



Valutazione del TCP con NS2

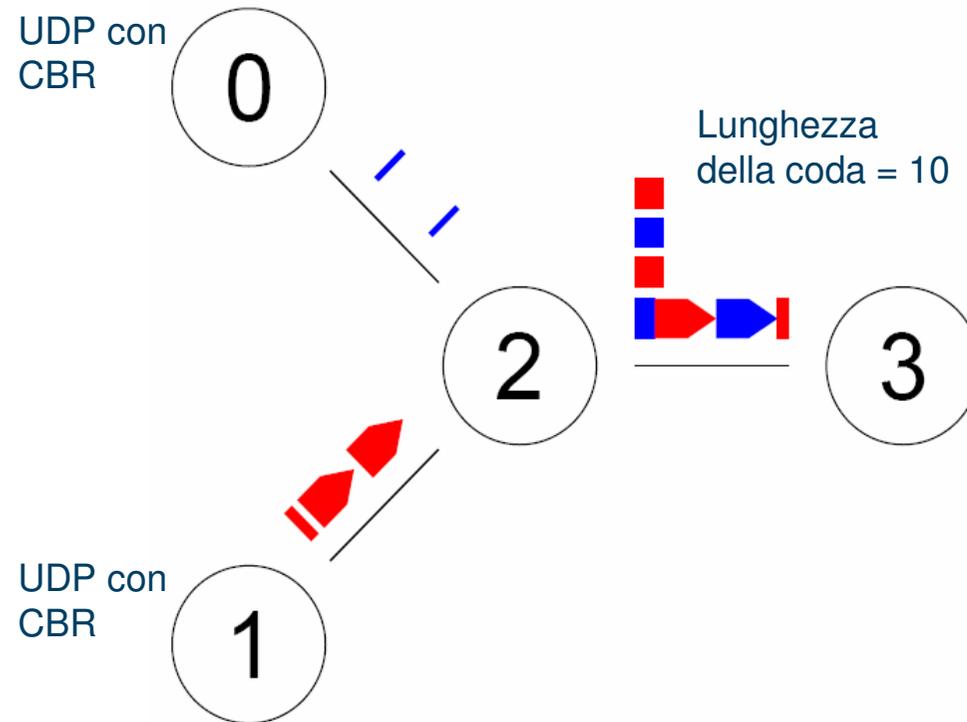
Gaia Maselli

maselli@di.uniroma1.it



Esempio di simulazione

- example2.tcl



- Cosa succede se il traffico tra il nodo 0 e 3 è di tipo TCP?



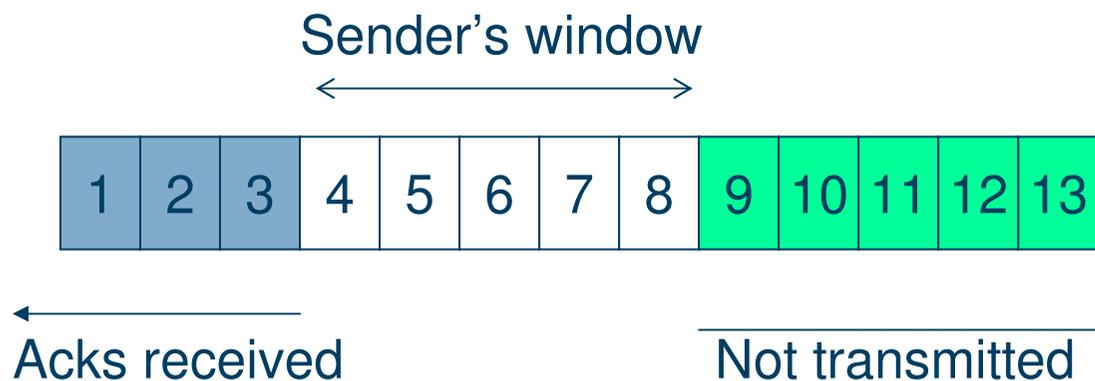
Articoli di riferimento

- K. Fall, S. Floyd, "Simulation-based comparisons of Tahoe, Reno and SACK TCP", in *SIGCOMM Comput. Commun. Rev.* 26, 3 (Jul. 1996), 5-21.
- G. Holland, N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks", in *Wireless Networks* 8, pp. 275-288, 2002, Kluwer.
- A. Al Hanbali, E. Altman, P. Nain, "A survey of TCP over Ad Hoc Networks", *IEEE Communications Surveys & Tutorials*, Vol.7 No.3, P. 22-36, Aug. 2005.
- Haejung Lim; Kaixin Xu; Gerla, M., "TCP performance over multipath routing in mobile ad hoc networks", in *IEEE International Conference on Communications, 2003. ICC '03. Volume 2*, 11-15 May 2003 Page(s):1064 - 1068 vol.2



Window based flow control

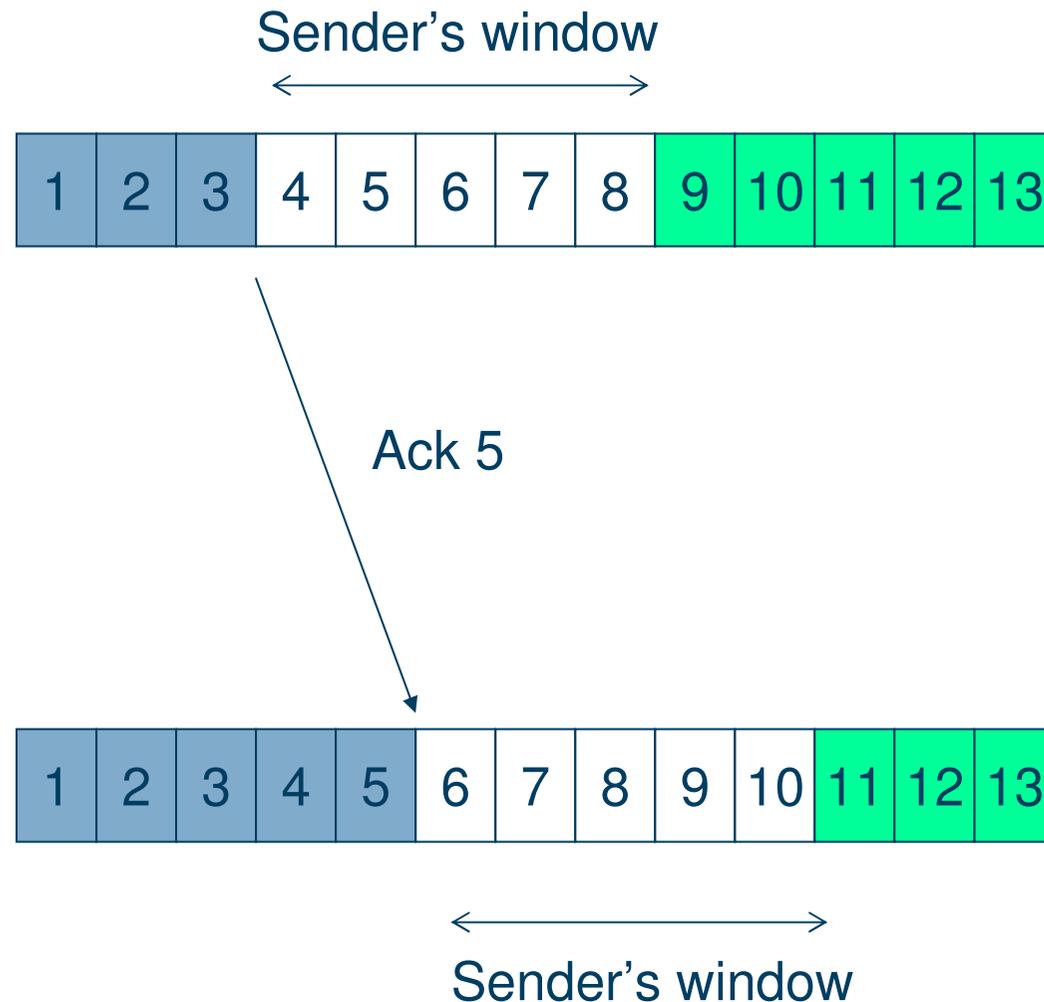
- Protocollo basato su *sliding window*
- L'ampiezza della finestra è data dal valore minimo tra
 - **receiver's advertised window** – determinata dal receiver in base allo spazio disponibile nel buffer
 - **congestion window** – determinata dal sender, in base al feedback dalla rete





