



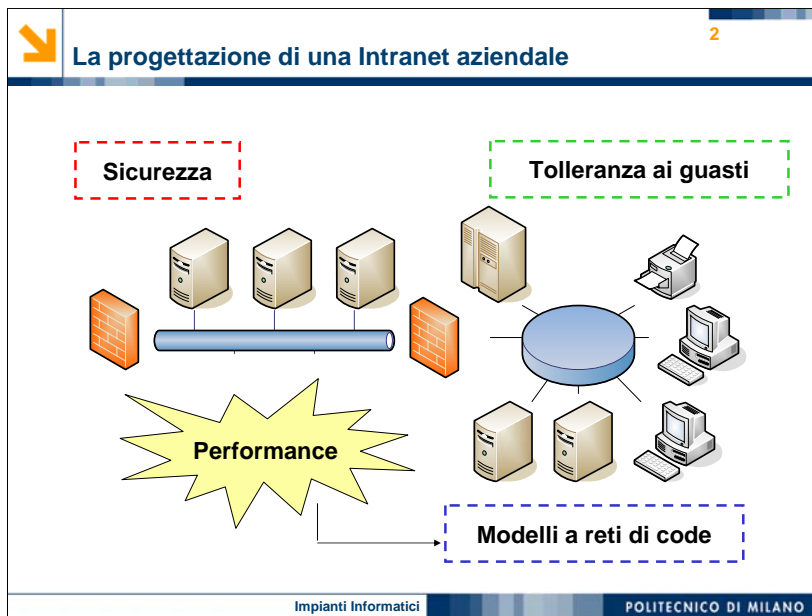
*Paolo Cremonesi*

# Impianti Informatici

 POLITECNICO DI MILANO

 **Modelli a rete di code:  
concetti base**

- classi di carico
- server
- visite
- utilizzo



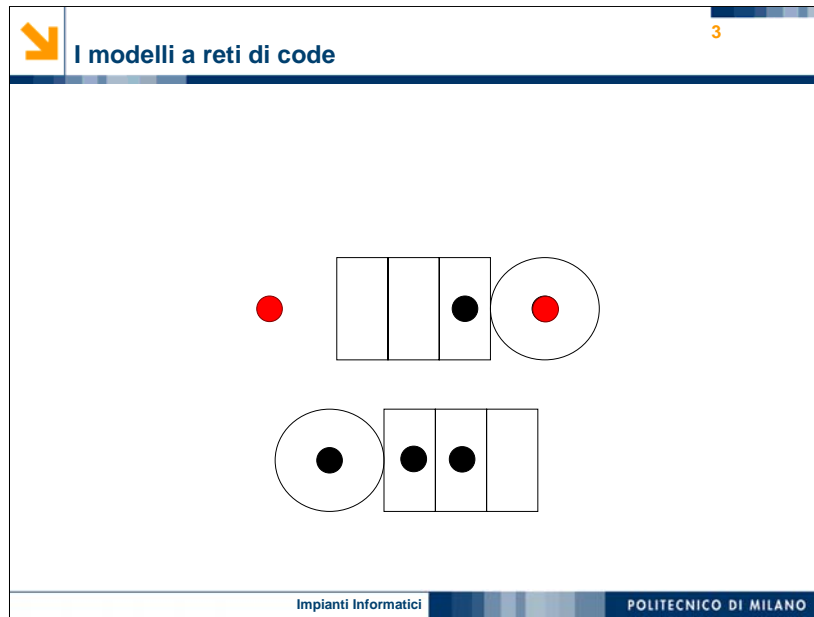
La progettazione di una Intranet aziendale e' un'attivita' complessa, che richiede di scegliere la migliore architettura possibile sulla base delle tecnologie hardware e software disponibili al momento sul mercato.

Esistono diversi criteri possibili per decidere se un'architettura e' migliore di un'altra. Solitamente la valutazione avviene sulla base di alcuni criteri: quali

- i livelli di sicurezza forniti dal sistema,
- la sua tolleranza ai guasti,
- la qualità, in termini di prestazioni, del servizio erogato.

