



Università di Bergamo

*Dipartimento di Ingegneria dell'Informazione e
Metodi Matematici*

Reti di Telecomunicazione

Prof. Fabio Martignon



Università di Bergamo

*Dipartimento di Ingegneria dell'Informazione e
Metodi Matematici*

7 – Controllo di Flusso e di Congestione

Reti di Telecomunicazione

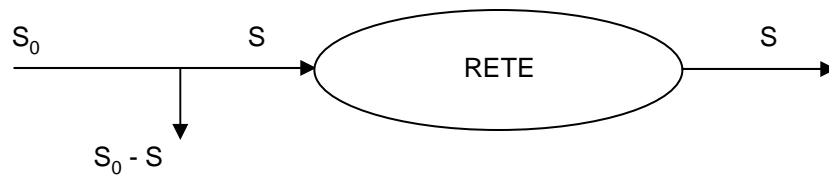
Il Problema della Congestione

- Una rete ideale smaltisce tutto il traffico offerto S_0 fino al raggiungimento della capacità di rete C .
- Se $S_0 > C$ il traffico in eccesso deve essere rifiutato.
- Nelle Reti a commutazione di circuito
 $S = C$ la capacità è data dal numero di circuiti assegnabili
- Nelle Reti a commutazione di pacchetto
 $S < C$ $S = C$ si avrebbe per $\rho = 1 \Rightarrow$ code infinite

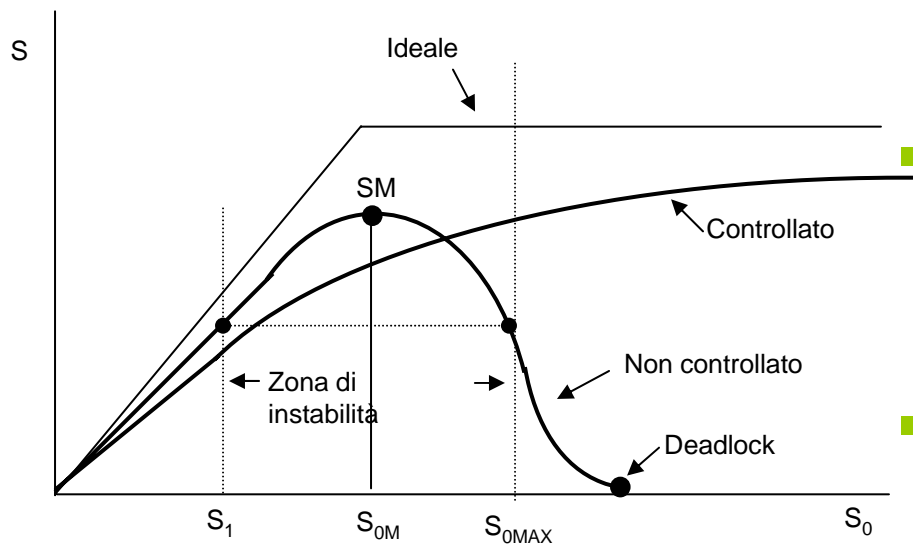
Il Problema della Congestione

- Il fenomeno della congestione si verifica quando all'aumentare di S_0 , S (il traffico smaltito), cala
- Possibili cause:
 - ✓ Cattivo utilizzo delle risorse (buffer dei canali di trasmissione, capacità di elaborazione)
 - ✓ Variazioni nella matrice di traffico usata nel dimensionamento
 - ✓ Variazioni nella topologia (guasti di nodi e canali)
- La saturazione di un nodo provoca trabocco sui nodi vicini con conseguente possibilità di saturazione.
- Il fenomeno della congestione è intrinseco nelle reti a commutazione di pacchetto a causa della moltiplicazione statistica.

Il Problema della Congestione



Relazione tra traffico offerto e traffico smaltito



■ Traffico perso in S_1 (rispetto all'Ideale) a causa di fluttuazioni statistiche del ritardo, code finite.

■ Se il traffico rifiutato viene accodato e se non si esercita controllo della congestione la rete non può funzionare in regime stazionario con $S > S(S_{0MAX})$

■ Anche se $S_0 < S_{0M}$ a causa di fluttuazioni statistiche S_0 può superare S_{0M} ed innescare una reazione positiva: al crescere di S_0 , S cala.

Esempio: incrocio e semaforo

Scopo del controllo della congestione

- **Prevenire la degradazione del traffico smaltito S**
- **Evitare situazioni di dead-lock**
- **Garantire utilizzo equo delle risorse**

Esempi di effetti della Congestione

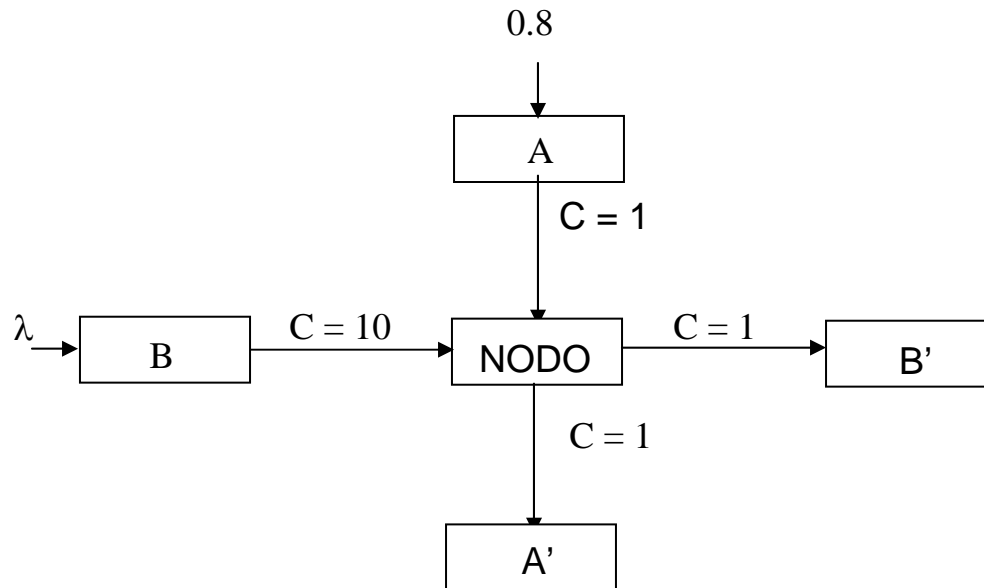
■ Perdita di Efficienza

Spreco di risorse

- ✓ Capacità canali
- ✓ Memoria nodi
- ✓ Si assegnano risorse in eccesso a discapito di altre comunicazioni (Per es. un terminale lento è collegato ad un canale veloce. In arrivo ci sono code che intasano la rete.)

Esempi di effetti della Congestione

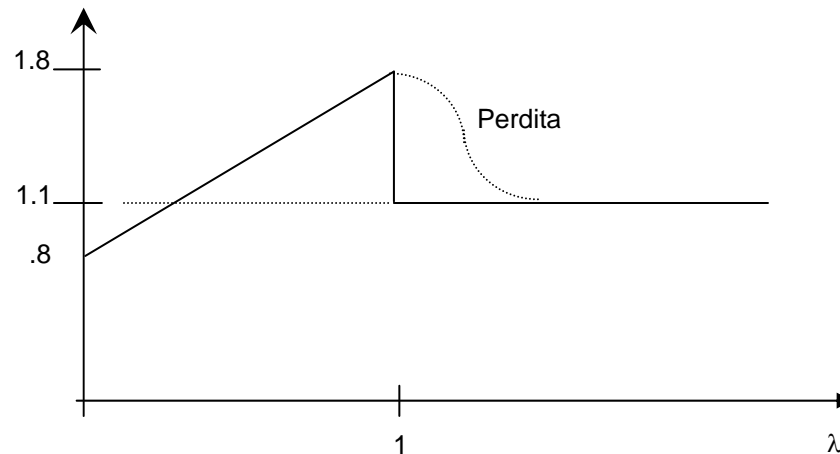
Perdita di Efficienza



- Se λ tende ad 1, le code in uscita verso B' tendono ad ∞ . Il NODO scarta pacchetti causa buffer finito.
- Supponiamo che i pacchetti scartati vengano poi ritrasmessi dalle rispettive sorgenti

Esempi di effetti della Congestione

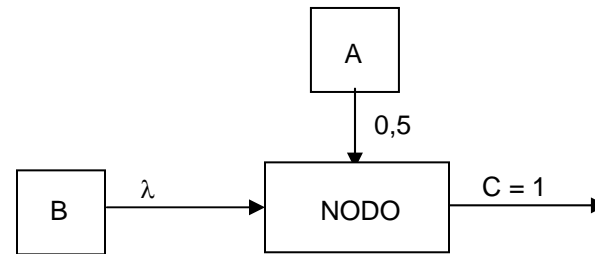
Perdita di Efficienza



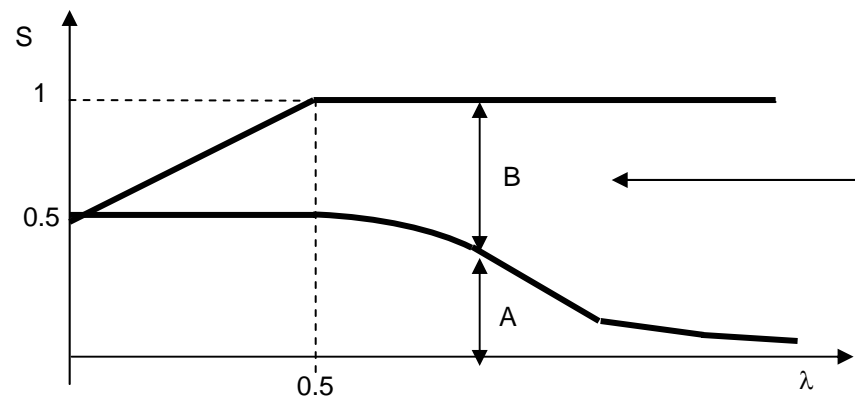
- Il nodo smaltisce tutto il traffico finché $\lambda < 1$
- A causa della differenza di capacità delle linee in ingresso, B può offrire un traffico 10 volte quello di A.
- Quando si libera un buffer nel NODO, la probabilità che lo occupi un pacchetto di B è 10 volte maggiore di A.
- Il traffico smaltito da B-B' è 1 quindi A-A' tende a 0.1.
- Totale traffico smaltito 1.1
- *Controllo della congestione: politica di assegnazione dei buffer nel nodo.*

Esempi di effetti della Congestione

Non Equità (Unfairness)



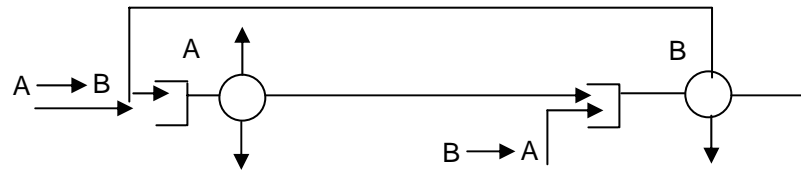
- **Certi utenti ottengono più risorse di altri**
- **In una rete la definizione di equità non è semplice.**



Qui non ho degradazione delle performance *complessive*, ma B ottiene tutta la capacità del canale, tagliando fuori A

Esempi di effetti della Congestione

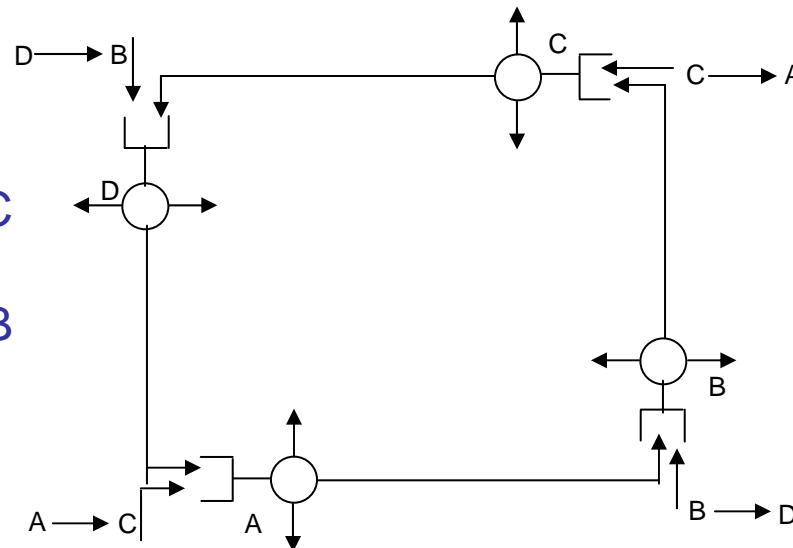
Deadlock



Diretto

A: buffer saturo di trasmissioni verso B

B: buffer saturo di trasmissioni verso A



Indiretto

A: buffer saturo di trasmissioni verso C

.....

D: buffer saturo di trasmissioni verso B

Funzioni del Controllo di Flusso

- Insieme di procedure che proteggono la rete da sovraccarichi e dead-lock
- Agiscono su:
 - ✓ Allocazione buffer nodi
 - ✓ Blocco traffico nella rete
 - ✓ Blocco traffico in ingresso
- Realizzazione del controllo a vari livelli
- Il controllo usa risorse di rete: Trade-off

Costo  **Benefici**

Prestazioni di Algoritmi di Controllo di Flusso

■ Prestazioni:

- ✓ Criteri per ottimizzare i parametri
- ✓ Valutazione costo-benefici
- ✓ Confrontare soluzioni alternative

Traffico smaltito/Traffico offerto

- ✓ Carichi critici
- ✓ Efficienza del controllo

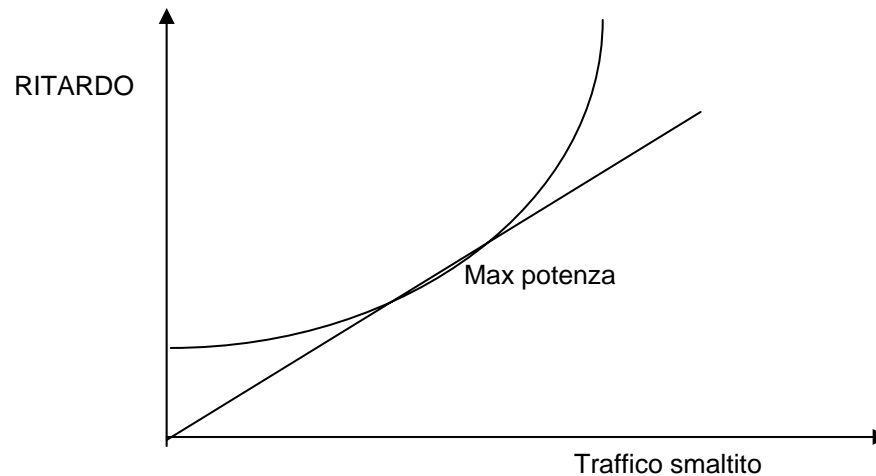
Ritardo/Traffico smaltito

- ✓ Ritardo introdotto dal controllo
- ✓ Ad elevato traffico possono corrispondere ritardi intollerabili
- ✓ Elevati ritardi anche a basso traffico

Formulazione Problema

- **Definizione:**

$$Potenza = \frac{\text{Traffico smaltito}}{\text{Ritardo}}$$



Osservazione: per sistemi $M|M|1$, e più in generale $M|G|1$, si ottiene il massimo della potenza per $N_{\text{medio}}=1$

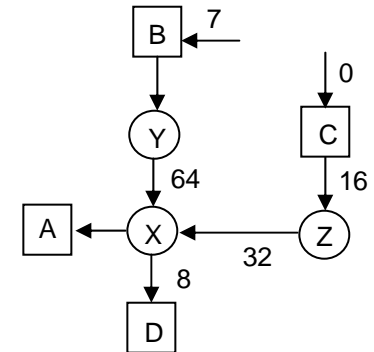
Esempio

- **La rete è non controllata:**
 - ✓ **Ogni traffico può accedere a tutte le risorse**
 - ✓ **Non ci sono limiti al traffico offerto**
 - ✓ **Si possono perdere pacchetti per saturazione dei buffer nei vari nodi**
 - ✓ **I pacchetti persi vengono ritrasmessi dai nodi immediatamente a monte della perdita**

Esempio

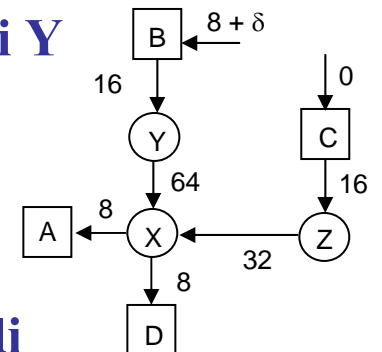
- **Scenario (a):**

- ✓ La rete non presenta situazioni di congestione



- **Scenario (b):**

- ✓ Il traffico offerto è maggiore di quello che può essere consegnato alla destinazione. Il buffer di X si satura. Pacchetti vengono scartati. Y li ritrasmette. Il buffer di Y si riempie. Il traffico in $Y \rightarrow X$ aumenta causa ritrasmissioni. Spreco di risorse



- **Soluzione:**

- ✓ Aumentare la capacità di $X \rightarrow A$ (da realizzare in fase di progetto)
- ✓ Limitare il traffico generato da B (può funzionare in tempo reale ed intervenire solo quando necessario).

Esempio

■ Scenario (c):

- ✓ La rete non presenta situazioni di congestione. Totale traffico smaltito $7 + 7 = 14$

■ Scenario (d):

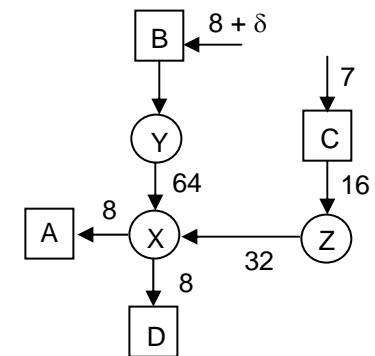
- ✓ La rete ha sufficienti risorse per gestire il traffico $C \rightarrow D$ ma a causa del traffico in eccesso $B \rightarrow A$ si ha congestione come in (b)
- ✓ Quando si libera un posto in X entra un pacchetto da B o C. Essendo la velocità da Y doppia, è doppia la percentuale di pacchetti $B \rightarrow A$ rispetto a $C \rightarrow D$. Poiché il traffico smaltito $B \rightarrow A = 8$ sarà $C \rightarrow D = 4$. Traffico totale 12.

■ Conseguenze:

- ✓ Riduzione di traffico smaltito $14 \rightarrow 12$
- ✓ La relazione di traffico $C \rightarrow D$ è trattata inequamente

■ Soluzione:

- ✓ Come in (b)
- ✓ Riservare buffer in X per il traffico $C \rightarrow D$



Controllo di Flusso e della Congestione

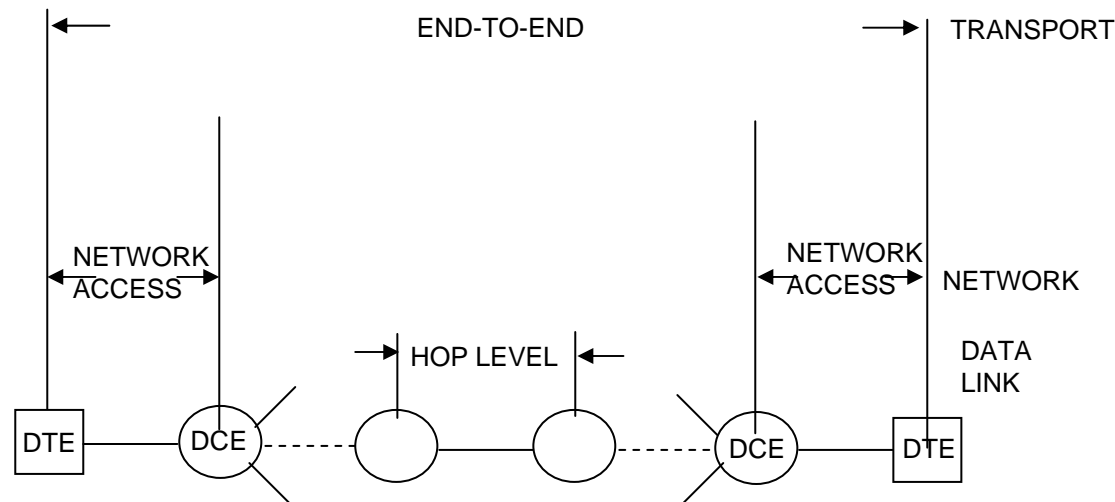
- Le procedure di *controllo di flusso* sono tipicamente utilizzate per garantire che la sorgente non inondi una destinazione con più traffico di quanto essa possa ricevere (Speed matching).
 - ✓ Generalmente è realizzato end-to-end.
 - ✓ Hop-by-hop se il controllo di flusso è esercitato fra due nodi consecutivi nel cammino fra sorgente e destinazione.
- Le procedure di *controllo della congestione* sono tipicamente utilizzate per garantire che alla rete nella sua globalità non sia richiesto di trasmettere un traffico maggiore della sua capacità.

Tali procedure possono essere realizzate all'accesso in rete per evitare che un host invii troppo traffico, oppure agire sull'allocazione delle memorie per garantire fairness e assenza di dead lock.
- In questo corso non faremo distinzione fra le due terminologie.

Livelli di controllo della congestione

Le tecniche di controllo possono essere realizzate ad ogni livello dell'architettura a strati

- ✓ Internodo (Hop Level): congestioni locali buffer di linea
- ✓ Accesso alla rete (Network Access): congestione nella rete di accesso
- ✓ End-to-End (Transport): congestione nel nodo destinazione



Controllo di Flusso Hop-by-Hop

- Agisce localmente controllando code e occupazione buffer
- Se si superano soglie prefissate vengono scartati pacchetti

Channel Queue Limit (CQL)

- Distingue i pacchetti a seconda dei canali in uscita
- Limite massimo alla coda in uscita
- Previene il deadlock diretto dato che garantisce la non saturazione delle memorie

Channel Queue Limit

- **Esistono varie versioni di CQL:**

Partizione completa

- **N = numero code in uscita**
- **n_i = numero pacchetti nella coda i**
- **B = dimensione del buffer**
- **$0 \leq n_i \leq \frac{B}{N}$**

Partizione limitata

- **b_M = massima dimensione di coda**
- **$b_M > \frac{B}{N}$**
- **$0 \leq n_i \leq b_M$**
- **$\sum_i n_i \leq B$**

Channel Queue Limit

Partizione con buffer minimo

- $b_m =$ minimo buffer garantito ad ogni coda
- $b_m < \frac{B}{N}$
- $\sum_i \max(0, n_i - b_m) \leq B - N b_m$

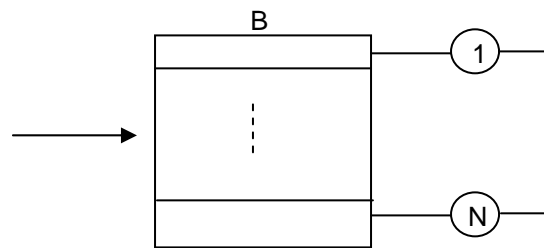
Partizione limitata con buffer minimo

- I parametri delle diverse versioni possono cambiare dinamicamente da coda a coda in funzione del traffico.

Analisi Prestazioni CQL

- Modelli CQL

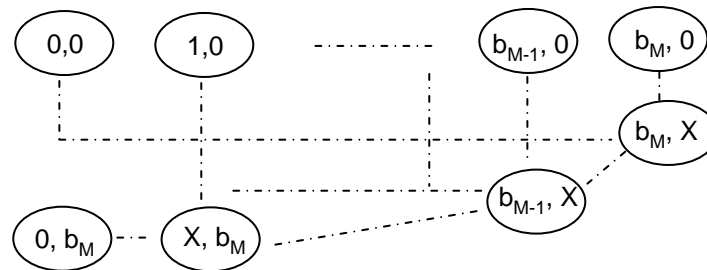
Partizione completa



$$M|M|1|C$$

$$C = \frac{B}{N}$$

Partizione limitata (Es. $N = 2$)

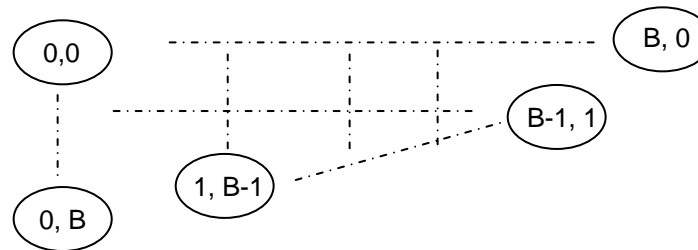


$$b_M + X = B$$

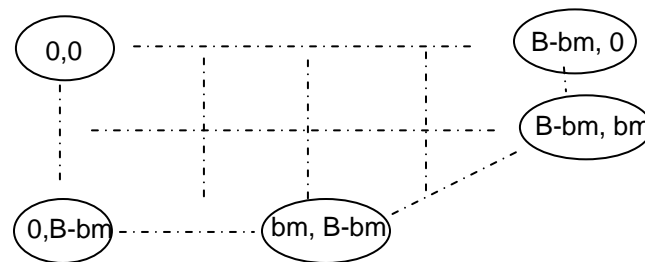
Analisi Prestazioni CQL

- Modelli CQL

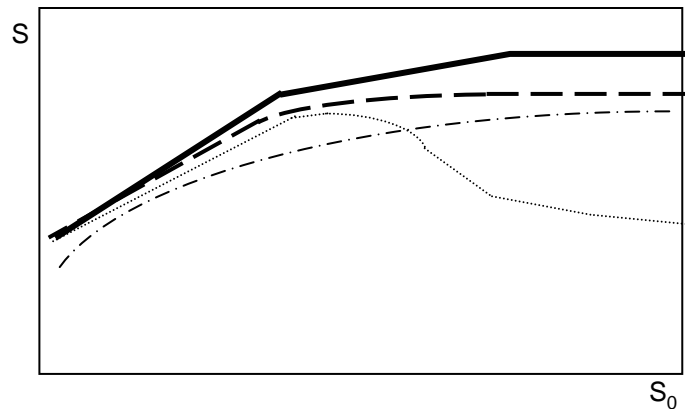
Partizione non controllata



Partizione con buffer minimo (bm)



Prestazioni Qualitative per un Singolo Nodo

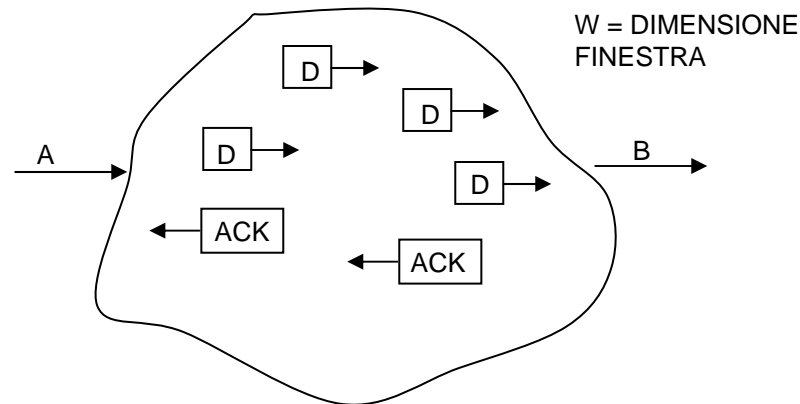


- Partizione non controllata
Assenza di controllo di flusso
- - - - - Partizione completa (no sharing)
- Partizione limitata dinamica
Ottimizzazione di b_M in funzione traffico offerto
- . - . - Euristicia radice quadrata

$$b_M = \sqrt{B/N}$$

Controllo del Flusso End-to-End

- Tipicamente viene implementato utilizzando un controllo di flusso a Finestra Scorrevole (Sliding Window)
- La dimensione della finestra impone un limite al numero delle trame trasmesse da A verso B, ma non ancora confermate
- A può trasmettere una nuova trama solo dopo aver ricevuto un riscontro (ACK).

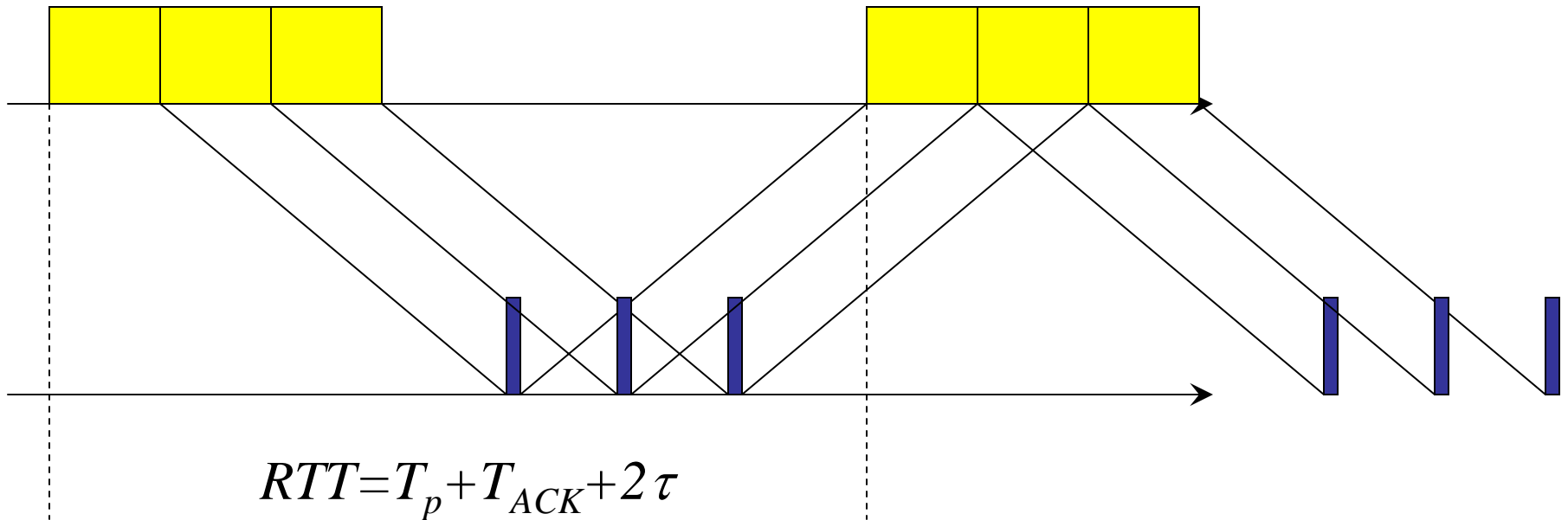


Controllo a Finestra

- La funzione del controllo a finestra è duplice:
 - Limitare il flusso di pacchetti che arrivano alla destinazione (controllo di *flusso*)
 - Limitare il flusso di pacchetti offerti ai sottosistemi intermedi (nodi, canali) (controllo di *congestione*)
- Parametro critico: Dimensione finestra W

Controllo a Finestra

- La dimensione della finestra può limitare la velocità di invio dei pacchetti anche quando i riscontri vengono trasmessi immediatamente dopo la ricezione del pacchetto
- Nel caso di singolo link:

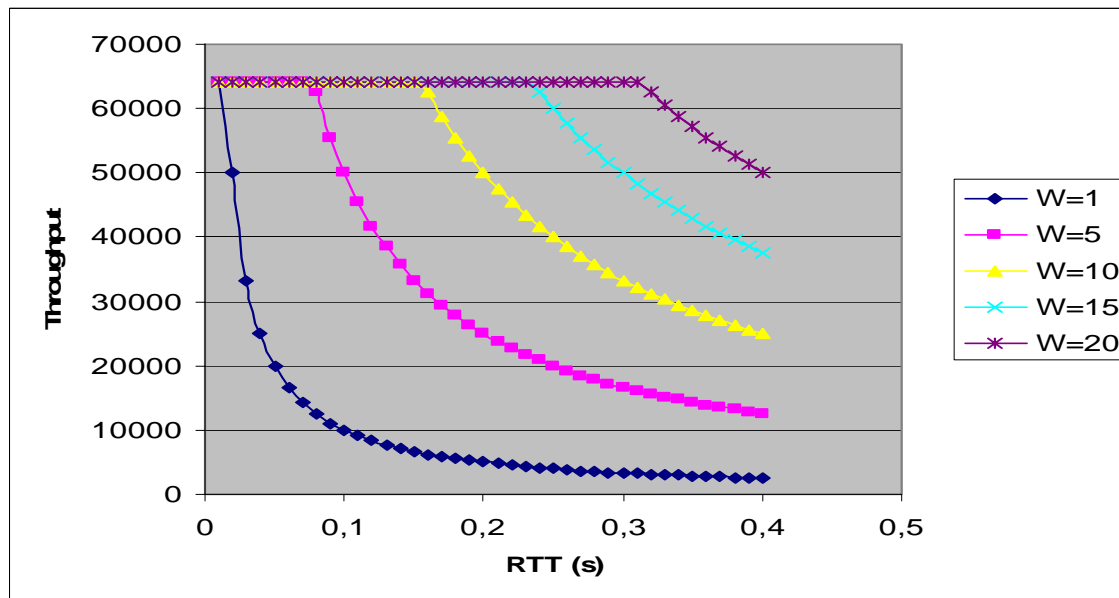


Controllo a Finestra

- la velocità di trasmissione risulta limitata da:

$$c_{eff} = \min \left\{ c, \frac{W \cdot L}{RTT} \right\}$$

- dove c è la capacità del link ed L la lunghezza dei pacchetti



L=1000 bits
c=64 kb/s

Controllo a Finestra

- affinché la finestra non sia limitante occorre che:

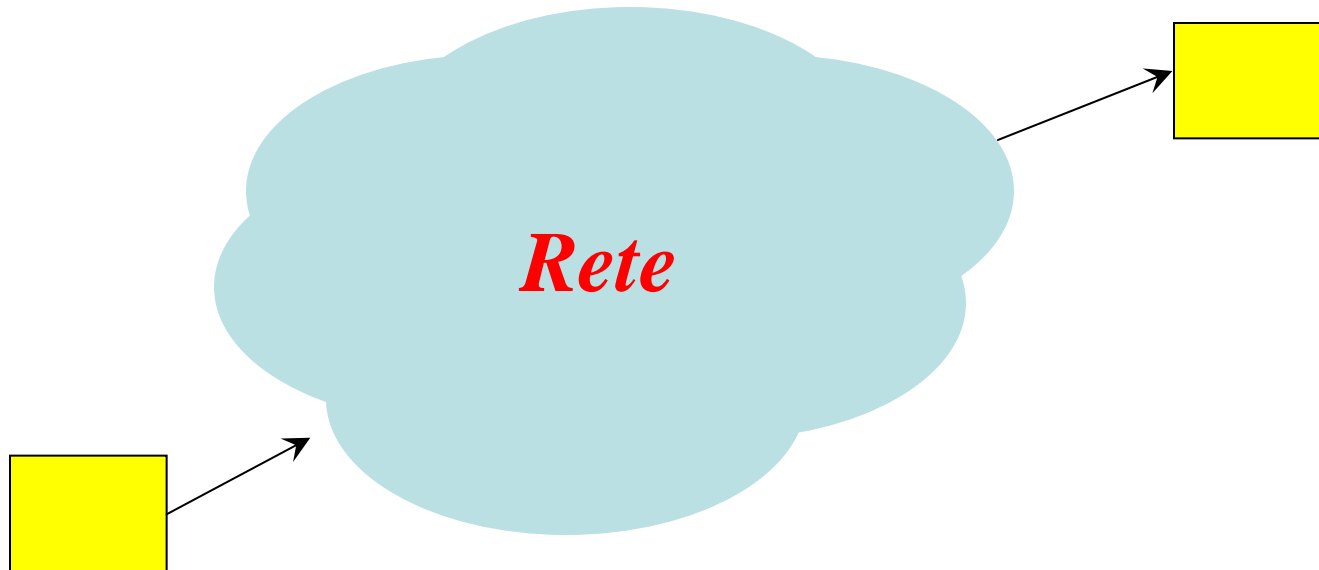
$$W \cdot L \geq c \cdot RTT$$

- che prende il nome di prodotto banda-ritardo

Controllo di congestione a finestra

■ Approccio end-to-end

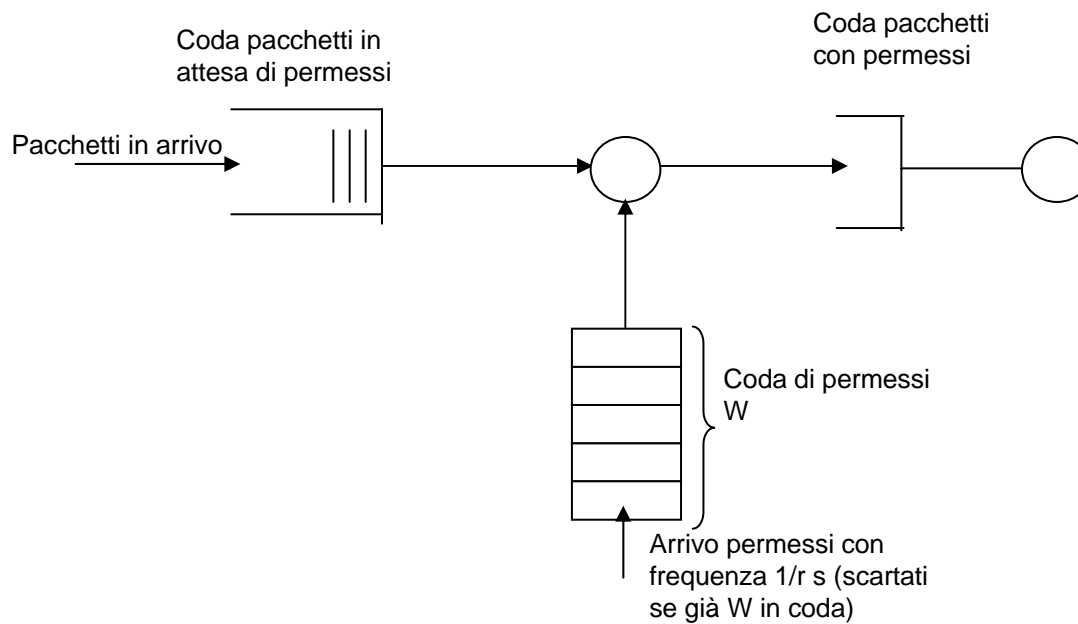
- se la rete va in congestione aumenta il ritardo (RTT) e diminuisce la velocità
- in questo caso nel valore di RTT sono compresi anche i tempi di accodamento nei nodi intermedi
- molti approcci sono più sofisticati e variano la finestra dinamicamente in base alle condizioni della rete



Rate Control

- **Il Rate Control viene utilizzato in genere nei punti di accesso alla rete**
- **In ambienti di rete in cui l'utente contratta con il Service Provider una banda garantita, il gestore della rete verifica che il traffico offerto sia conforme al Rate di trasmissione contrattato**
- **In genere viene garantito un ritmo medio di trasmissione, cercando inoltre di accomodare “burst” istantanei di traffico**

Token Bucket



Token Bucket

- Regola il traffico in ingresso alla rete, ma non garantisce ritardo o perdita.
- Se W è piccolo, aumenta l'attesa per permessi
- Se $W = 1$, il comportamento è simile alla multiplazione a divisione di tempo (Leaky bucket)
- Se W è grande, si permette l'ingresso immediato di burst in rete (possibilità di congestione e perdita)
- Un'opportuna scelta dei parametri riduce la saturazione dei buffer e il ritardo massimo.
- E' possibile eseguire un aggiustamento dinamico di W (e/o r) sulla base di informazioni di controllo (per es. *choke packet* come nel controllo a finestra).
- Esegue un efficace controllo del ritardo.

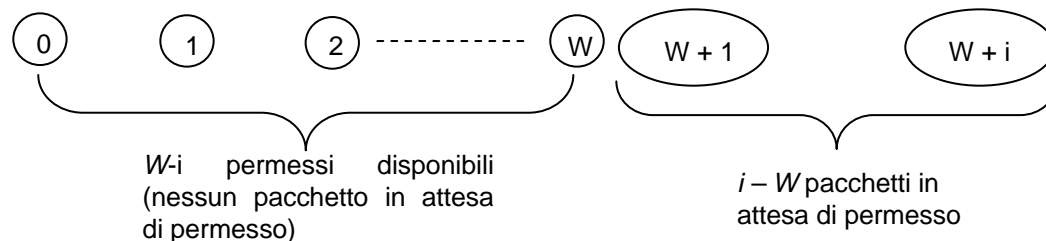
Analisi del Token Bucket

■ Assunzioni

- Processo degli arrivi dei pacchetti Poissoniano di parametro λ ,
- I permessi arrivano ogni $1/r$ secondi e, se già W permessi sono in coda, vengono scartati

■ Modello

- Catena di Markov a tempo discreto



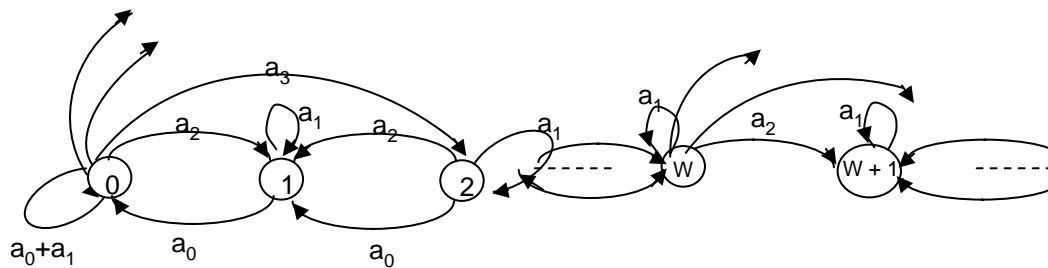
Analisi del Token Bucket

- Le transizioni avvengono negli istanti di arrivo dei permessi, ovvero ogni $1/r$ secondi:

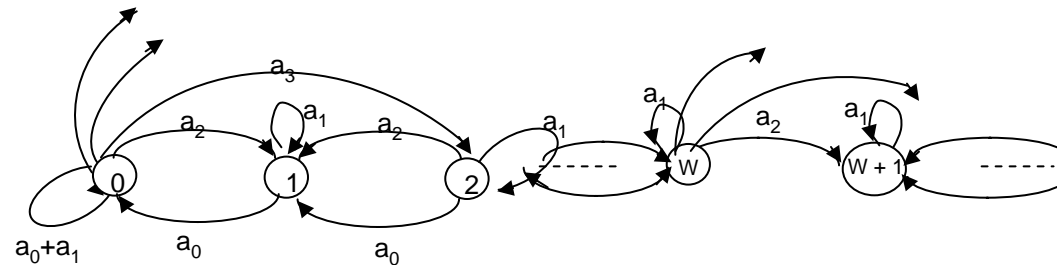
- $P(\text{ci siano } k \text{ arrivi di } p \text{ in } 1/r) = a_k = \frac{(\lambda/r)^k}{k!} e^{-\lambda/r}$

$$P_{0i} = \begin{cases} a_{i+1} & \text{se } i \geq 1 \\ a_0 + a_1 & \text{se } i = 0 \end{cases}$$

$$P_{ji} = \begin{cases} a_{i-j+1} & \text{se } j \leq i+1 \\ 0 & \text{altrimenti} \end{cases} \quad \text{per } j \geq 1$$



Analisi del Token Bucket



■ Soluzione ricorsiva

$$p_0 \cdot 1 = (a_0 + a_1)p_0 + a_0 p_1$$

$$p_i \cdot 1 = \sum_{j=0}^{i+1} a_{i-j+1} \cdot p_j \quad i \geq 1$$

$$\text{Nodo 0} \quad \left\{ \begin{array}{l} p_1 = \frac{(1 - a_0 - a_1)}{a_0} p_0 \end{array} \right.$$

$$\text{Nodo 1} \quad \left\{ \begin{array}{l} p_1 \cdot 1 = a_2 p_0 + a_1 p_1 + a_0 p_2 \\ p_2 = \frac{p_0}{a_2} \left(\frac{(1 - a_0 - a_1)(1 - a_1)}{a_0} - a_2 \right) \end{array} \right.$$

Analisi del Token Bucket

- Calcolo di $p_0 \longrightarrow \Sigma p_i = 1$
 - Oppure più semplicemente:
 - Frequenza media con cui i permessi entrano nel sistema: $(1-p_0 a_0) r$
 - Frequenza media pacchetti in arrivo: λ
 - Condizioni stazionarie $\lambda = (1-p_0 a_0) r$
- Condizione di stabilità $\lambda < r$

Nota: ipotizzo che la coda dei pacchetti abbia capacità infinita

$$p_0 = \frac{r - \lambda}{r a_0}$$

Analisi del Token Bucket

- **Tempo medio di attesa per ottenere un permesso**

$$T = \sum_{j=W+1}^{\infty} \frac{1}{r} (j - W) p_j$$

- **Nel caso reale gli arrivi possono non essere Poissoniani, ma avvenire a burst.**