Window based flow control

- TCP window flow control è "self-clocking"
- Nuovi dati vengono inviati quando quelli inviati sono stati ricevuti (ack)
- Mantiene "equilibrio"



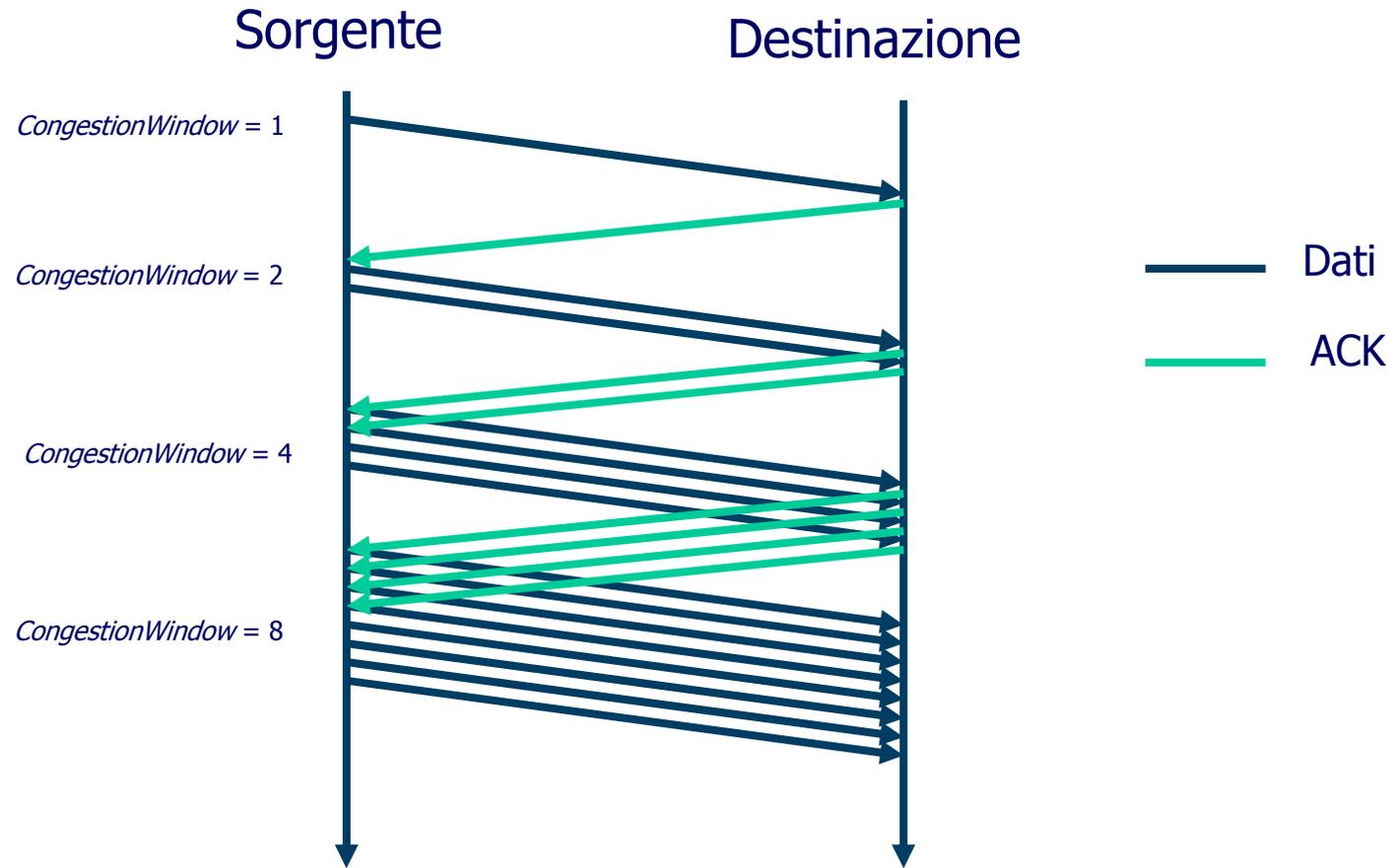


Meccanismi del TCP

- Slow start
- Congestion avoidance
- Fast retransmit
- Fast recovery



Slow start

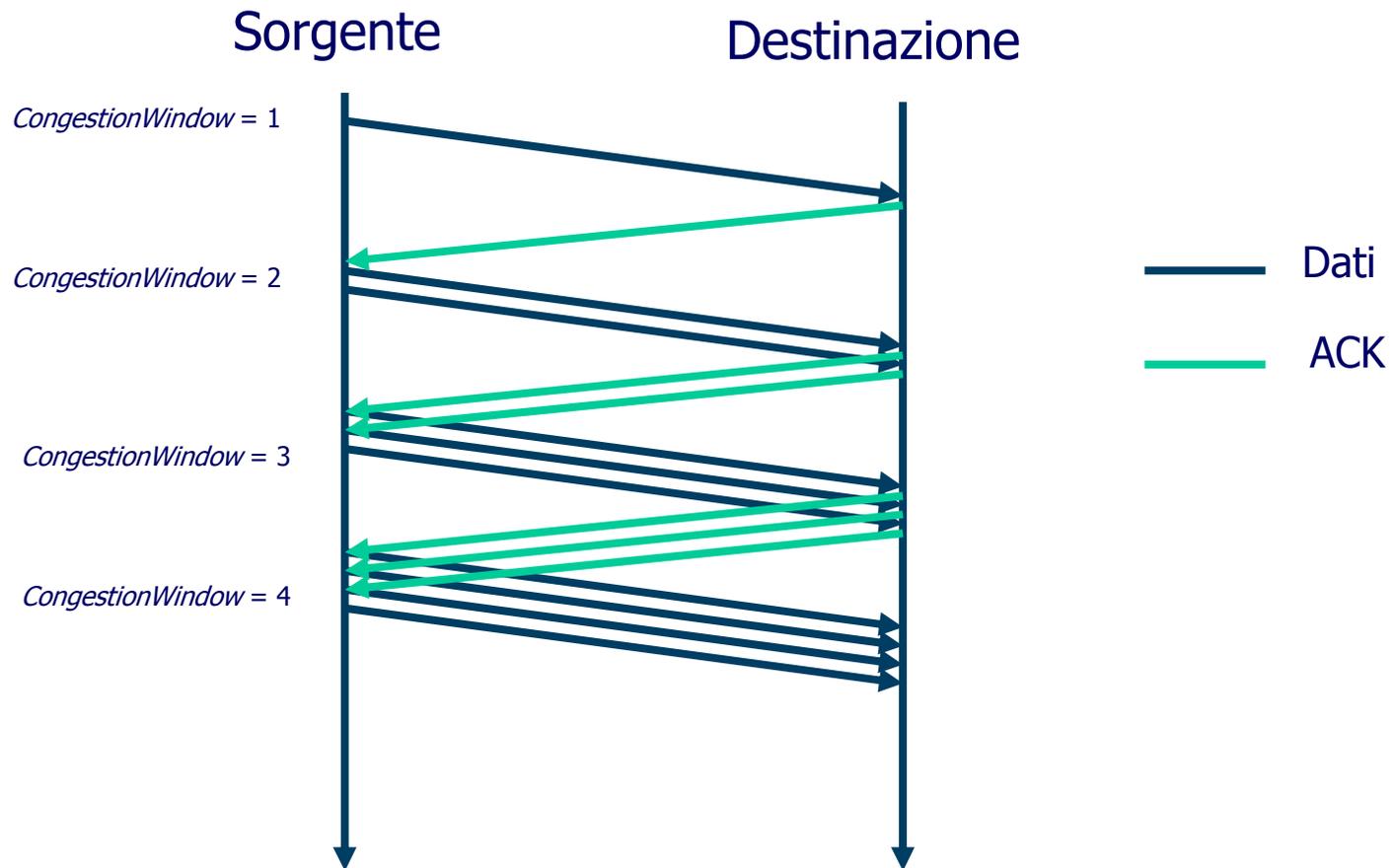


CongestionWindow = CongestionWindow + 1