In particolare, quest'ultimo tipo di valutazione richiede l'impiego di modelli quantitativi che consentano di stabilire se l'impianto potra' servire le richieste con sufficiente rapidita'.

I modelli piu' popolari che permettono di studiare questi problemi sono noti come i modelli a reti di code.

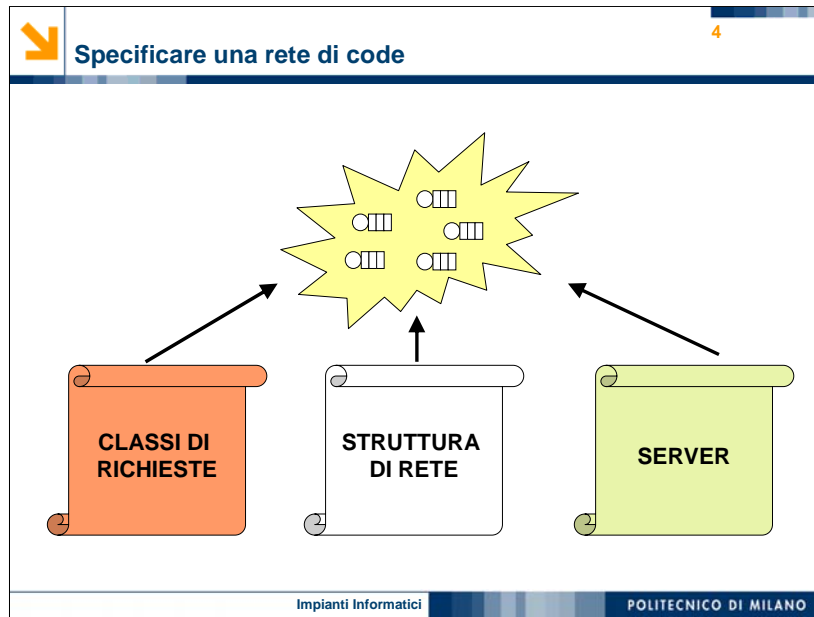


Come suggerito dal nome, in un rete di code ciascun server e' rappresentato come una coda

- dove ciascuna richiesta arriva
- attende di essere servita mentre il server processa le richieste già presenti in coda
- riceve il servizio dal servente INSERIRE CLIPART OROLOGIO
- e quindi ritorna sotto forma di risposta all'utente che le ha generate

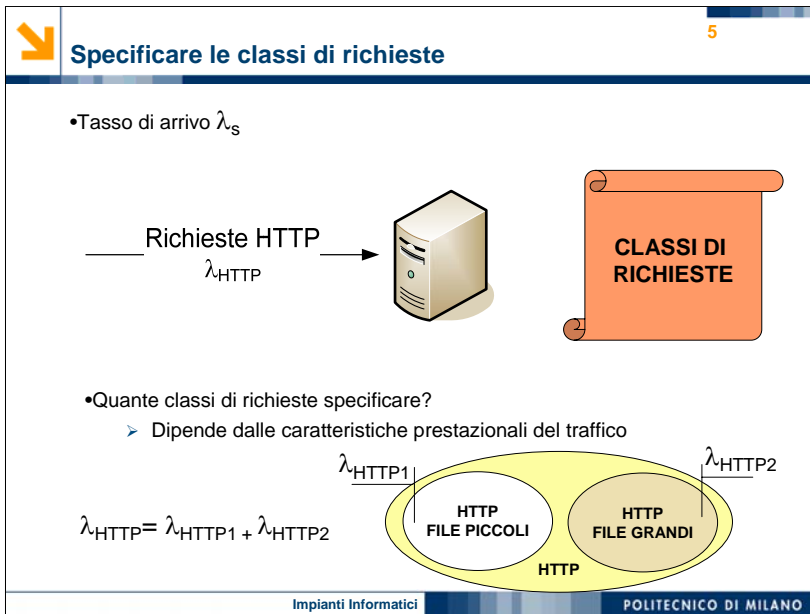
In alternativa le richieste possono non tornare immediatamente all'utente,

- ma proseguire verso un altro server, dove andranno nuovamente ad accodarsi in attesa di essere servite.
- Quest'ultima caratteristica dei modelli a reti di code permette di rappresentare anche il comportamento di richieste complesse che necessitino di essere serviti da diverse macchine.



Un modello a reti di code viene specificato attraverso diversi parametri, classificabili in tre gruppi:

- parametri che descrivono le diverse classi di richieste che giungono all'impianto, che corrispondono in prima approssimazione ai diversi tipi di servizi offerti dai server del nostro impianto
- parametri che specificano la struttura delle rete e il modo in cui fluiscono le richieste delle diverse classi al proprio interno
- parametri che descrivono il comportamento dei server all'atto di processare le richieste delle diverse classi



Cominciamo considerando i parametri che descrivono le richieste. In una rete di code è sufficiente indicare per ogni classe di richieste  $s$ , il suo tasso di arrivo medio al sistema.

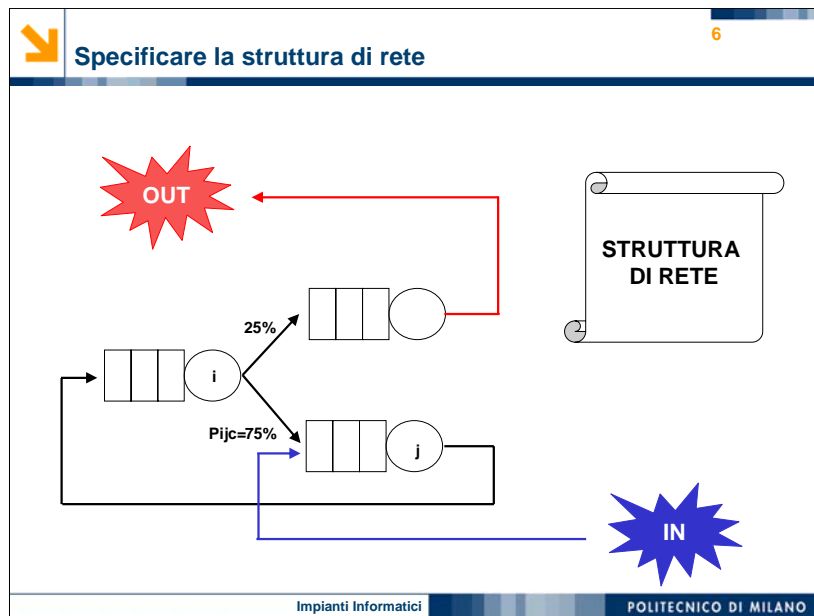
- Tale tasso è indicato dal simbolo  $\lambda_s$ .

- 

- Ad esempio, se un impianto Web riceve 10 richieste HTTP al secondo, allora avremo  $\lambda_{\text{HTTP}} = 10$  richieste/s.

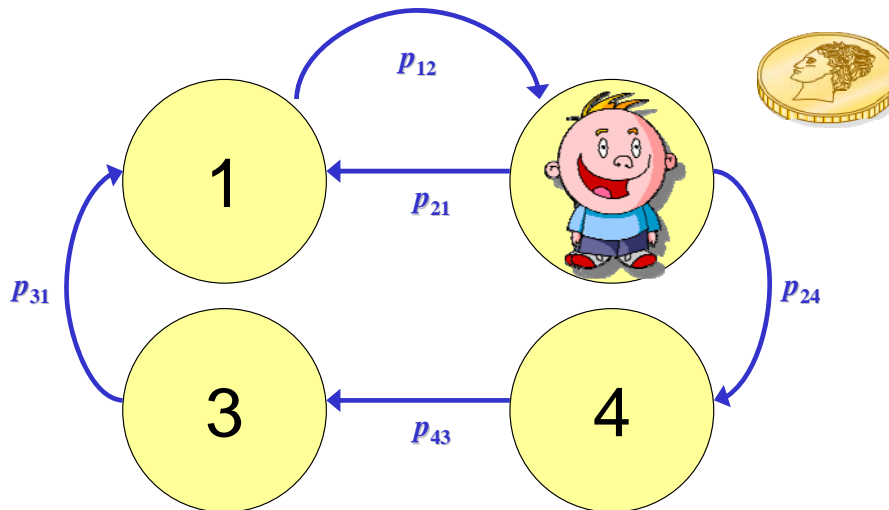
- La scelta del numero di classi da utilizzare per rappresentare il traffico a cui è o sarà soggetto l'impianto, varia da caso a caso: la regola generale è quella di fare tante classi quante sono le diverse tipologie di comportamenti delle richieste dal punto di vista delle prestazioni, in modo che ciascuna classe sia più omogenea possibile.

- Ad esempio, la classe delle richieste HTTP potrebbe essere suddivisa ulteriormente in richieste dirette verso file di piccole dimensioni e in richieste dirette verso file di grandi dimensioni, in quanto il tempo speso dal sistema per servire questi due tipi di richieste può essere estremamente diverso, oltre al fatto che i tassi di arrivo dei due tipi di richieste HTTP potrebbero essere molto diversi fra loro.



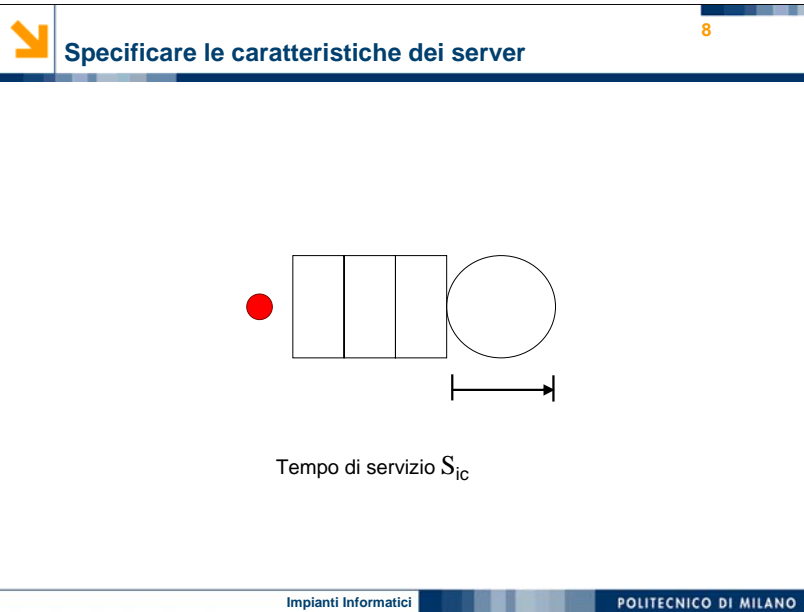
Per quanto riguarda la specifica della struttura delle rete, è sufficiente andare a definire quali connessioni, che si assumono orientate, presenti tra i diversi server.

- Nel caso le connessioni creino delle biforcazioni in uscita da uno stesso server, viene richiesto di indicare la probabilità, detta probabilità di routing, che una richiesta di tipo *c* vada dal server *i* al server *j*.
- Questa probabilità è indicata col simbolo  $p_{ijc}$ .
- Inoltre è importante specificare le connessioni di ingresso/uscita dal modello che consentono di definire il punto di ingresso delle richieste al sistema e il punto in cui esse tornano agli utenti sotto forma di risposta.



Supponiamo di avere, In un modello a rete di code,  $n$  server numerati da 1 ad  $n$ .

- Una richiesta si trova in elaborazione presso un server.
- