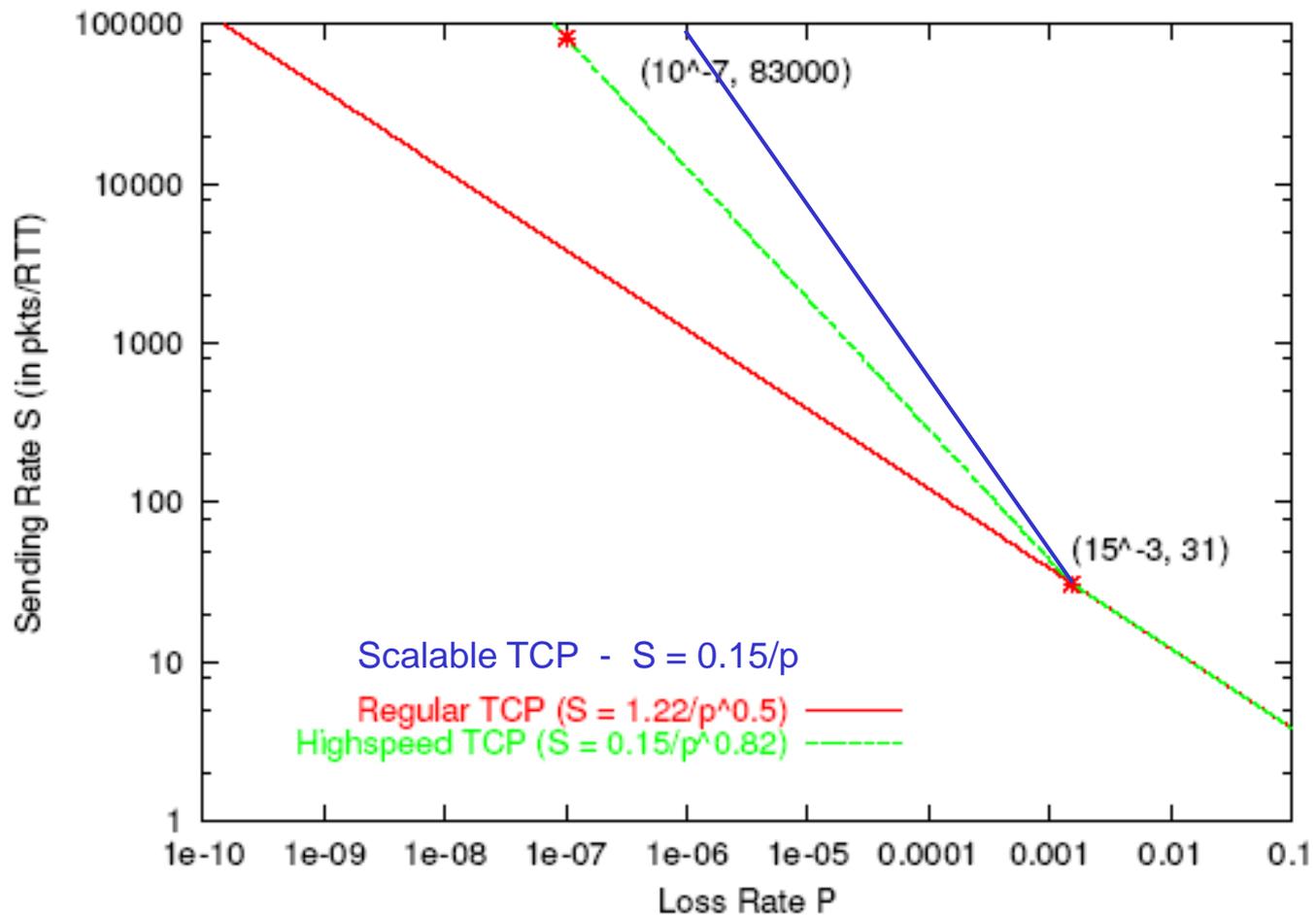
- cambiamenti semplici, incrementali, e facilmente implementabili a protocolli esistenti
 - HighSpeed TCP (TCP con parametri modificati);
 - QuickStart (opzione IP per permettere grandi valori di finestra di congestione iniziale)

Dall'altro lato dello spettro:

- nuovi protocolli di trasporto, eventualmente con feedback esplicito dai router

High speed TCP

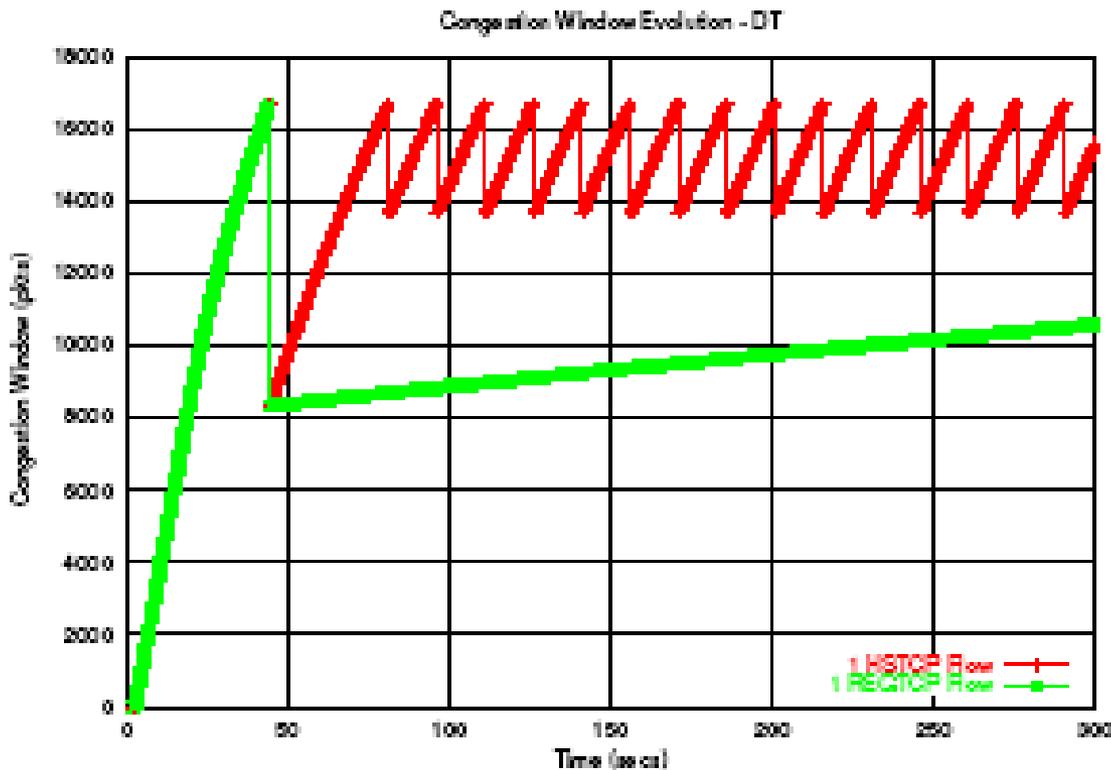
HighSpeed TCP: the modified response function.



High speed TCP (RFC 3649)

- additive increase, multiplicative decrease
- ma con incrementi-decrementi che dipendono dalla 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

$$W \leftarrow W + a \quad \text{per ACK}$$

$$W \leftarrow W - b W \quad \text{per finestra in cui si rileva perdita}$$

- Tempo T tra due segnali di congestione:

$$B = \frac{a}{R b p} \quad \longrightarrow \quad T = \frac{b R}{a}$$

dinamiche
indipendenti
da B !

for STCP, what is β ?

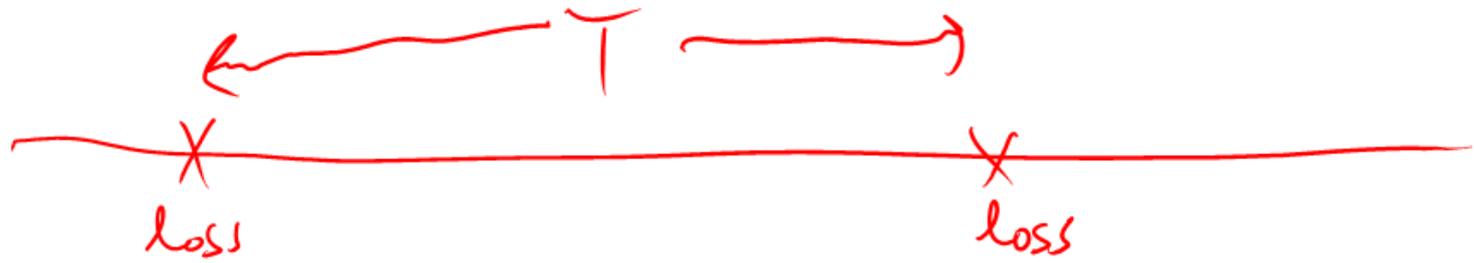
$$\frac{dW}{dt} = \frac{W}{R} \left((1-p)a - bWp \right)$$

$$= (1-p)a \frac{W}{R} - \frac{bW^2 p}{R}$$

$$0 = (1-p)a \frac{W}{R} - \frac{bW^2 p}{R} = (1-p)a - bWp$$

$$\Rightarrow W = \frac{(1-p)a}{bp}$$

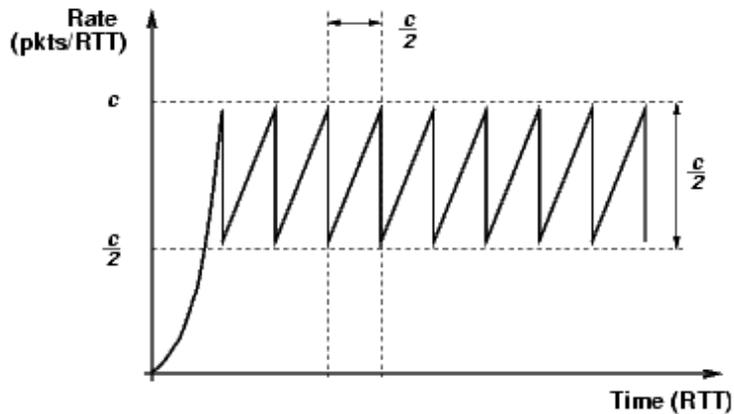
$$\beta = \frac{W}{R} = \frac{(1-p)a}{bRp} \approx \frac{a}{bRp} \quad p \rightarrow 0$$



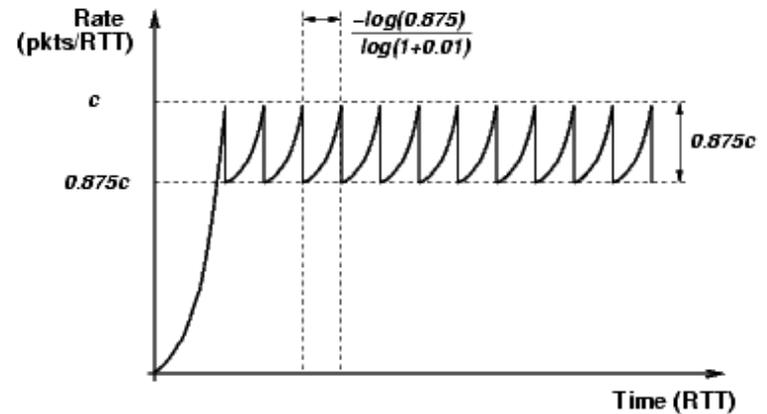
$$T = \frac{1}{B} \times \frac{1}{P} = \frac{1}{B} \times \frac{BbR}{a} = \frac{bR}{a}$$

$$B = \frac{a}{bRp} \rightarrow \frac{1}{P} = \frac{BbR}{a}$$

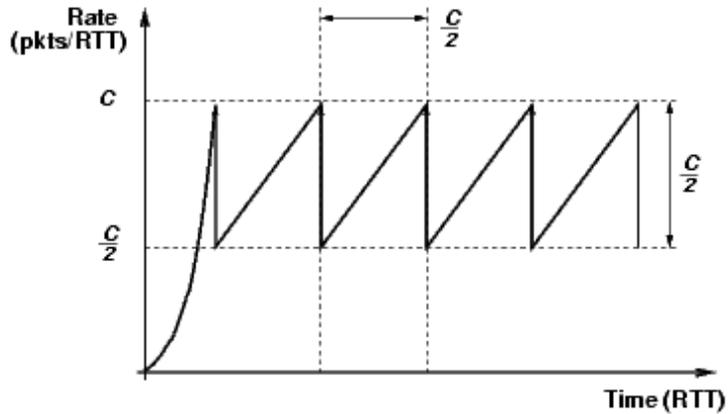
STCP in immagini



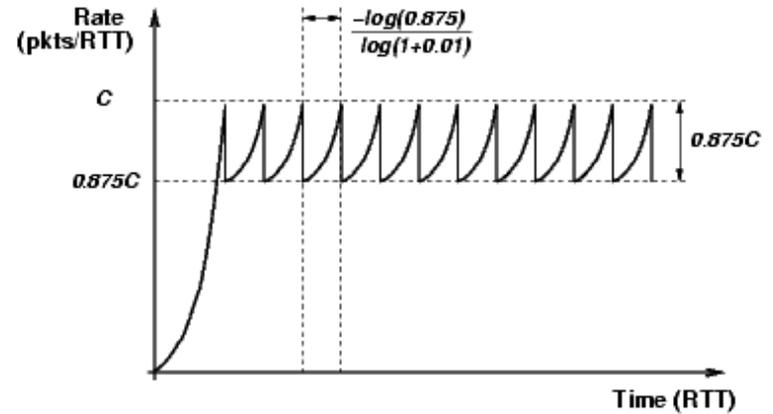
Traditional TCP with small capacity



Scalable TCP with small capacity



Traditional TCP with large capacity



Scalable TCP with large capacity

Osservazioni

- High-performance TCP è ancora problema aperto
- esistono anche proposte di protocolli di congestione delay-based (TCP FAST, TCP VEGAS, COMPOUND TCP)
- soluzioni con supporto dei router (XCP)