

Università di Genova  
Facoltà di Ingegneria

Telematica  
**6. Controllo di flusso e  
congestione**

Prof. Raffaele Bolla



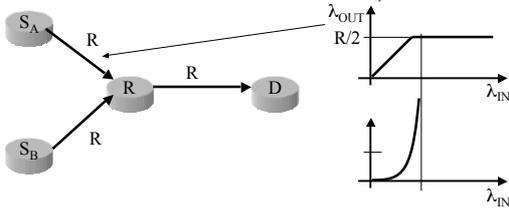
### Controllo di flusso

- Si tratta delle tecniche che consentono ad una sorgente di adattare il proprio tasso di trasmissione a quello correntemente disponibile al ricevitore e nella rete.
- Sono quindi tecniche che regolano/controllano il livello del traffico introdotto nella rete

6.2

### Controllo di flusso Senza il controllo di flusso

- In assenza di controllo di flusso si potrebbe incorrere in queste conseguenze:
  - Al crescere del carico, cresce il ritardo (perché le risorse della rete sono limitate)



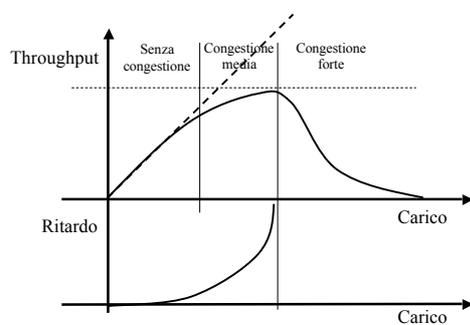
6.3

### Controllo di flusso Senza il controllo di flusso

- Se il carico diventa ancora più alto si hanno due effetti
  - Se i nodi hanno dei buffer grandi, i ritardi superano i *timeout* (recupero d'errore) e si hanno ritrasmissioni inutili
  - Se i nodi hanno buffer piccoli (o se il carico continua a mantenersi molto elevato a lungo) si hanno delle perdite
- L'effetto finale è comunque:
  - **Spreco di risorse** (pacchetti scartati e duplicati)
  - **Aumento del carico offerto** (ritrasmissioni)

6.4

### Controllo di flusso Effetti della congestione



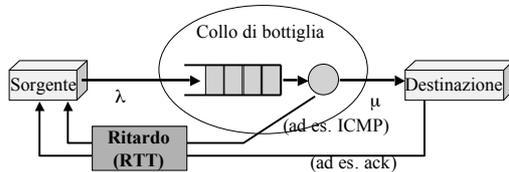
6.5

### Controllo di flusso

- Quindi, il controllo di flusso è una delle tecniche che permettono di operare un controllo della congestione
- Gli obiettivi del controllo di flusso sono:
  - Limitare le perdite nei *buffer* (nella rete e alla destinazione) e limitare il ritardo ( $\Rightarrow$  migliorare il *throughput*)
  - Essere equo
  - Usare poche risorse di rete (poca segnalazione)
  - Stabile
  - Scalabile
  - Semplice da implementare

6.6

### Controllo di flusso Modello



Il controllo di flusso è un adattatore di tasso con retroazione ritardata

Il ritardo è dato dal **Round Trip Time (RTT)** e ha due effetti:

1. Ritarda la conoscenza dello stato del "collo di bottiglia".
2. Ritarda l'effetto del controllo.

6.7

### Controllo di flusso Modello

- L'influenza del ritardo non è assoluta, ma va riportata alla banda:

$$\text{Prodotto banda-ritardo} = Pbr = B \times RTT$$

più è alto questo prodotto più è critico il meccanismo di controllo

- Esempio

- RTT = 1 ms; B = 1 Mb/s; Pbr = 1 Kbit  
ossia è stato inviato 1 Kbit prima che ritorni la retroazione
- RTT = 1 ms; B = 1 Gb/s; Pbr = 1 Mb/s

6.8

### Controllo di flusso Classificazione

- **Anello aperto (Open-loop)** (preventivo):
  - Descrivere il traffico.
  - "Costringerlo" comportarsi come la descrizione impone.
  - Effettuare un controllo d'accesso.
  - Assicurare le risorse.
- **Anello chiuso (Close-loop)**: adattamento dinamico del tasso su base retroazione, due tipi:
  - Retroazione esplicita.
  - Retroazione implicita.
- **Schemi ibridi (ATM -ABR)**

*Si osservi che gli ultimi due metodi possono essere usati solo con sorgenti elastiche*

6.9

### Controllo di flusso Anello chiuso - Classificazione

- In prima analisi si può distinguere fra meccanismi che regolano il flusso in funzione:
  - della sola destinazione (1° generazione);
  - della destinazione e della rete (2° generazione)
- Nella prima generazione si distingue fra
  - *On-Off*: segnale esplicito dalla destinazione che blocca o riattiva la trasmissione (Xon, Xoff);
  - *Stop-and-wait*: il meccanismo di recupero d'errore usato anche per il controllo di flusso;
    - » Inefficiente se Prop/Transp è alto
    - » Come tutti i casi in cui si realizza in modo congiunto il recupero d'errore ed controllo di flusso, non si riesce a distinguere fra indicazioni di rallentare e perdite.
  - *Sliding Window con finestra statica*: Go-back-N o *Selective repeat* usati anche per il controllo di flusso.

6.10

### Controllo di flusso Anello chiuso - Classificazione

- I meccanismi di 2° generazione si distinguono per
  - Tipo di retroazione:
    - » Esplicita
    - » Implicita
  - Tipo di controllo
    - » Finestra dinamica
    - » Tasso dinamico
  - Localizzazione del controllo
    - » End-to-end
    - » Hop-by-hop

6.11

### ATM: Controllo di flusso per il traffico ABR End to End Rate based flow Control (EERC)

- Caratteristiche:
  - Esplicito (con *feedback*)
  - Tasso dinamico
  - End to end
- La segnalazione avviene attraverso delle celle dedicate chiamate **Resource Management (RM)**
- Ne vengono generate (dalla sorgente) una ogni **NRM (Number of RM)** che vale 32 di default.
- Le RM che viaggiano dalla sorgente alla destinazione vengono chiamate **Forward RM (FRM)**.
- La sorgente che riceve una FRM la ri-invia indietro (i collegamenti sono tutti *full-duplex*); mentre compie il cammino inverso la cella RM prende il nome di **Backward RM (BRM)**.

6.12

### Prestazioni del TCP

- Per poter valutare il comportamento del controllo di flusso del TCP bisogna elencare le principali assunzioni di progetto che la sua definizione ha richiesto:
  - La rete deve avere una bassa probabilità d'errore (per es., non essere *wireless*)
  - La banda complessiva deve rimanere relativamente stabile (altrimenti il TCP tende a diventare troppo conservativo).
  - Il flusso viaggia su un singolo percorso (ottimo)
  - Buffer sono serviti FIFO e possibilmente dimensionati secondo il prodotto banda-ritardo.
  - Le connessioni hanno durata "non troppo corta".
  - La dimensione del campo dati è relativamente grande e le bande sono adeguate.
  - Tutti i flussi devono usare il TCP.

6.13

### Prestazioni del TCP

- In prima analisi queste assunzioni portano a dedurre che il TCP tende a funzionare in modo poco efficiente se:
  - Le reti hanno RTT elevati con bande elevate (elevato prodotto banda-ritardo)
  - Le connessioni hanno breve durata
  - Si hanno elevati tassi d'errore
  - Esiste un significativo carico trasportato su UDP.

6.14

### Prestazioni del TCP

- Considerazioni più puntuali sulle prestazioni possono essere fatte in relazione a
  - Influenza dei buffer di trasmissione
  - Aspetti di equità
  - Effetti di "raggruppamento" con elevati prodotti banda/ritardo
  - Influenza della latenza
  - Connessioni brevi
  - Oscillazioni

6.15

### Prestazioni del controllo di flusso del TCP Buffer al ricevitore

- Le dimensioni dei buffer al ricevitore sono importanti nel determinare il massimo tasso di trasmissione

$$R_{\max} = \min \left\{ B [\text{bit/s}], \frac{\text{Dim. Buffer} [\text{bit}]}{RTT [\text{s}]} \right\}$$

- Ad esempio: con  $B = 2 \text{ Mb/s}$ ;  $RTT = 0,5 \text{ s}$ ; Buffer = 16 Kbit si ottiene  $R_{\max} = 32 \text{ Kb/s}$ .  
Per raggiungere la velocità massima sarebbe necessario un buffer di 1 Mbit.
- Se  $B = 32 \text{ Mb/s}$  e  $RTT = 1 \text{ s}$ , per raggiungere la velocità della linea, il buffer dovrebbe essere pari a 32 Mbit.

6.16

### Prestazioni del controllo di flusso del TCP Effetto della *congestion avoidance*

- Se si suppone che il meccanismo di *congestion avoidance* prenda il sopravvento, si ha un comportamento ciclico in cui la finestra cresce linearmente fino ad un massimo per tornare ad una dimensione pari alla metà della massima raggiunta.
- Se RTT è circa costante e la finestra massima (in # segmenti di dimensione *Minimum Segment Size* (MSS)) è pari a  $W$ , allora il tasso di trasmissione  $R$  diventa:

$$\frac{W \cdot MSS}{2 \cdot RTT} \leq R \leq \frac{W \cdot MSS}{RTT} \Rightarrow \bar{R} = \frac{3 \cdot W \cdot MSS}{4 \cdot RTT}$$

6.17

### Prestazioni del controllo di flusso del TCP Effetto della *congestion avoidance*

- Un conto più preciso può essere fatto come segue:
  - Se la una perdita avviene ogni volta che la finestra passa da  $W/2$  a  $W$ , vuol dire che si perde un pacchetto ogni  $D = RTT \times W/2$

- Durante il periodo  $D$  vengono trasmessi

$$N_T = \sum_{i=0}^{W/2} \left( \frac{W}{2} + i \right) = \frac{3}{4} W \left( \frac{W}{2} + 1 \right) = \frac{3}{8} W^2 \text{ pacchetti}$$

- Il tasso di perdita  $P$  durante il periodo  $D$  è pari a

$$P = \frac{1}{\frac{3}{8} W^2} = \frac{8}{3W^2} \Rightarrow W = \sqrt{\frac{8}{3P}}$$

6.18

**Prestazioni del controllo di flusso del TCP**  
**Effetto della congestion avoidance**

- Il tasso medio di trasmissione R diventa

$$R = \frac{N_T \cdot MSS}{D} = \frac{3W^2 \cdot MSS}{8D} = \frac{3W^2 \cdot MSS}{8RTT \frac{W}{2}} = \frac{3W \cdot MSS}{4RTT}$$

$$= \frac{3\sqrt{8} \cdot MSS}{4\sqrt{3P} \cdot RTT} \approx 1,22 \frac{MSS}{RTT \sqrt{P}}$$

- Vale solo per  $P \leq 1\%$ , altrimenti entra in gioco la *slow-start*
- Ad es. con  $P = 1\%$ ,  $RTT = 100$  ms,  $MSS = 200$  ottetti, si ha  $R = 195$  Kb/s

6.19

**Prestazioni del controllo di flusso del TCP**  
**Equità**

- Si consideri una situazione del tipo:

- Supponendo che le due connessioni
  - abbiano la stessa MSS e lo stesso RTT
  - Abbiano molti dati da inviare
  - Siano le uniche su queste linee
  - Non applichino la fase di *slow-start*

6.20

**Prestazioni del controllo di flusso del TCP**  
**Equità**

La banda totale è maggiore della disponibile, quindi si ha perdita e ciascuno ritorna a metà della finestra attuale (C)

Stanno usando meno banda della disponibile (A di più e B di meno) e quindi ciascun flusso aumenta di una MSS per RTT

6.21

**Prestazioni del controllo di flusso del TCP**  
**Raggruppamento**

- La presenza di elevato prodotto banda-ritardo dà origine, oltre a limitazioni sulla velocità di risposta del controllo ed effetti legati ai buffer nei nodi (limitazioni sulla velocità di picco), ad un effetto di “**raggruppamento**” delle trasmissioni ogni RTT:
  - Alla scadenza di ogni RTT si tende a ricevere gruppi di ACK consecutivi e quindi ad inviare segmenti al tasso massimo d’uscita, saturando i buffer nei nodi a valle degli eventuali colli di bottiglia (anche se il tasso medio risultante dalle trasmissioni risulterebbe compatibile con il collo di bottiglia).

6.22

**Prestazioni del controllo di flusso del TCP**  
**Raggruppamento**

Segmenti con  $MSS = R/(4 RTT)$  quindi il tempo di tx su A-B è  $RTT/8$ , mentre su B-C è  $RTT/4$

Arrivi a B

Partenze da B

Il buffer è già pieno e quindi perdo il pacchetto numero 7

6.23

**Prestazioni del controllo di flusso del TCP**  
**Latenza**

- Con latenza si intende, in questo caso, il tempo che intercorre fra l’istante in cui la sorgente apre una connessione per il trasferimento di un oggetto e quello in cui l’oggetto è stato completamente ricevuto dalla destinazione
- La sua valutazione serve a valutare l’impatto dello *slow-start* nelle comunicazioni
- Consideriamo un contesto semplificato del tipo:

Il client apre una connessione col Server e chiede il trasferimento di un file (oggetto)

6.24

**Prestazioni del controllo di flusso del TCP**  
**Latenza**

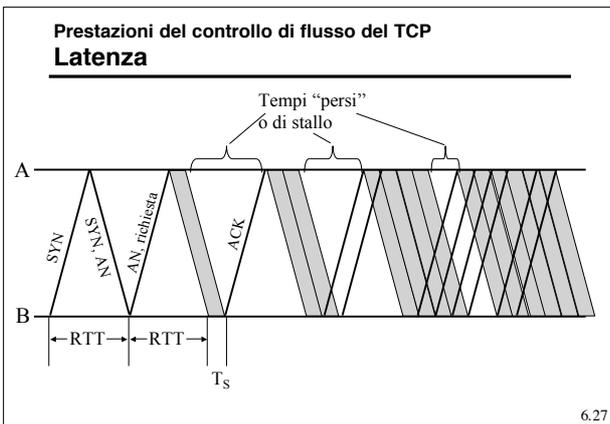
- Si faccia le seguenti assunzioni:
  - Il buffer in ricezione è ampio (non limita mai la finestra)
  - Non vengono mai persi pacchetti
  - Il peso delle intestazioni viene trascurato
  - L'oggetto da trasferire è composto da un numero intero  $N_S$  di segmenti di dimensione MSS [bit]
  - Tutti i pacchetti diversi dai segmenti che trasportano dati (per esempio i pacchetti per l'apertura della connessione o gli ACK) hanno un tempo di trasmissione trascurabile
  - La soglia iniziale del TPC è molto grande e non viene mai raggiunta

6.25

**Prestazioni del controllo di flusso del TCP**  
**Latenza**

- Notazioni
  - $O$  : dimensione dell'oggetto da trasmettere in bit
  - $R$  : tasso di trasmissione del canale in bit/s
  - $RTT$  : ritardo di andata e ritorno senza il tempo di trasmissione; è simmetrico.
  - $N_S = O/MSS$  : numero di segmenti per oggetto.
  - $T_S = MSS/R$  : tempo di trasmissione di un segmento.
  - $T_O = O/R = T_S N_S$  : tempo di trasmissione dell'oggetto).

6.26



**Prestazioni del controllo di flusso del TCP**  
**Latenza**

- Il numero  $K$  di finestre all'interno delle quali si conclude la trasmissione dell'oggetto sono

$$K = \min \left\{ k : 2^0 + 2^1 + 2^2 + \dots + 2^{k-1} = 2^k - 1 \geq N_S \right\} = \lceil \log_2(N_S + 1) \rceil$$

- Da cui la latenza  $L$  diventa

Tempo perso nelle finestre più piccole di RTT

$$L = 2RTT + T_O + \sum_{i=1}^K [T_S + RTT - 2^{i-1}T_S] + \text{Max}\{0, \dots\}$$

1,5 RTT per il Three-way-handshake e 0,5 RTT per ricevere il primo seg.      Lunghezza attuale della finestra per il tempo di tx di un segmento

6.28

**Prestazioni del controllo di flusso del TCP**  
**Latenza**

- Se  $O$  tendesse ad infinito, il numero di RTT necessari a non avere più stallo diverrebbe

$$Q = \max \left\{ k : RTT + T_S - 2^{k-1}T_S \geq 0 \right\} = \left\lceil \log_2 \left( 1 + \frac{RTT}{T_S} \right) \right\rceil$$

- Per cui, ponendo  $P = \min\{Q, K-1\}$  si ottiene

$$L = 2RTT + T_O + \sum_{i=1}^P (RTT + T_S - 2^{i-1}T_S)$$

$$L = 2RTT + T_O + P \cdot RTT - (2^P - P - 1)T_S$$

6.29

**Prestazioni del controllo di flusso del TCP**  
**Latenza**

- Risulta più agevole valutare la latenza normalizzata  $L_{norm}$  ossia il rapporto fra  $L$  e la latenza minima  $L_{min} = 2RTT + T_O$ , ricavandone un limite superiore

$$L_{norm} = \frac{L}{L_{min}} \leq 1 + \frac{P}{2 + \frac{T_O}{RTT}}$$

- Si può facilmente osservare che, nel caso in cui  $RTT \ll T_O$ , il valore della latenza normalizzata diventa prossimo ad uno, ossia a quello di  $L_{min}$

6.30

**Prestazioni del controllo di flusso del TCP**  
**Latenza**

- Per ricavare un limite superiore a L si procede come segue
  - Anzitutto si può osservare che L cresce (in particolare la seconda parte) al crescere di P
  - Il limite massimo di P con O fissato è Q. Quindi un primo *upper bound* può essere porre  $P = Q+1 = \log_2(1+X)+1$  dove  $X = RTT/T_s$  ed il "+1" è aggiunto per eliminare l'arrotondamento all'intero superiore
  - A questo punto si possono fare i seguenti calcoli

$$L_{norm} \leq 1 + \frac{RTT \cdot P + T_s(P-2^P+1)}{2RTT + T_o} \leq 1 + \frac{P + \frac{1}{X}[\log_2(1+X) + 1 - 2 - 2X + 1]}{2 + \frac{T_o}{RTT}} = 1 + \frac{P + \frac{1}{X} \log_2(1+X) - 2}{2 + \frac{T_o}{RTT}}$$

– Osservando quindi che  $\log_2(1+X)/X$  ha un suo massimo in  $X = 1$  (dato che 1 è anche il minimo valore possibile di X) dove vale 1, si ha infine che

$$L_{norm} \leq 1 + \frac{P+1-2}{2 + \frac{T_o}{RTT}} = 1 + \frac{P-1}{2 + \frac{T_o}{RTT}} \leq 1 + \frac{P}{2 + \frac{T_o}{RTT}}$$

6.31

**Prestazioni del controllo di flusso del TCP**  
**Latenza**

- Alcuni esempi
  - MSS=536 Byte =4288 bit, RTT = 100 ms; O = 100 Kbytes = 800 Kbit; da cui si ottiene un  $K = 8$  ( $N_s = 187$ )

R	$T_o$	P	$L_{min}$	L
28 Kb/s	28,6 s	1	28,8 s	28,9 s
100 Kb/s	8 s	2	8,2 s	8,4 s
1 Mb/s	800 ms	5	1 s	1,5 s
10 Mb/s	80 ms	7	0,28 s	0,98 s

– Lo *slow-start*, quando si trasferiscono oggetti grandi, ha un'influenza significativa solo quando le velocità diventano elevate.

6.32

**Prestazioni del controllo di flusso del TCP**  
**Latenza**

- MSS=536 Byte =4288 bit, RTT = 100 ms; O = 5 Kbytes = 40 Kbit; da cui si ottiene un  $K = 4$  ( $N_s = 10$ )

R	$T_o$	P	$L_{min}$	L
28 Kb/s	1,43 s	1	1,63 s	1,73 s
100 Kb/s	0,4 s	2	0,6 s	0,757 s
1 Mb/s	40 ms	3	0,24 s	0,52 s
10 Mb/s	4 ms	3	0,2 s	0,50 s

– Anche in questo caso lo *slow-start* ha un'influenza sempre più significativa al salire delle velocità, ma l'effetto è più evidente

6.33

**Prestazioni del controllo di flusso del TCP**  
**Latenza**

- MSS=536 Byte =4288 bit, RTT = 1 s; O = 5 Kbytes = 40 Kbit; da cui si ottiene un  $K = 4$  ( $N_s = 10$ )

R	$T_o$	P	$L_{min}$	L
28 Kb/s	1,43 s	3	3,4 s	5,8 s
100 Kb/s	0,4 s	3	2,4 s	5,2 s
1 Mb/s	40 ms	3	2 s	5 s
10 Mb/s	4 ms	3	2 s	5 s

– In questo caso lo *slow-start* ha un'influenza significativa comunque, a qualunque velocità ed anche per oggetti non grandi

6.34

**Prestazioni del controllo di flusso del TCP**  
**Oscillazioni**

- Si è osservato che, in presenza di più connessioni TCP che attraversano un collo di bottiglia, i controlli di flusso tendono a sincronizzarsi esasperando il comportamento oscillatorio del meccanismo (e limitando quindi fortemente l'efficienza del sistema).
- Per ovviare a questo problema si è proposto l'uso di sistemi di scarto dalle code (RED) che cercano di "punire" le sessioni con finestre più larghe (per le quali la probabilità di scarto è più alta e che incrementano più rapidamente il tasso di trasmissione) prima che la congestione diventi troppo elevata.
- Tali sistemi hanno anche il pregio di ridurre l'incidenza di una perdita completa di ACK

6.35

**Prestazioni del controllo di flusso del TCP**  
**Oscillazioni - RED**

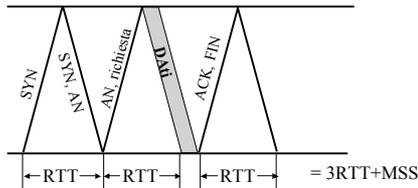
- Il metodo più noto prende il nome *Random Early Detection (RED)*; al di sopra di una certa soglia sulla coda media (calcolata con una media esponenziale) introduce una probabilità di perdita che varia linearmente con il valore della coda media misurata

Qui si scarta i pacchetti in arrivo a caso con probabilità p

6.36

### Prestazioni del controllo di flusso del TCP Piccole transazioni

- Con le transazioni con pochi dati, che necessitano di un solo segmento per il trasporto dell'informazione, sono fortemente penalizzate (in presenza di RTT significativi) dagli scambi legati all'apertura e chiusura della connessione.



6.37

### Prestazioni del controllo di flusso del TCP Miglioramenti

- Le modifiche/miglioramenti realizzabili facilmente per il TCP:
  - Uso di uno *stack* protocollare ben realizzato (software ben scritto ed efficiente con hardware ben dimensionato).
  - Uso dell'opzione SACK (*Selective ACK*)
  - Uso di buffer lunghi con l'opzione di scala della finestra
  - Uso del RED e/o della notifica esplicita della congestione
  - Alzare la finestra iniziale a 1 o 2 (o 4)

6.38

### Controllo di congestione: Modelli di Traffico

- Le reti a pacchetto sono solitamente caratterizzate da un livello di complessità molto elevato (rispetto alle rete a commutazione di circuito).
- I processi di pianificazione, dimensionamento e controllo (ad esempio allocazione di banda e controllo d'accesso) richiedono, per essere efficaci, dei modelli in grado di descrivere il comportamento reale della rete anche se in forma approssimata, ma in questo contesto tali modelli non sono ancora del tutto consolidati.
- Nel caso della suite DoD l'elemento più critico risulta essere modellare correttamente il comportamento del controllo di flusso/congestione del TCP, che risulta necessario per determinare
  - le performance *end-to-end*;
  - lo stato della rete.

6.39

### Controllo di congestione: Modelli di Traffico

- Nella Letteratura Scientifica si distinguono tre categorie di modelli:
  - Modelli analitici;
  - Modelli simulativi;
  - Modelli ibridi.
- Nel caso del TCP, i modelli si differenziano anche per il livello di astrazione a cui lavorano:
  - singola connessione TCP / aggregato di connessioni.
  - Modello discreto (pacchetti IP) / Modello continuo (flussi di traffico).
- Possono, inoltre, essere orientati allo studio sia dello sviluppo dinamico del sistema, sia delle performance medie in *steady-point*.

6.40

### Controllo di congestione: Modello Analitico del TCP Reno

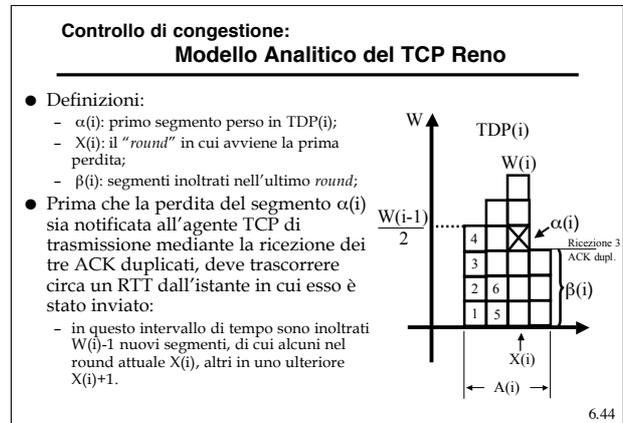
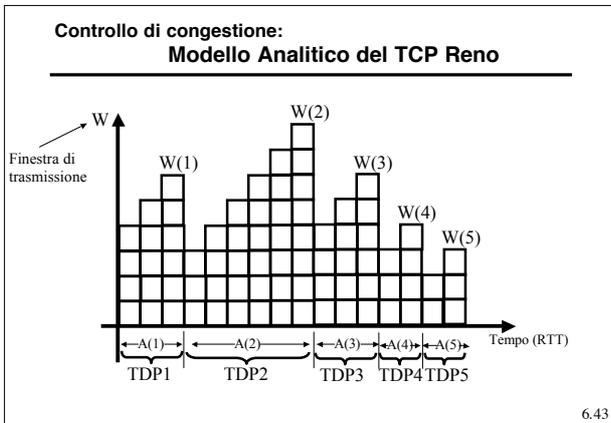
- Consideriamo una connessione TCP Reno.
- Se si ipotizza che
  - la dimensione del file da trasferire è relativamente grande;
  - la probabilità di perdita dei pacchetti è piccola.
 Allora l'effetto della fase di Slow Start è trascurabile.
- Inoltre, se si ipotizza che:
  - La probabilità di perdere più segmenti nella medesima finestra di trasmissione sia trascurabile,
  - La rivelazione delle perdite di segmenti avvenga solo attraverso la ricezione di tre Ack duplicati (e mai, quindi, con lo scadere di un timeout).
 Allora il *throughput* medio della connessione è determinato principalmente dalle dinamiche della *congestion avoidance* e del *fast retransmit/fast recovery*.

6.41

### Controllo di congestione: Modello Analitico del TCP Reno

- Definizioni:
  - TDP(i): L'intervallo compreso tra l'i-esimo e l'i-esimo +1 segmento perso;
  - W(i): massima dimensione della finestra di trasmissione raggiunta in TDP(i);
  - A(i): durata temporale di TDP(i);
 dove i è l'i-esimo segmento perso.

6.42



**Controllo di congestione: Modello Analitico del TCP Reno**

- Quindi in TDP(i) verranno inviati un numero totale di segmenti pari a:
 
$$Y(i) = \alpha(i) + W(i) - 1 \Rightarrow E\{Y\} = E\{\alpha\} + E\{W\} - 1 \quad (*)$$
- Considerando il processo stocastico  $\{\alpha(i)\}$ , si assuma che le v.a.  $\alpha(i)$  siano una sequenza di variabili tra loro indipendenti e identicamente distribuite.
- Quindi, possiamo esprimere la probabilità che  $\alpha(i) = k$  come:
 
$$P\{\alpha = k\} = (1-p)^{k-1} \cdot p \quad k > 1$$

$$\Rightarrow E\{\alpha\} = \sum_{k=1}^{\infty} P\{\alpha = k\} \cdot k = \sum_{k=1}^{\infty} (1-p)^{k-1} \cdot p \cdot k = \frac{1}{p}$$
- Sostituendo nella (\*), si ottiene:
 
$$\Rightarrow E\{Y\} = \frac{1-p}{p} + E\{W\} \quad (**)$$

6.45

**Controllo di congestione: Modello Analitico del TCP Reno**

- Si introduce la v.a.  $r(i,j)$  come la durata del j-esimo round in TDP(i).
 
$$A(i) = \sum_{j=1}^{X(i)+1} r_{ij}$$
- Ipotizzando che le  $r(i,j)$  siano i.i.d. ed indipendenti dal valore di  $X(i)$ , si ricava:
 
$$E\{A\} = (E\{X\} + 1) \cdot E\{r\}$$
 dove  $E\{r\} = E\{t_{ij}\} = RTT$
- $W(i)$  è, invece, esprimibile come:
 
$$W(i) = \frac{W(i-1)}{2} + X(i)$$

$$E\{W\} = \frac{1}{2}E\{W\} + E\{X\} \Rightarrow E\{W\} = 2E\{X\}$$

6.46

**Controllo di congestione: Modello Analitico del TCP Reno**

- Riscrivendo  $Y(i)$ , si ottiene:
 
$$Y(i) = \sum_{k=0}^{X(i)} \left( \frac{W(i-1)}{2} + k \right) + \beta(i) = \frac{X(i)}{2} \left( \frac{W(i-1)}{2} + W(i-1) \right) + \beta(i)$$
- Sostituendo il tutto nella (\*\*):
 
$$\frac{1-p}{p} + E\{W\} = \frac{E\{X\}}{2} \left( \frac{E\{W\}}{2} + E\{W\} - 1 \right) + E\{\beta\}$$

$$\Rightarrow \frac{3}{8}E\{W\}^2 - \frac{5}{4}E\{W\} + E\{\beta\} + \frac{p-1}{p} = 0$$
- Ipotizzando che  $\beta(i)$  sia uniformemente distribuita in  $[1, W(i)-1]$ , allora:  $E\{\beta\} = \frac{1}{2}E\{W\}$
- Quindi:
 
$$\frac{3}{8}E\{W\}^2 - \frac{3}{4}E\{W\} + \frac{p-1}{p}$$

6.47

**Controllo di congestione: Modello Analitico del TCP Reno**

- Risolvendo:
 

Dimensione media della finestra massima	$E\{W\} = 1 + \sqrt{1 + \frac{8}{3} \cdot \frac{1-p}{p}}$
Numero medio "RTT" prima che avvenga la perdita	$E\{X\} = \frac{1}{2} + \sqrt{\frac{2(1-p)}{3p} + \frac{1}{4}}$
Durata media del TDP	$E\{A\} = RTT \cdot \left( \frac{4}{3} + \sqrt{\frac{2(1-p)}{3p} + \frac{1}{4}} \right)$

6.48

**Controllo di congestione:  
Modello Analitico del TCP Reno**

● Risolvendo:

Tasso Medio della connessione TCP

$$B(p) = \frac{E\{Y\}}{E\{A\}}$$

$$\Rightarrow B(p) = \frac{1}{RTT} \cdot \frac{\frac{1}{p} + \sqrt{\frac{8 \cdot (1-p)}{3} + 1}}{\frac{3}{2} + \sqrt{\frac{2 \cdot (1-p)}{3} + \frac{1}{4}}} \Rightarrow B(p) \xrightarrow{p \rightarrow 0} \frac{1}{RTT} \sqrt{\frac{3}{2p}} + o\left(\frac{1}{\sqrt{p}}\right)$$

6.49

**ATM: Controllo di flusso per il traffico ABR  
End to End Rate based flow Control (EERC)**

- Il tasso istantaneo della sorgente viene chiamato *Allowable Cell Rate (ACR)*
- La sorgente inizia a trasmettere con un tasso pari a *ACR = ICR (Initial Cell Rate)* ed invia nelle celle FRM un tasso desiderato *ER (Explicit Rate)*
- Ogni nodo attraversato dalle celle RM (in entrambe le direzioni compresa la destinazione), può modificare il valore di ER; in particolare, se calcola una banda disponibile minore all'ER ricevuto ne sostituisce il valore con questa.

6.50

**ATM: Controllo di flusso per il traffico ABR  
End to End Rate based flow Control (EERC)**

● Quando la sorgente riceve una cella RM modifica il proprio tasso come segue:

**Minimum Cell Rate**  
Tasso minimo assicurato all'instaurazione della connessione

In sostanza: se  $ER > ACR$  incremento il tasso in modo additivo, mantenendolo però sempre  $\leq ER$

**Peak Cell Rate**  
Massimo tasso concesso alla connessione

**Rate Increase Factor**  
Valore  $< 1$ , in percentuale rispetto al tasso massimo

$$ACR_{i+1} = \max\{MCR, \min\{ER, PCR, ACR_i + RIF \times PCR\}\}$$

6.51

**ATM: Controllo di flusso per il traffico ABR  
End to End Rate based flow Control (EERC)**

- Per mettere la presenza di nodi che non siano in grado di segnalare esplicitamente il tasso permesso, sono presenti due bit aggiuntivi:  
**NI = Non Increase; CI = Congestion Indication** che i nodi possono portare a 1 (ma non a zero) per segnalare condizioni di sovraccarico.
- In particolare:
  - Se  $CI = 1$ :  $ACR_{i+1} = \max\{MCR, \min\{ER, ACR_i (1 - RDF)\}\}$  dove  $RDF = Rate Decrease Factor$ ; in sostanza si opera un decremento moltiplicativo.
  - Se  $NI = 1$   $ACR_{i+1} = \max\{MCR, \min\{ER, ACR_i\}\}$  ossia si mantiene il tasso  $\leq ACR_i$ , dove il  $<$  interviene se lo suggerisce ER.

6.52

**ATM: Controllo di flusso per il traffico ABR  
End to End Rate based flow Control (EERC)**

- Nel caso un nodo non gestisca la RM, può portare a 1 il bit di *Explicit Forward Congestion Indication (EFCI)* nell'intestazione di una cella normale del flusso
- Se l'ultima cella del flusso ricevuta aveva  $EFCI = 1$ , la destinazione porrà il bit  $CI = 1$  delle successiva cella RM generata.
- Per accelerare la velocità di retroazione, un nodo lungo il percorso può generare direttamente un BRM.

6.53

**ATM: Controllo di flusso per il traffico ABR  
End to End Rate based flow Control (EERC)**

- Ci sono ancora due situazioni da considerare:
  - Cosa succede se la sorgente non usa il flusso (non trasmette per un certo tempo)?
    - » Se la sorgente rimane ferma per un tempo minimo stabilito (default = 500 ms), pone  $ACR = ICR$ .
  - Cosa succede se, per qualche problema, la sorgente non riceve RM per un certo tempo?
    - » In questo caso decrementa in modo moltiplicativo la propria cella.

6.54

**ATM: Controllo di flusso per il traffico ABR**  
***End to End Rate based flow Control (EERC)***

---

● Limiti di questo approccio:

- Il flusso delle celle di controllo è mescolato a quello dati (nello stesso VC) e quindi ci vuole un hardware complesso perché ogni nodo possa estrarre, elaborare e ritrasmettere le RM.
- Il tasso della sorgente deve essere modificato (potenzialmente) con la frequenza di arrivo delle RM, quindi molto frequentemente (complesso).
- Lo standard presuppone che le celle siano esattamente spaziate di  $1/ACR$ , se ci sono più flussi attivi fra una coppia sorg.- dest. Devono essere "*schedulati*" opportunamente (complesso da fare) .
- L'incremento additivo limita l'efficienza e rende critico il valore dell'incremento, ma è lo scotto da pagare per avere compatibilità con il meccanismo più semplice (CI).
- La sorgente deve comunque essere controllata (misure e *policing* all'ingresso)

6.55