



Politecnico di Milano

Facoltà di Ingegneria dell'Informazione

7 – TCP su wireless

Reti Mobili Distribuite

Prof. Antonio Capone

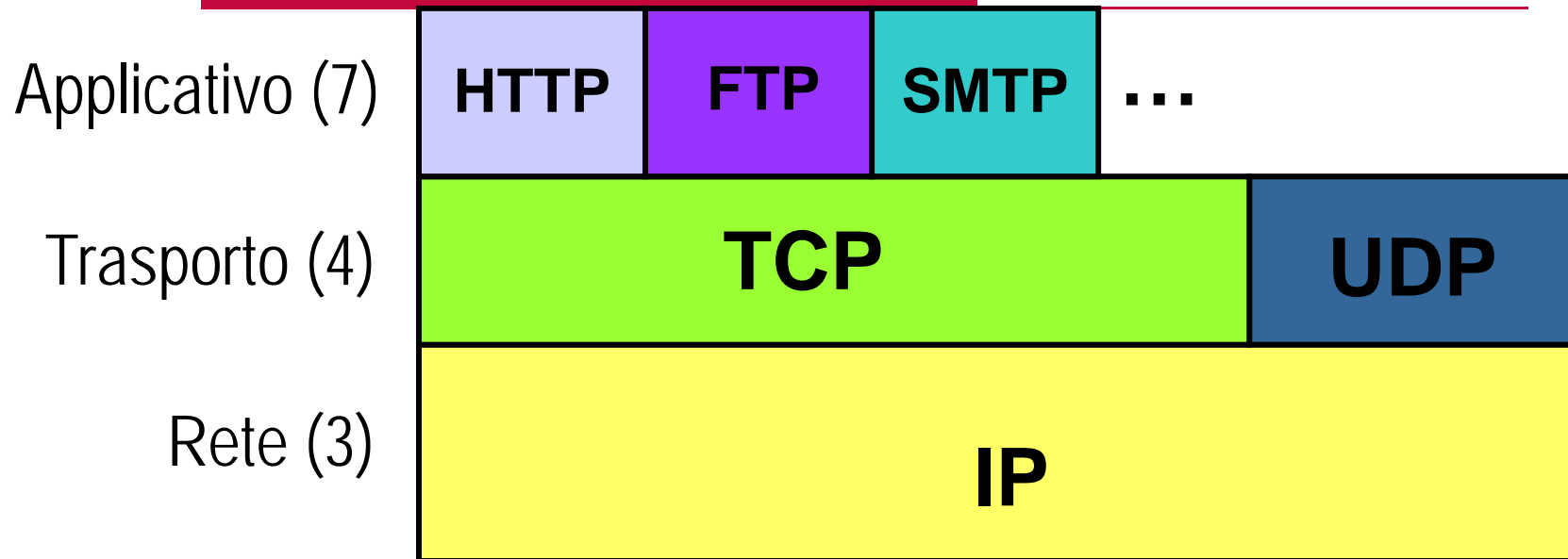


Sommario

- Il TCP in reti eterogenee cablate/radio
- Caratteristiche delle reti radio ed impatto sulle performance di TCP
- Tassonomia delle soluzioni proposte
 - Soluzioni Link-Layer
 - Indirect Protocols
 - Soluzioni End-to-end
- Modelli matematici per il calcolo di prestazioni di TCP su canali radio



Lo stack di Internet



- ❑ Il Transmission Control Protocol (TCP) è il protocollo di trasporto più utilizzato nella rete Internet
- ❑ Si stima che circa il 95% del traffico della rete Internet sia gestito da TCP
- ❑ Questo traffico è dovuto in larga parte a traffico Web, gestito dal protocollo HTTP (che utilizza TCP come protocollo di trasporto)



TCP su reti cablate

- Il TCP è stato progettato ed ottimizzato circa 20 anni fa per reti cablate, dimostrando di ottenere:
 - Efficienza nell'uso della banda disponibile
 - Equità nella condivisione delle risorse di rete con altre connessioni TCP
- Tuttavia, nel progettare i meccanismi di controllo del flusso e di recupero degli errori di TCP non si è tenuto conto delle caratteristiche di link wireless, introdotti e diffusi nella rete Internet in anni successivi
- Ad esempio, il TCP assume che tutte le perdite di pacchetto siano imputabili a congestione nella rete
- Se il TCP trasmittente sperimenta la perdita di un pacchetto, riduce in ogni caso il ritmo di trasmissione dimezzando il valore della finestra di trasmissione, ottenendo così prestazioni inferiori



Le Reti Wireless

- La rete Internet ha subito, negli ultimi 10 anni, notevoli trasformazioni dovute allo sviluppo ed alla diffusione di tecnologie wireless
- Il numero di utenti mobili della rete Internet è in costante aumento, e si stimano attualmente all'incirca
 - 100 milioni di portatili
 - 30 milioni di palmari
 - 100 milioni di smart phones.
- E' dunque necessario studiare l'impatto che le tecnologie wireless hanno sui differenti strati dello stack protocollare di rete
- In particolare, sul livello **Trasporto**