All'arrivo di ogni ACK



Congestion avoidance

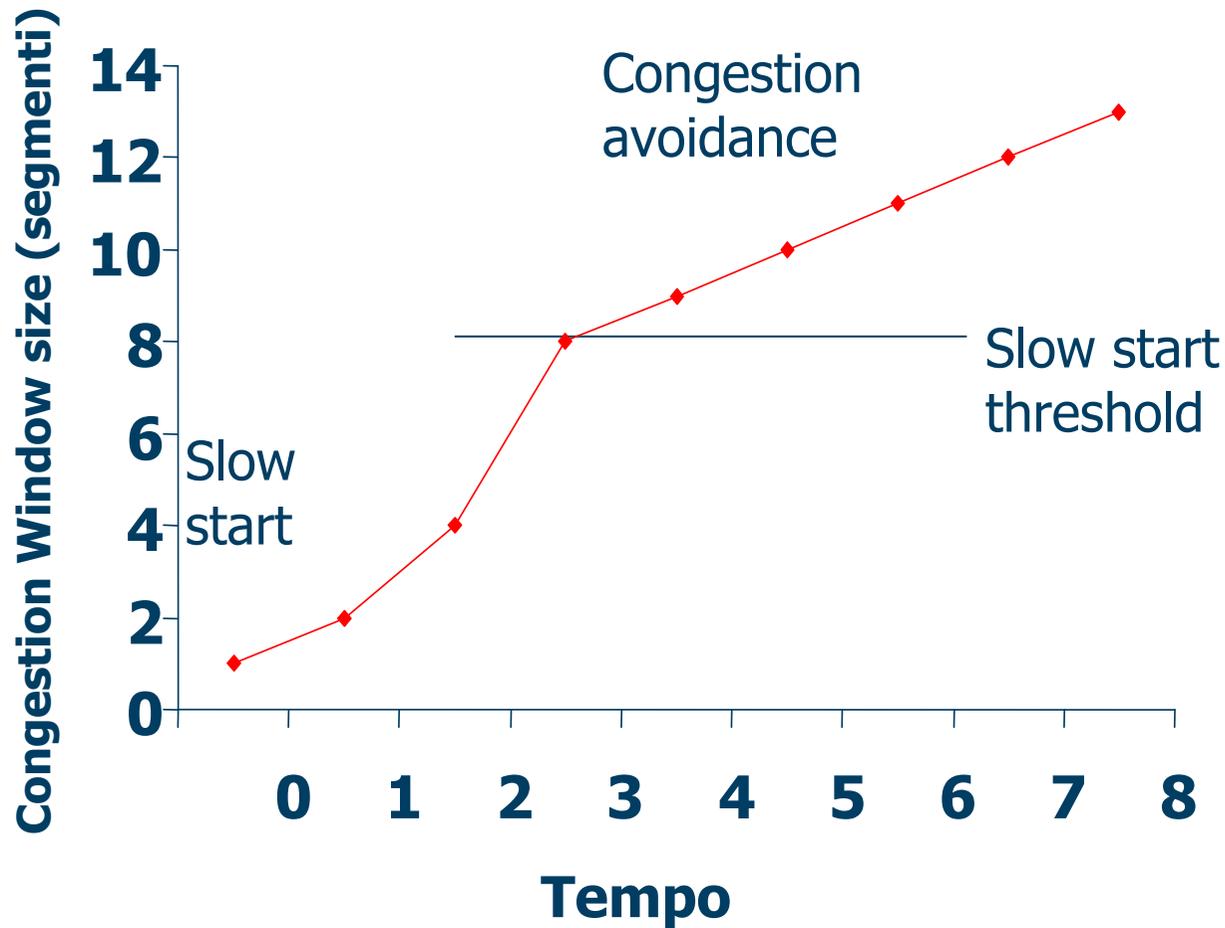
ADDITIVE INCREASE



Incremento lineare



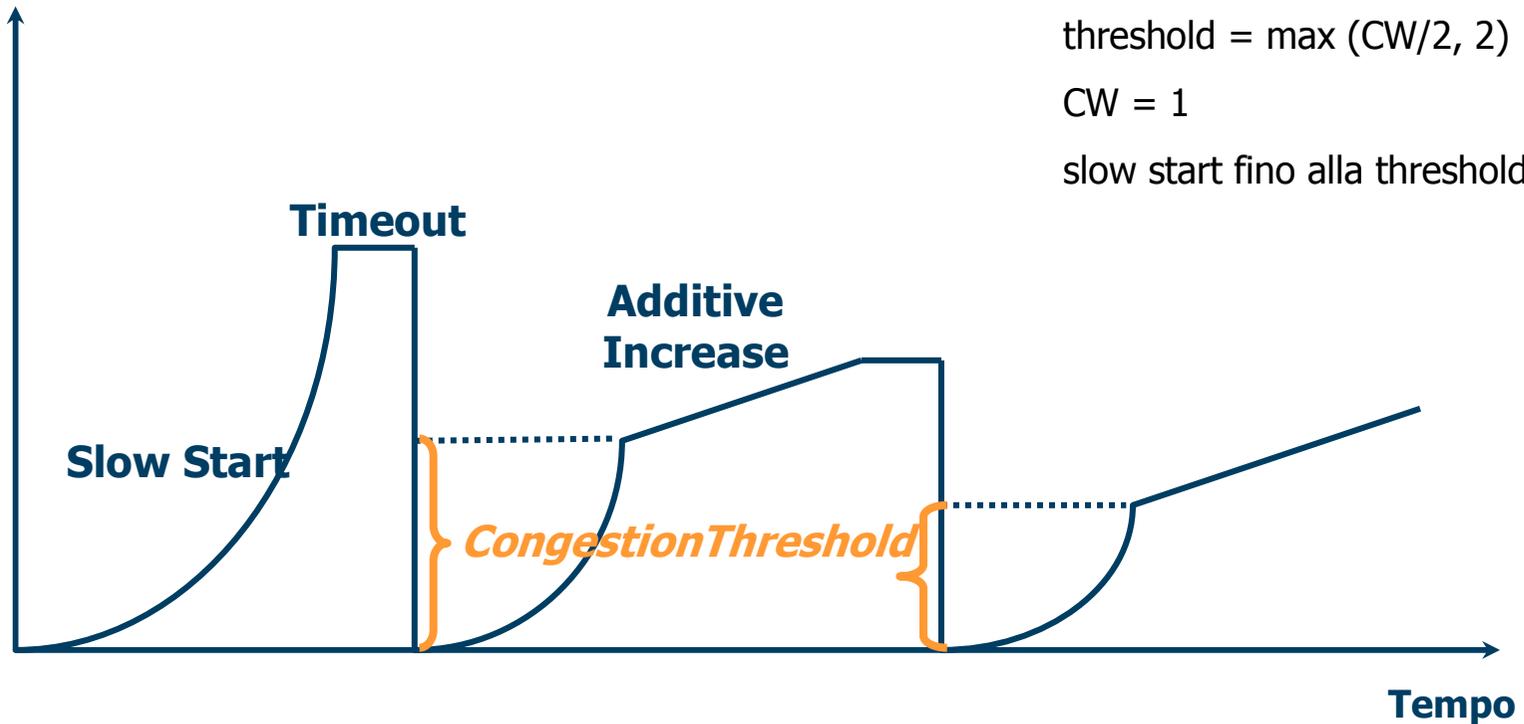
Slow start + Congestion Avoidance





Congestion control - timeout

Congestion Window



Additive increase multiplicative decrease

threshold = $\max(CW/2, 2)$

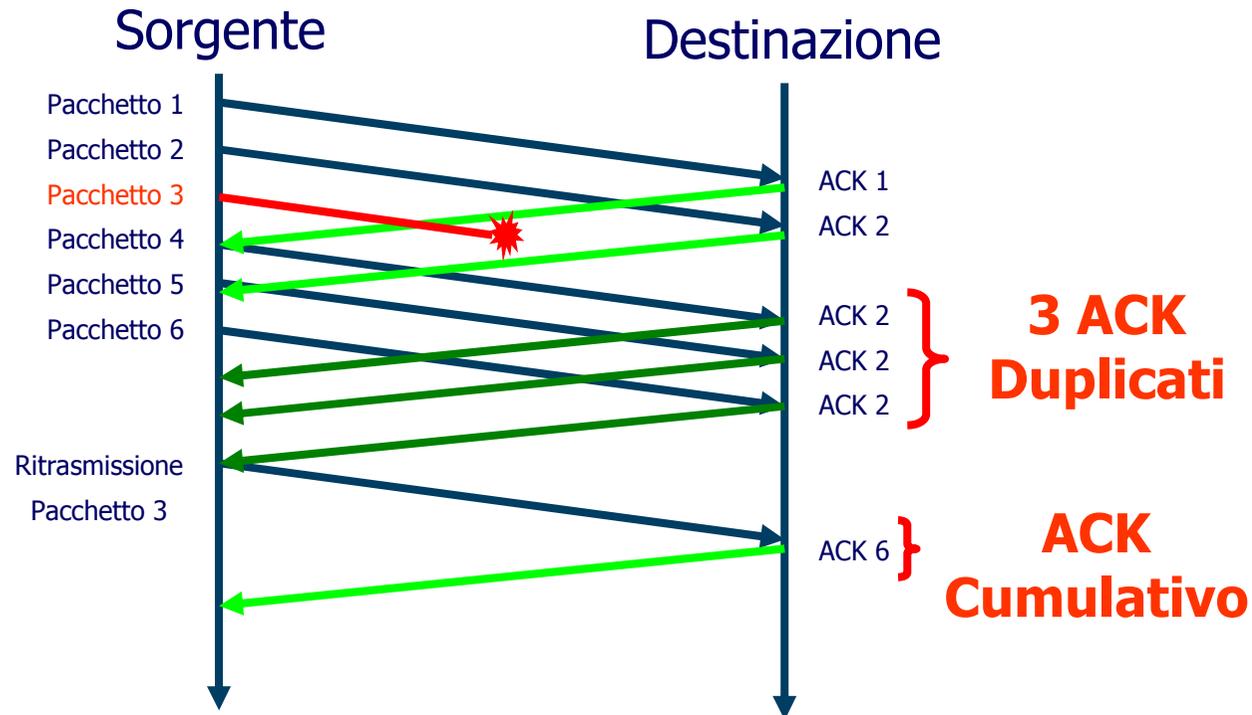
CW = 1

slow start fino alla threshold

$$CongestionThreshold = \text{Max}(CongestionWindow/2, 2)$$



Fast retransmit (ack duplicati)



- Fast retransmit non aspetta lo scadere del timeout
 - Attivo quando la sorgente riceve multipli (≥ 3) ack duplicati (prima del timeout)



Timeout VS Fast retransmit

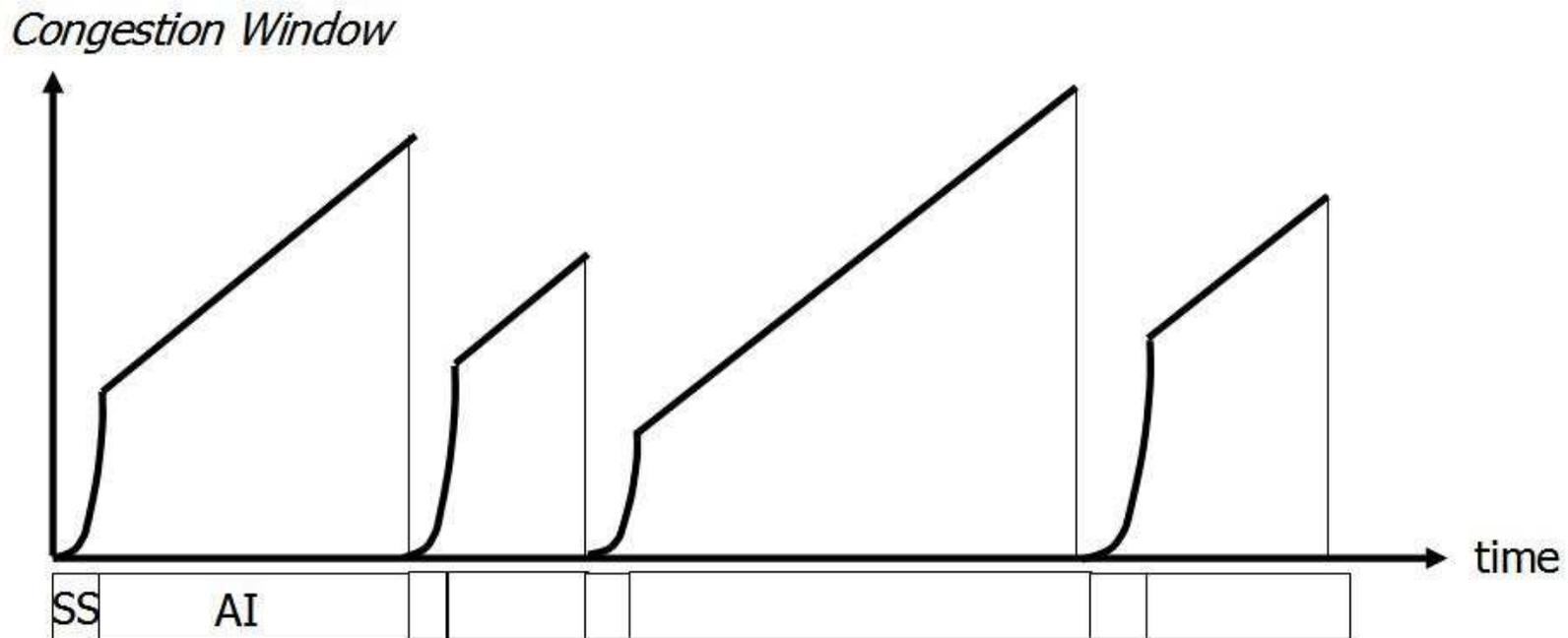
Differente dal timeout : slow start segue timeout

- timeout -> pacchetti non arrivano più a destinazione
 - fast retransmit -> un pacchetto viene perso, ma i successivi arrivano a destinazione
 - Non c'è necessità di eseguire slow start, perchè si è persa una piccola quantità di pacchetti
-
- Fast retransmit in genere è seguita dalla fase di **Fast recovery**



Andamento temporale della CW

Slow Start, Additive Increase, Fast Retransmit



SS: Slow Start
AI: Additive Increase

In caso di perdita di un segmento

- $ssthresh = \max(CWND/2, 2)$
- $CW = 1$



Fast Recovery

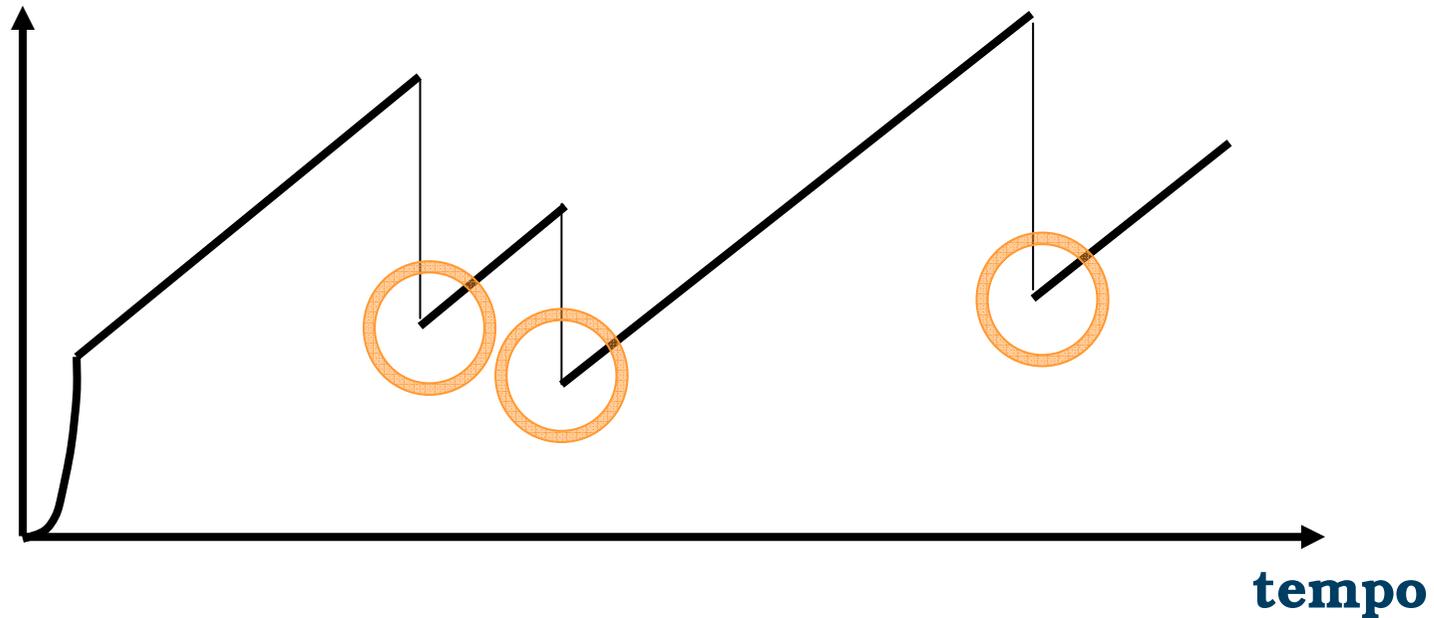
- $ssthresh = cwnd/2$
- $cwnd = ssthresh + \text{number of dupacks}$
- Ritrasmette il pacchetto perso (fast retransmit) e attende l'ack
- Quando riceve un nuovo ack (non di un pacchetto ritrasmesso) :
 - $cwnd = ssthresh$
 - esegue congestion avoidance

Congestion window dimezzata



Fast recovery

Congestion Window



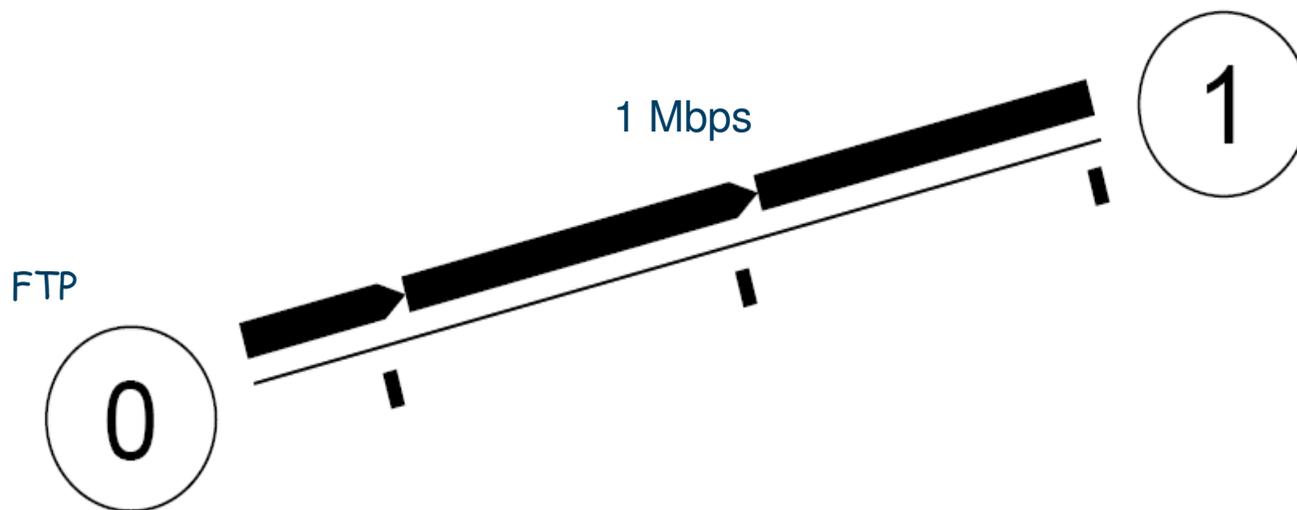
Stato di Congestione Leggera:

la rete è congestionata, ma alcuni pacchetti arrivano a destinazione. Con 3 DUPACK viene rilevata la perdita del pacchetto, si effettua il **Fast Retransmit** e si entra nella fase di **Fast Recovery**.



Esempio: scenario simulato

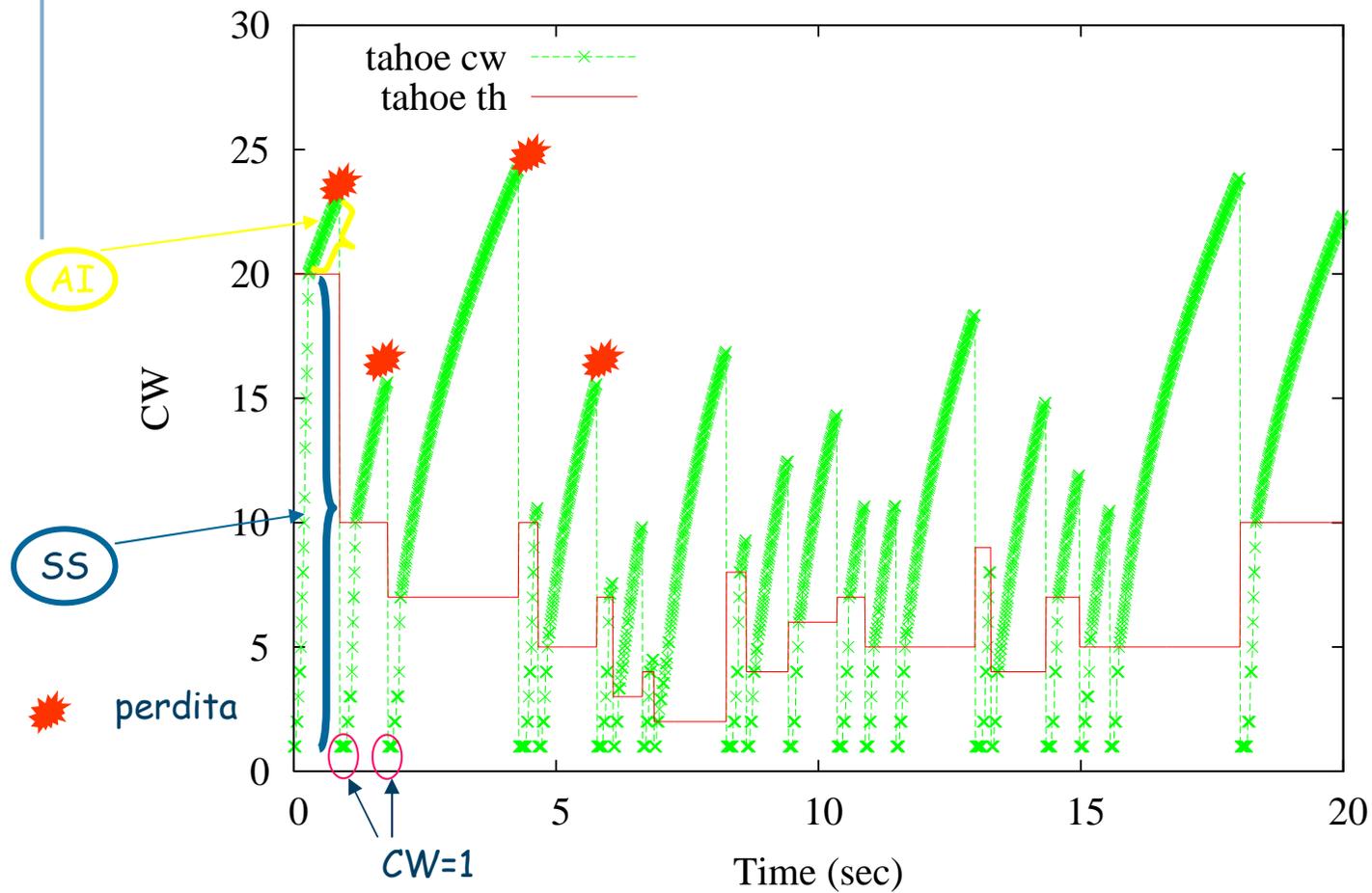
- 2 nodi, una connessione TCP 0 -> 1, perdita dei pacchetti randomica (packet error rate = 1%)





TCP Tahoe

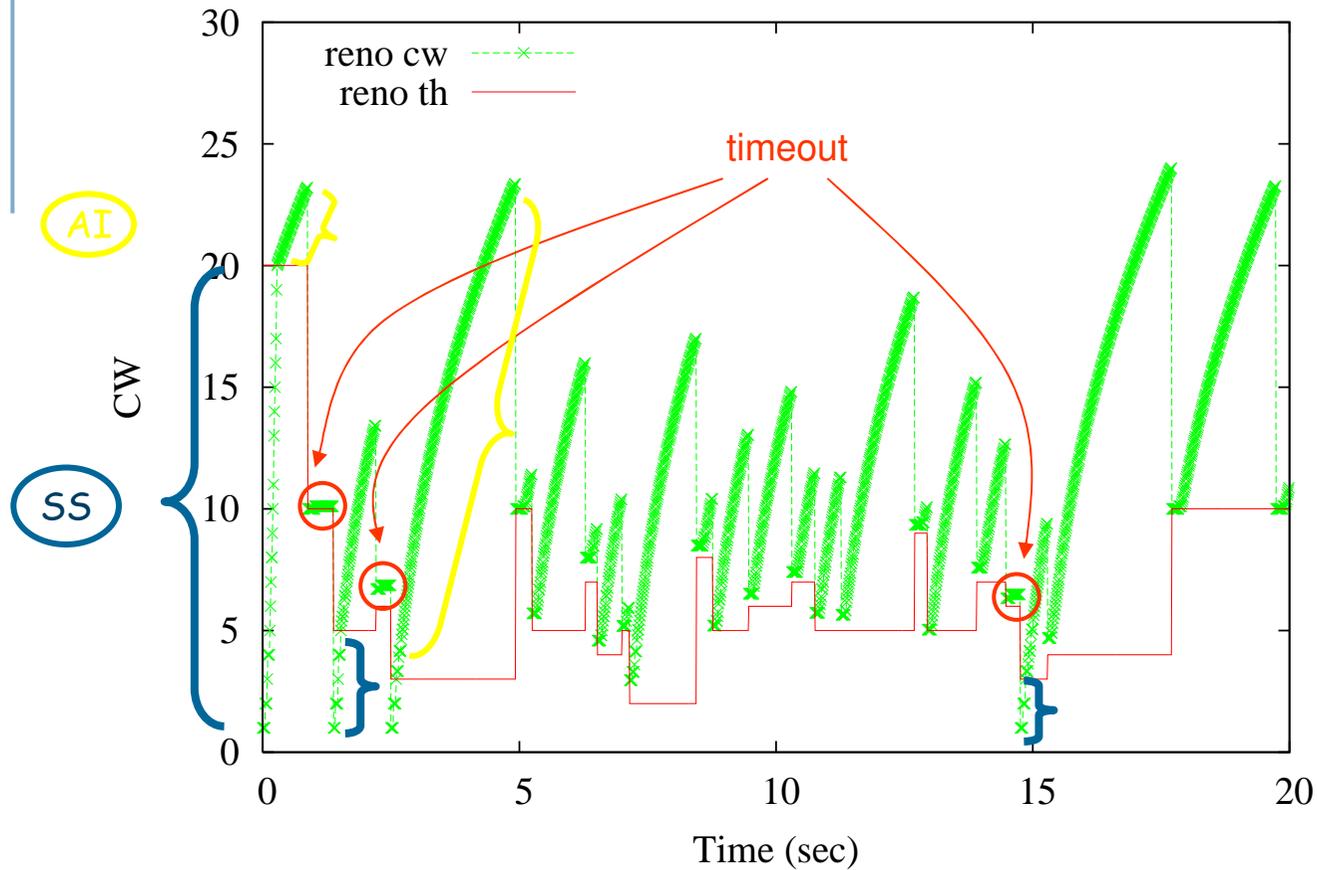
- SS + CA + fast rtx





TCP Reno

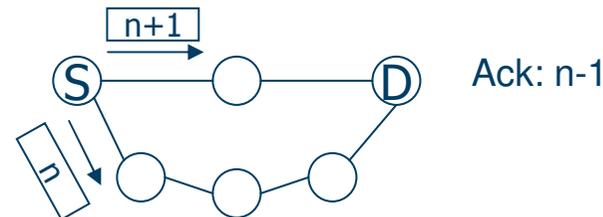
- SS + CA + fast rtx + fast recovery





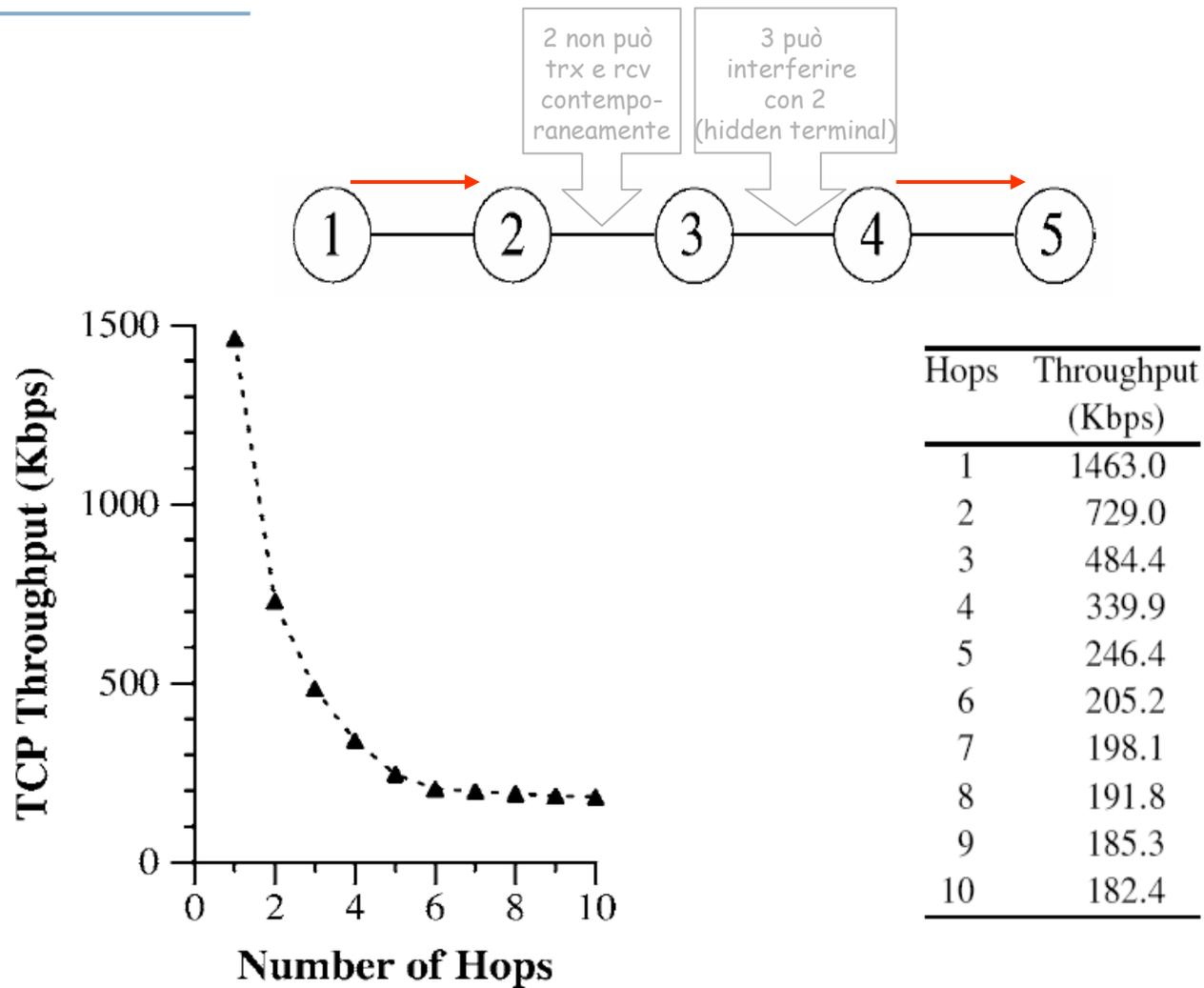
TCP in ambiente wireless ad hoc

- TCP esegue controllo della congestione in caso di perdita di pacchetti
- Nelle reti wireless i pacchetti possono andare persi **anche** a causa di
 - Errori nei link (es. multipath fading)
 - Mobilità -> interruzione delle rotte
- Queste perdite vengono **erroneamente** interpretate come effetti della congestione
 - Packet's timeout (or DupACKS) → riduzione della congestion window
- Molte soluzioni ai problemi di performance delle reti ad hoc si basano sull'utilizzo rotte multiple (multipath routing) per aggirare eventuali percorsi interrotti, nodi sovraccarichi, link inefficienti, etc.
- Multipath drawbacks
 - Stima inaccurata del RTT
 - Frequenti ACK duplicati
- Il protocollo di MAC limita l'utilizzo della capacità della rete, riducendo le performance del trasferimento dei dati





Effetto di 802.11 sul throughput TCP*

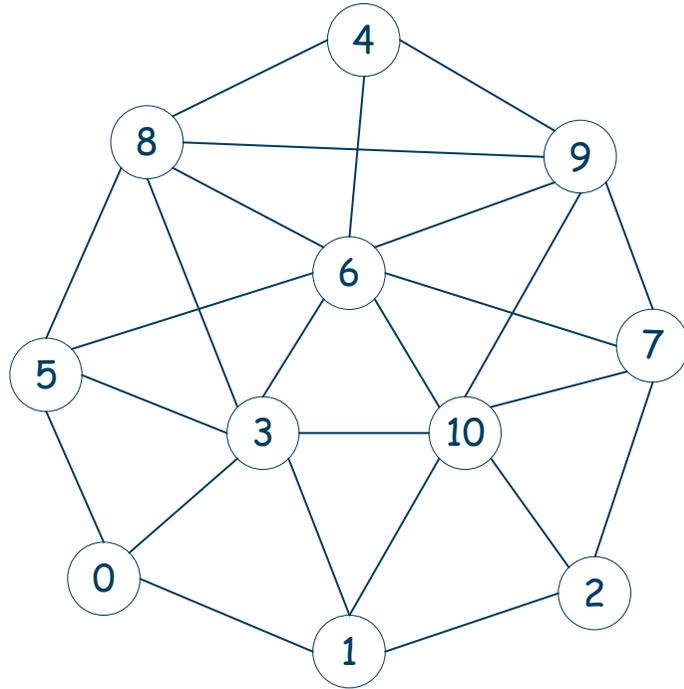


TCP-Reno throughput over an 802.11 fixed, linear, multi-hop network of varying length (in hops).

*[Holland, Vaidya]



Routing multipath



Active TCP connections
6 → 2
1 → 6
0 → 7
2 → 9

-- NODE 0 Routing Table at time t --

```

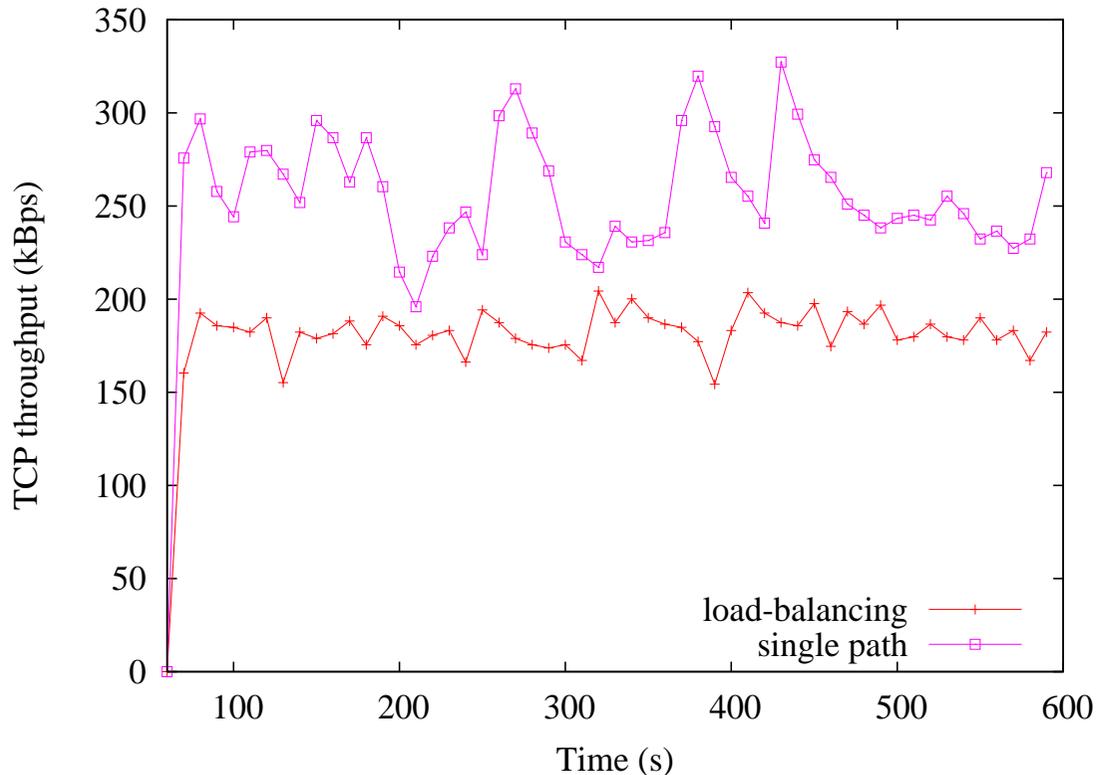
-----
R_dest=1 R_next=1 R_dist=1
R_dest=3 R_next=3 R_dist=1
R_dest=5 R_next=5 R_dist=1
R_dest=6 R_next=5 R_dist=2
R_dest=6 R_next=3 R_dist=2
R_dest=8 R_next=5 R_dist=2
R_dest=8 R_next=3 R_dist=2
R_dest=10 R_next=3 R_dist=2
R_dest=10 R_next=1 R_dist=2
R_dest=2 R_next=1 R_dist=2
R_dest=7 R_next=1 R_dist=3
R_dest=7 R_next=3 R_dist=3
R_dest=7 R_next=5 R_dist=3
R_dest=9 R_next=1 R_dist=3
R_dest=9 R_next=3 R_dist=3
R_dest=9 R_next=5 R_dist=3
R_dest=2 R_next=3 R_dist=3
R_dest=4 R_next=3 R_dist=3
R_dest=4 R_next=5 R_dist=3
R_dest=10 R_next=5 R_dist=3
R_dest=2 R_next=5 R_dist=4
R_dest=4 R_next=1 R_dist=4
R_dest=6 R_next=1 R_dist=4
R_dest=8 R_next=1 R_dist=4
----- end routing table -----

```

OLSR
routing
protocol



Effetto del routing multipath



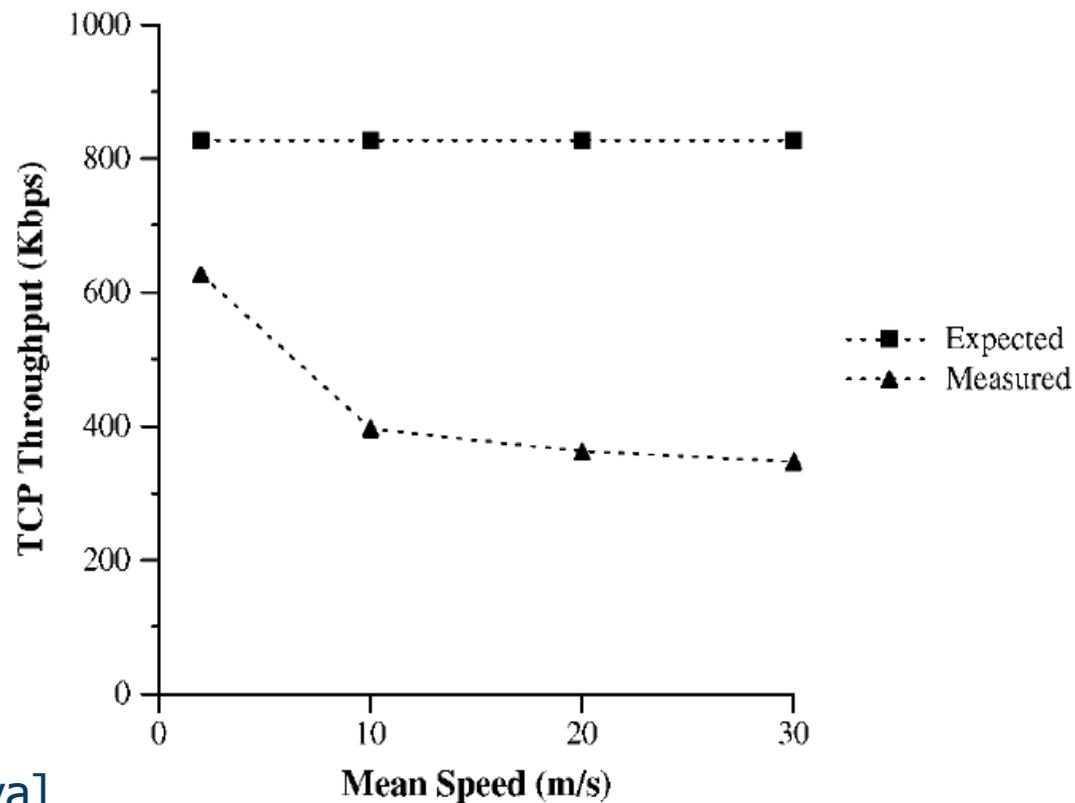
Throughput: quantità di dati correttamente ricevuti dalla destinazione nell'unità di tempo

- Load-balancing: usa rotte multiple verso la destinazione contemporaneamente
- + La probabilità che non ci sia una rotta tra sender e receiver è minima
- La stima del RTT può non essere accurata
- Pacchetti disordinati alla destinazione



Effetto della mobilità*

- Incrementando la velocità dei nodi le rotte si interrompono più facilmente, causando perdita di pacchetti, e necessità di ristabilirne di nuove (route discovery), con conseguente overhead e ritardo



*[Holland, Vaidya]



Alcune soluzioni

- Feedback esplicito
 - TCP Feedback (TCP-F)
 - Explicit Link Failure Notification (ELFN)
 - Il sender viene informato della route failure in modo che non attivi il controllo della congestione
 - Host unreachable (ICMP), or route failure (routing)
 - Congestion control viene disabilitato fino al ristabilimento della rotta
 - Le variabili del TCP vengono “congelate”
 - Messaggi di probe