Caratteristiche Reti Radio

- Le reti radio presentano caratteristiche peculiari, che hanno un impatto sulle prestazioni di connessioni TCP:
 - Perdite casuali di segmenti
 - Ritardi notevoli e variabili nel tempo
 - Capacità di canale relativamente basse
 - Canali asimmetrici
 - Disconnessioni frequenti



Caratteristiche Reti Radio

Perdite Casuali

- Le reti radio presentano un elevato tasso di perdita, causato ad esempio da:
 - interferenza co-canale
 - mobilità dell'utente
 - multipath fading
 - disconnessioni dell'utente dovute a limitazioni dell'area di copertura
- Il tasso di perdita dei pacchetti può essere nell'ordine di 1-10% (misurato sperimentalmente)
- Tali perdite degradano notevolmente le prestazioni di connessioni TCP, che dimezzano il proprio ritmo di trasmissione ritenendo tali perdite dovute a congestione
- Vedremo in seguito dei semplici modelli matematici che legano le prestazioni di sorgenti TCP al tasso di perdita dei segmenti p ed al ritardo medio sperimentati dalla connessione.



Caratteristiche Reti Radio

Ritardi variabili e Canali a bassa capacità

- ❑ Alcuni canali radio possono avere bassa capacità (20 kbit/s), e di conseguenza portare una connessione ad avere elevati ritardi, ovvero un Round Trip Time (RTT) medio molto elevato
- ❑ Vedremo che le performance di una sorgente TCP sono all'incirca inversamente proporzionali al RTT sperimentato.

$$RTO = RTT_{medio} + 4RTT_{dev}$$

- ❑ Inoltre, si è misurato sperimentalmente che in ambienti radio i ritardi misurati da una connessione TCP variano notevolmente.
- ❑ Questa caratteristica può degradare le prestazioni di connessioni TCP, in quanto il Retransmission Time-Out (RTO) viene settato ad un valore molto più elevato del necessario, proporzionale alla deviazione standard del ritardo misurato:



Caratteristiche Reti Radio

Canali Asimmetrici

- Le connessioni TCP sono "ACK Clocked", ovvero:
 - Nel caso in cui i riscontri, per qualche ragione, non giungano al trasmettitore, la sorgente non può inviare altri segmenti e la trasmissione si blocca.
- Questo fenomeno può verificarsi ad esempio quando alcune sorgenti "catturano" il canale radio, impedendo per lungo tempo alle altre connessioni attive in parallelo di ricevere riscontri.
- Come conseguenze:
 - Le performance delle sorgenti TCP sono degradate
 - Le trasmissioni, quando avvengono, avvengono in burst.



Caratteristiche Reti Radio

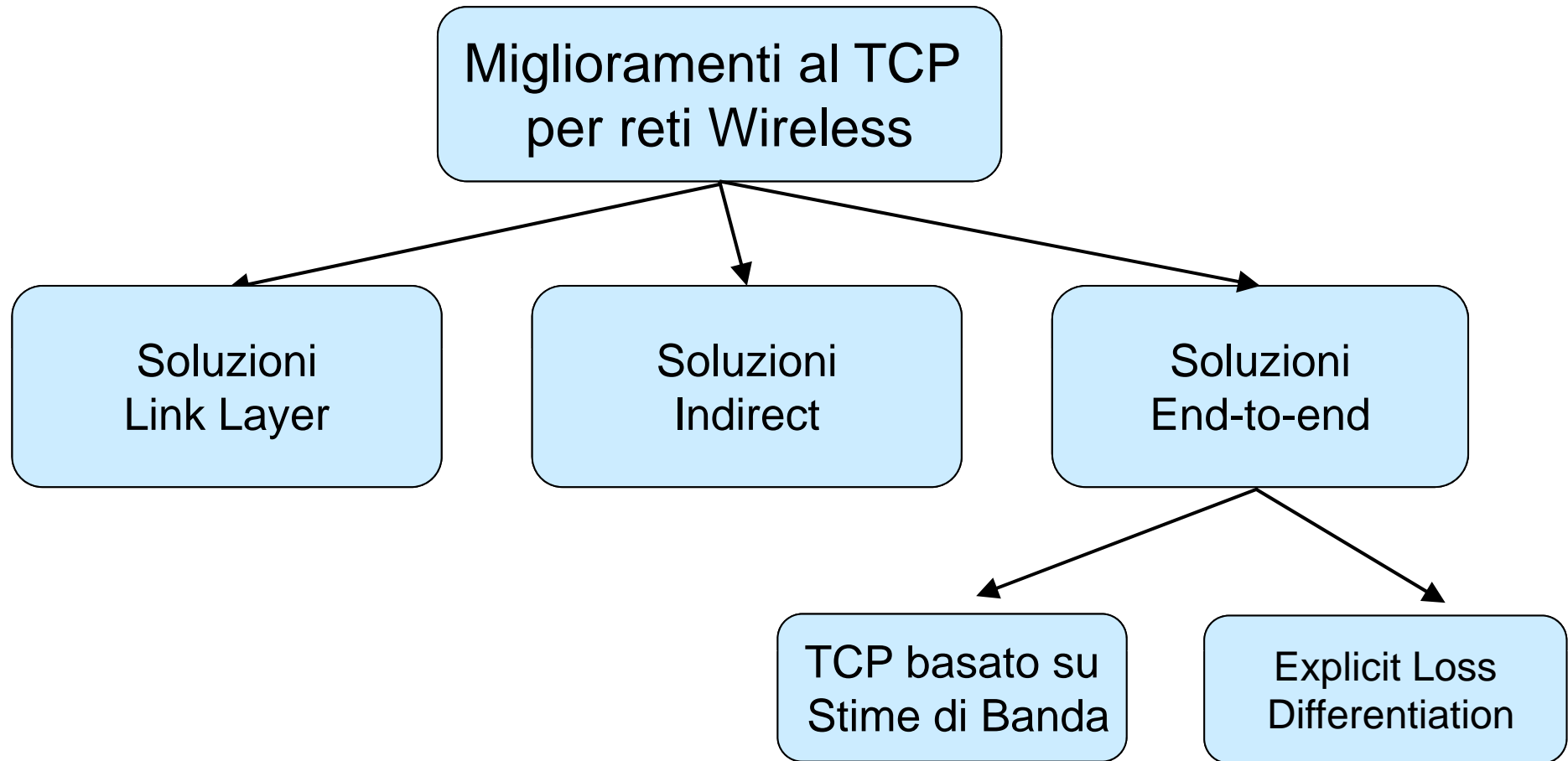
Disconnessioni

- Poiché spesso gli utenti di reti wireless sono mobili, possono sperimentare disconnessioni frequenti e prolungate a causa di fenomeni come:
 - Handoff tra celle (base-stations) differenti
 - Mancanza di copertura (ad es. quando l'utente entra in una galleria)
 - Fading profondo e prolungato nel tempo
- In questi casi la connessione TCP osserva lo scadere di vari Time-Out di ritrasmissione
- Quando il "blackout" termina, la sorgente TCP ricomincia a trasmettere in slow-start, partendo da una finestra pari ad un segmento.
- Come conseguenza, le disconnessioni possono ridurre notevolmente le performance di connessioni TCP



TCP su reti cablate/radio

Tassonomia delle soluzioni proposte in letteratura



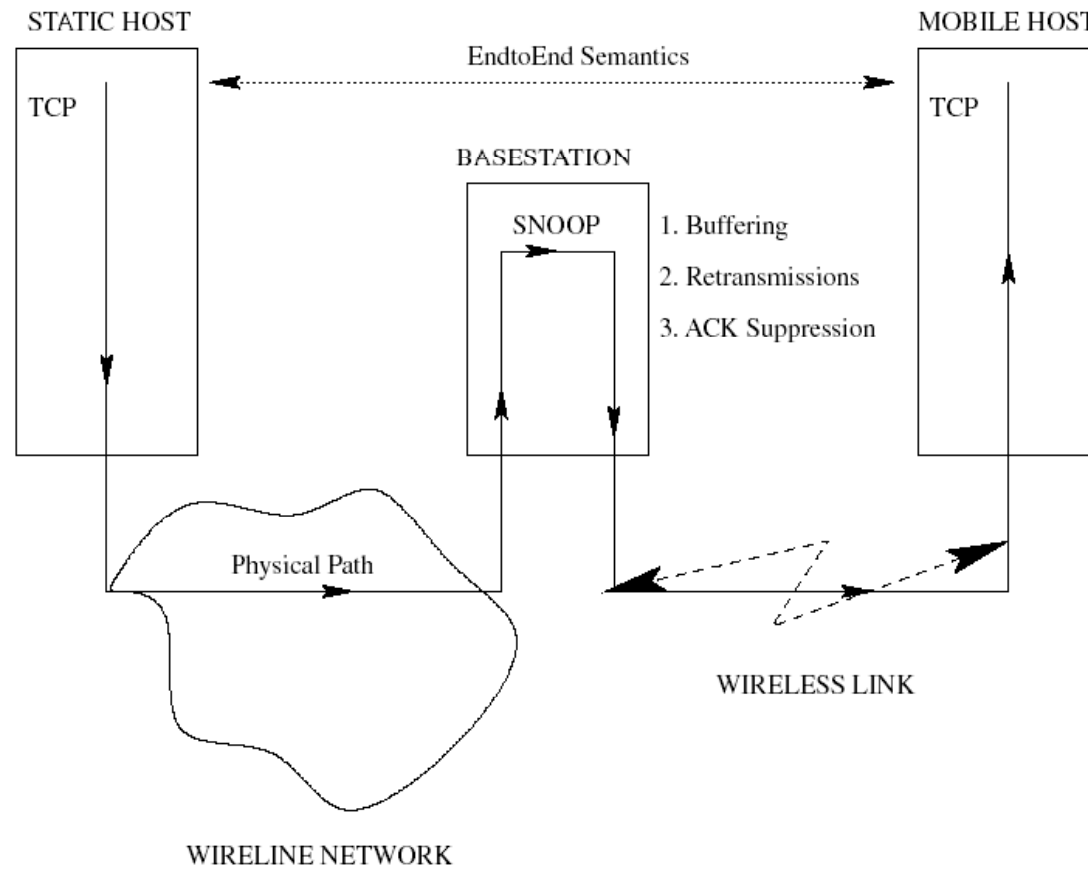


Soluzioni Link Layer

- ❑ Le soluzioni che ricadono in questa categoria cercano di mascherare le caratteristiche delle reti wireless utilizzando speciali meccanismi di livello link-layer sui canali radio
- ❑ Si tratta di soluzioni trasparenti rispetto al protocollo TCP soprastante
- ❑ Di solito sfruttano la conoscenza del comportamento di TCP per migliorarne le performance
- ❑ Soluzione tipica: bufferizzare i pacchetti ai nodi appartenenti al link wireless (ad es. access points) e ritrasmettere i pacchetti persi a causa di errori sul link wireless
- ❑ Conseguenza: l'host è esposto solamente a perdite dovute a congestione



Esempio: Architettura del protocollo Snoop





Soluzioni Link Layer

- ❑ Le soluzioni Link Layer presentano tutte, dunque, le seguenti caratteristiche:
- 1. Mascherano le caratteristiche tipiche dei link wireless ai protocolli di livello trasporto soprastanti (TCP)
- 2. Sono *trasparenti* rispetto a TCP, e dunque non richiedono cambiamenti nello stack protocollare negli host e server coinvolti nella connessione TCP
- 3. Possono tenere conto o meno dei protocolli di livello trasporto soprastanti oppure no. Evidentemente le soluzioni Link Layer di tipo "TCP-aware" sono in genere le più efficaci.
- 4. Richiedono intelligenza aggiuntiva, buffer e funzioni di ritrasmissione implementati nei nodi wireless (in particolare in base-station o access-points)
- 5. Mantengono la semantica end-to-end del TCP

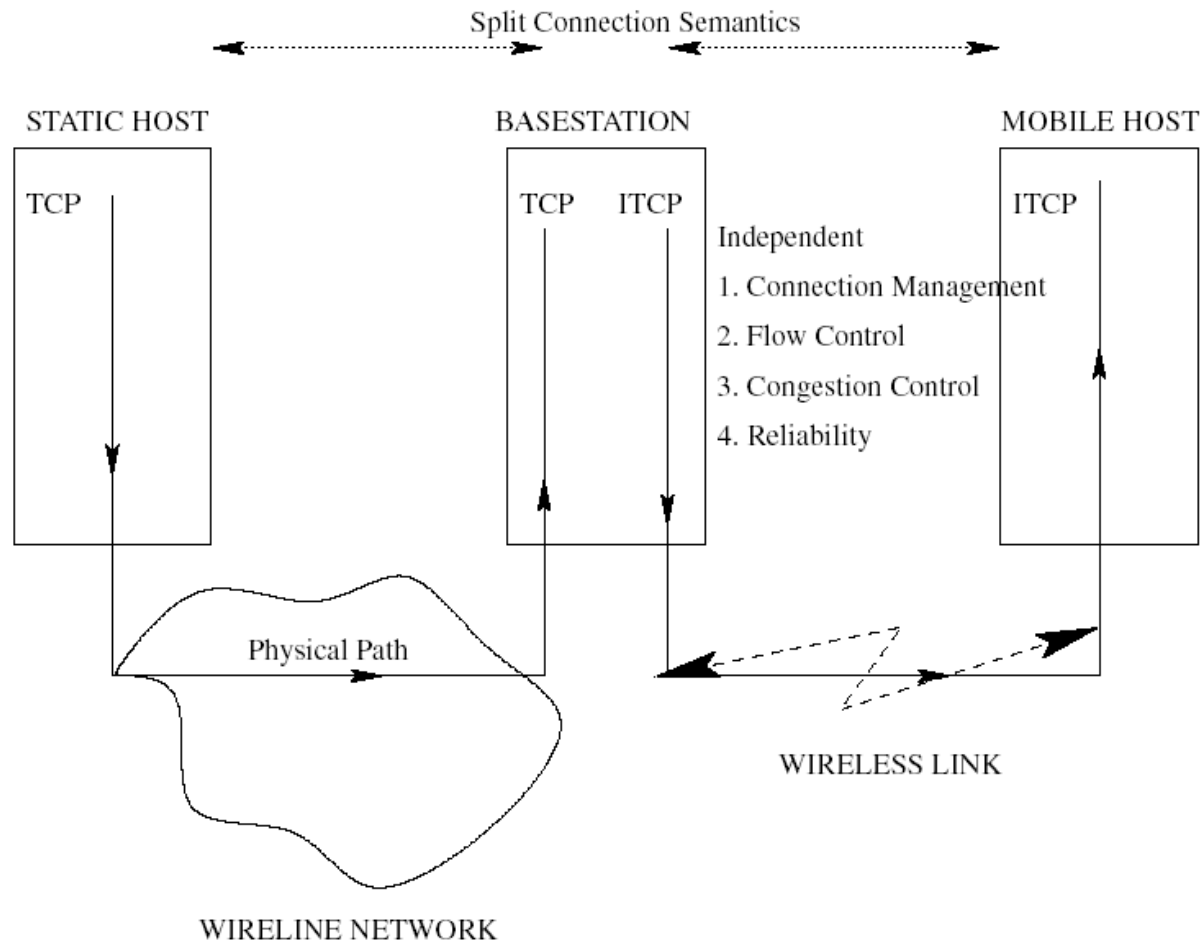


Soluzioni Indirect

- Le soluzioni che ricadono in questa categoria cercano anch'esse di mascherare le caratteristiche delle reti wireless, *spezzando* la connessione TCP alla base station
- Una singola connessione viene dunque suddivisa in due "sotto-connessioni":
 - La prima tra il mobile host e la base station
 - La seconda tra la base station e l'host fisso (ad es. Web server)
- Il tal modo sulla prima connessione (la più critica) può essere impiegato un protocollo di trasporto "ad hoc", adattato alle caratteristiche uniche del link wireless.



Esempio: architettura del protocollo ITCP





Soluzioni Indirect

- Le soluzioni Indirect presentano le seguenti caratteristiche:
 1. La connessione end-to-end viene suddivisa in due "sotto-connessioni" a livello della base-station
 2. La semantica end-to-end del TCP *non* viene mantenuta
 3. Vengono utilizzati protocolli di trasporti specializzati ed adattati alle caratteristiche del link wireless
 4. Tali protocolli richiedono poche modifiche ai mobile host al prezzo di maggior complessità nella base-station



Soluzioni End-to-end

- Le soluzioni di questo tipo mantengono la semantica end-to-end tipica del TCP
- Richiedono però modifiche che possono essere localizzate
 - Al lato trasmettitore (server)
 - Al lato ricevitore (host)
 - Ad entrambi i lati
- La modifica al solo lato server permette un'immediata diffusione del nuovo protocollo nella rete Internet, mentre un cambiamento nello stack protocollare degli host risulta di più lenta penetrazione.



Soluzioni End-to-End

- Le soluzioni End-to-end presentano le seguenti caratteristiche:
 1. Mantengono la semantica end-to-end del TCP
 2. Permettono di realizzare algoritmi sofisticati di controllo della congestione e recupero degli errori
 3. Possibilità di essere implementate e diffuse facilmente nella rete Internet



Esempi di Soluzioni End-to-End

Esistono in letteratura algoritmi che forniscono una stima end-to-end dello stato della rete. Fra questi:

- Algoritmi che differenziano **implicitamente** la causa delle perdite a partire da una stima della banda: Il ritmo di trasmissione tende ad essere proporzionale alla misura del throughput attuale.

TIBET e TCP Westwood

- Algoritmi che differenziano **esplicitamente** la natura delle perdite (LDA - Loss Differentition Algorithms): stimano la causa della perdita dei segmenti, utilizzando le variabili interne del TCP. Si usano reazioni diverse a secondo del tipo di perdita.

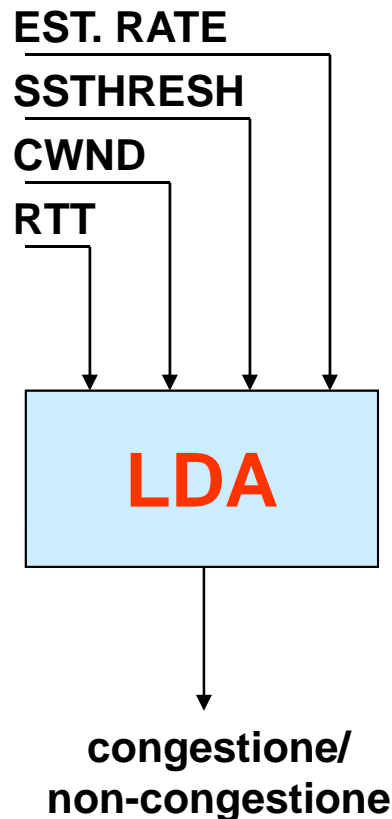


TCP NewReno-LDA



Esempio: TCP basato su Algoritmi di Loss Differentiation (LDA)

Stima end-to-end dello stato di congestione della rete



- Tali algoritmi stimano la causa della perdita dei pacchetti "in volo", utilizzando le variabili di stato del TCP
- La stima fornita permette di prendere decisioni differenti in caso di perdita di pacchetti
- Esempi di algoritmi proposti in letteratura: TCP NewReno-LDA, TCP Veno, J-TCP...



TCP NewReno-LDA

Controllo di congestione che utilizza la stima LDA

- A fronte di una perdita di pacchetto, il TCP trasmittente (lato server, dunque) utilizza la stima LDA per decidere se ridurre il ritmo di trasmissione
- Nel caso la perdita sia attribuita ad errori di trasmissione sul canale wireless, il ritmo *non* viene ridotto



Incremento del goodput su canali radio

- Nel caso la perdita venga attribuita alla congestione della rete, il ritmo di trasmissione viene ridotto come in TCP NewReno



Fairness nei confronti di connessioni TCP tradizionali



Modelli Matematici per sorgenti TCP su link Wireless

□ Obiettivi:

- Comprendere le dinamiche fondamentali del TCP
- Capire la relazione fondamentale tra tasso di perdita dei pacchetti, ritardo e prestazioni di connessioni TCP
- Utilizzare tali modelli per predire le performance di connessioni TCP in reti eterogenee comprendenti canali radio
- Utilizzare tali modelli per progettare algoritmi cosiddetti "TCP-friendly" per gestire traffico multimediale (per es. TFRC)



Modelli Matematici per sorgenti TCP su link Wireless

- Esistono in letteratura diversi modelli matematici che consentono di valutare le prestazioni di sorgenti TCP in presenza di canali wireless
- 1. M. Mathis, J. Semke, J. Mahdavi, T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", Computer Communications Review, vol. 27(3), July 1997
- 2. J. Padhye, V. Firoiu, D. Towsley, J. Kurose, "Modeling TCP Reno performance: a simple model and its empirical validation", IEEE/ACM Transactions on Networking, Vol. 8(2), April 2000, pagine: 133 - 145
- 3. N. Cardwell, "Modeling the Performance of Short TCP Connections"



Modello connessioni Long-Lived

□ Ipotesi:

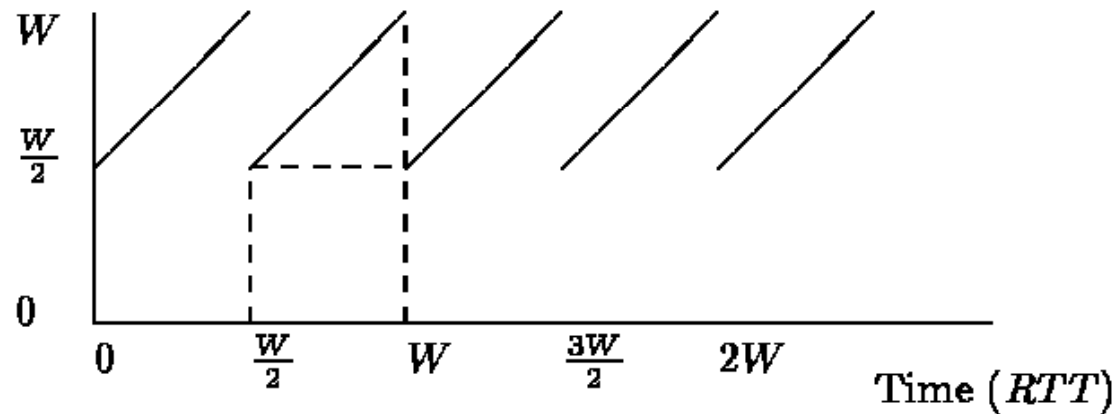
- La connessione effettua un trasferimento file lungo, ad esempio usando il protocollo FTP ("long-lived" connections).
- Le perdite sono tutte rilevate dalla sorgente TCP tramite ricezione di 3 ACK duplicati (e dunque non scadono mai Retransmission TimeOuts)
- Sempre in Congestion Avoidance
- Il Round Trip Time (RTT) della connessione è costante (ovvero: la banda disponibile alla connessione è sufficiente e non si creano code nei buffer lungo il path).
- Il link casuale perde pacchetti in maniera indipendente e casuale, con tasso di perdita pari a p
- *Conseguenza*: il link trasmette in media $\frac{1}{p}$ pacchetti corretti seguiti da una perdita wireless

✓ Siamo interessati a calcolare lo steady state goodput



Modello connessioni Long-Lived

- Con queste ipotesi, l'andamento della congestion window (cwnd) risulta periodico:



- Sia W il valore il massimo valore assunto da cwnd, nel momento in cui il canale wireless perde un pacchetto
 - All'equilibrio, la minima finestra sarà dunque pari a $W/2$
 - Se il ricevitore riscontra ogni segmento, ci vorranno esattamente $W/2$ Round Trip Times alla finestra per ritornare al valore W



Modello connessioni Long-Lived

- Ogni ciclo dura quindi $\frac{W}{2}$ Round Trip Times, ovvero $\frac{W}{2}RTT$ secondi
- Il numero totale di segmenti trasmessi in tale ciclo è pari all'area sottesa, come indicato in figura, ovvero
$$\left(\frac{W}{2}\right)^2 + \frac{1}{2}\left(\frac{W}{2}\right)^2 = \frac{3}{8}W^2$$
- Per definizione, in ogni ciclo vengono trasmessi anche $\frac{1}{p}$ segmenti, per cui risulta:

$$\frac{3}{8}W^2 = \frac{1}{p}$$



Modello connessioni Long-Lived

- Dunque W può essere espresso in funzione di p

$$W = \sqrt{\frac{8}{3p}}$$

- Il goodput ottenuto dalla connessione risulta dunque pari a:

$$Goodput = \frac{\text{dati trasmessi in un ciclo}}{\text{durata ciclo}} = \frac{MSS \cdot \frac{3}{8} W^2}{RTT \cdot \frac{W}{2}} = \frac{\frac{MSS}{p}}{RTT \cdot \sqrt{\frac{2}{3p}}}$$

- MSS= Maximum Segment Size (bit)
- RTT= Round Trip Time (secondi)



Modello connessioni Long-Lived

- Raccogliendo i termini comuni e definendo la costante
otteniamo il risultato:

$$C = \sqrt{\frac{3}{2}}$$

$$Goodput = \frac{MSS}{RTT} \frac{C}{\sqrt{p}}$$

- Nel caso in cui la connessione utilizzi la politica dei Delayed ACKs (come raccomandato nell'RFC 2581), inviando un riscontro ogni 2 segmenti ricevuti, semplici ragionamenti conducono alla stessa espressione, ponendo però:

$$C = \sqrt{\frac{3}{4}}$$



Modello connessioni Long-Lived

- Limitazioni:
- Nel caso in cui la receiver-window sia più bassa della congestion window, è il ricevitore a limitare il ritmo di trasmissione della sorgente
- Non viene considerato l'impatto dei Time-Out sulle performance di TCP
- Le connessioni TCP possono richiedere vari cicli per raggiungere uno stato stazionario, per cui connessioni brevi ("short-lived connections") non risultano ben modellizzate.



Modello connessioni Long-Lived

Modello completo per connessioni Long-Lived

- A differenza del modello precedente, il modello proposto in J. Padhye, V. Firoiu, D. Towsley, J. Kurose, "Modeling TCP Reno performance: a simple model and its empirical validation", IEEE/ACM Transactions on Networking, Vol. 8(2), April 2000, pagine: 133 – 145 cattura anche l'effetto dei Time-Out di ritrasmissione sulle performance di sorgenti FTP
- Da misure effettuate nella rete Internet si verifica infatti che il numero degli eventi di Time-Out è elevato, e dunque in molti casi non trascurabile.



Modello connessioni Long-Lived

- La derivazione del modello è molto più complessa rispetto al precedente. L'espressione approssimata del goodput ottenuto in stato stazionario risulta essere la seguente:

$$\text{Goodput} \approx \frac{MSS}{RTT \sqrt{\frac{2bp}{3}} + T_0 \min(1, 3\sqrt{\frac{3bp}{8}}) p(1 + 32p^2)}$$

- p rappresenta il tasso di perdita sul canale (si assumono perdite indipendenti da segmento a segmento)
- b rappresenta il numero di segmenti riscontrati da un ACK ($b=2$ se la sorgente implementa la politica dei Delayed ACKs, come da standard, $b=1$ altrimenti)
- T_0 rappresenta la durata di un Time-Out al momento dello scadere del primo Time-Out per un segmento. Gli eventuali Time-Out successivi dureranno $2T_0$, $4T_0$, $8T_0$, ... e così via fino a $64T_0$, dopodiché il timeout viene mantenuto pari a $64T_0$ per i Time-Out successivi.



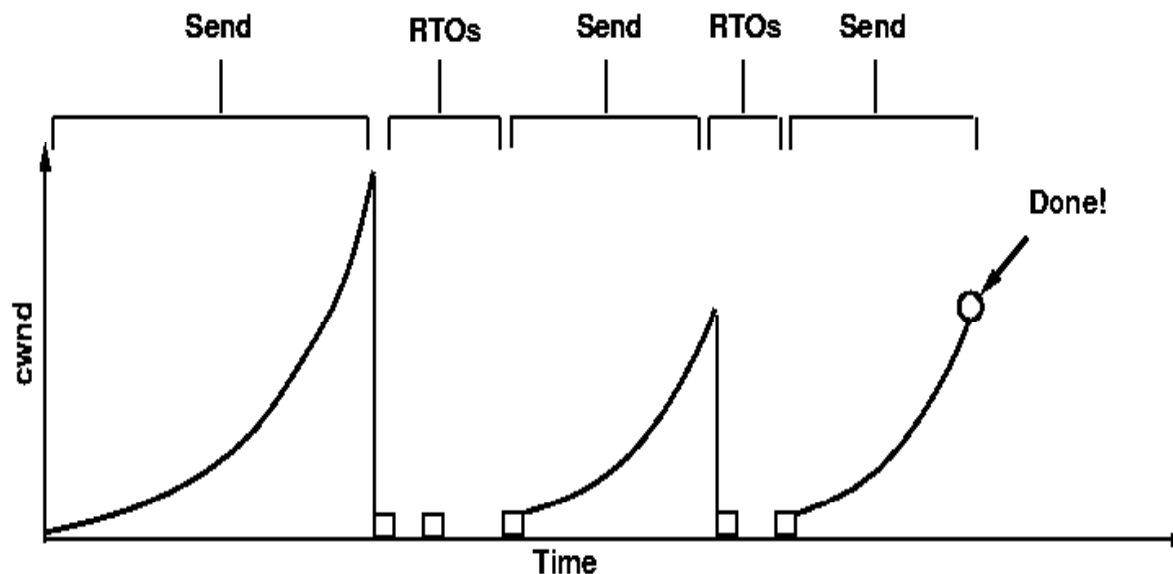
Modello connessioni Short-Lived

- A differenza dei due modelli precedenti, il modello proposto in: **N. Cardwell, "Modeling the Performance of Short TCP Connections"**, si focalizza sulle connessioni di breve durata ("short-lived"), tipiche di transazioni HTTP (Web browsing, ecc..)
- Poiché molte connessioni TCP sono di questo tipo, il modello risulta particolarmente interessante
- Vengono inoltre considerate le performance di connessioni TCP short-lived in presenza di canali radio affetti da errori casuali.
- Per queste connessioni siamo interessati a calcolare il tempo totale di trasferimento



Modello connessioni Short-Lived

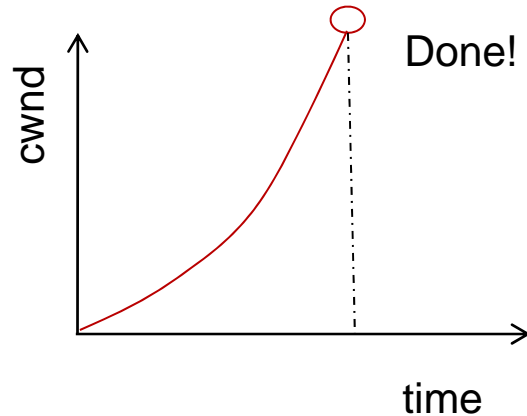
- ❑ Connessioni short-lived hanno in genere finestre troppo basse per entrare nella fase di Fast-Retransmit / Fast-Recovery, come ipotizzato nei modelli precedenti.
- ❑ Quindi il loro andamento è molto meglio modellizzato da una successione di fasi di slow-start, seguite da uno o più time-out di ritrasmissione, come illustrato in figura:





Modello connessioni Short-Lived

MODELLO ERROR FREE



$$cwnd_{i+1} = cwnd_i + ack_i = \left(1 + \frac{1}{b}\right) cwnd_i = \left(1 + \frac{1}{b}\right)^i w_0$$

$$r = 1 + \frac{1}{b}$$

$$data_i = \sum_{k=1}^i \left(1 + \frac{1}{b}\right)^{i-k} w_0 = w_0 \frac{r^i - 1}{r - 1}$$

$$i = \log_r \left(\frac{data(r-1)}{w_0} + 1 \right) = \log_r \left(\frac{d(r-1)}{MSS \cdot w_0} + 1 \right)$$

$$t_{TOT} = RTT \cdot \log_r \left(\frac{d(r-1)}{MSS \cdot w_0} + 1 \right) + RTT + t_{ACK}$$



Modello connessioni Short-Lived

MODELLO CON ERRORI

l : number of lost segments

p : loss probability

$$p = \frac{l}{l + data}$$

$$l = \frac{data \cdot p}{1 - p}$$

Probability that a loss leads to a RTO (proof in the paper) :

$$Q(p) = \min \left(1; \frac{3}{\sqrt{\frac{8}{3bp}}} \right)$$



Modello connessioni Short-Lived

MODELLO CON ERRORI

n : number of RTOs

$$n = l \cdot Q(p)$$

number of consecutive RTOs : $\frac{1}{1-p}$

u : number of groups of RTOs

$$u = \frac{l \cdot Q(p)}{\frac{1}{1-p}} = l \cdot Q(p) \cdot (1-p)$$

$$t_u = T_0 \frac{1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6}{1-p}$$

$$t_{RTO} = ut_u$$



Modello connessioni Short-Lived

MODELLO CON ERRORI

ν : number of slow start phases

$$\nu = u + 1$$

e : data trasfered on average per phase

$$e = \frac{data + l}{\nu}$$

$$t_{xfer} = \nu \log_r \left(\frac{e(r-1)}{w_0} + 1 \right) RTT$$

$$t_{TOT} = t_{RTO} + t_{xfer} + RTT + t_{ACK}$$



Bibliografia

- M. Allmann, V. Paxson, W. Stevens, "*TCP Congestion Control*", RFC 2581
- S. Floyd, T. Henderson, "*The NewReno Modification to TCP's Fast Recovery Algorithm*", RFC 2582
- V. Paxson, M. Allmann, "*Computing TCP Retransmission Timer*", RFC 2988
- Van Jacobson, "*Congestion Avoidance and Control*", SIGCOMM 1988
- S. Bregni, D. Caratti, F. Martignon, "*Improving TCP Performance over Wireless Networks using Loss Prediction*", IEEE GLOBECOM '03.
- A. Capone, L. Fratta, F. Martignon, "*Bandwidth Estimation Schemes for TCP over Wireless Networks*", IEEE Trans. on Mobile Computing, vol. 3, no. 2, April 2004, pp. 129-143.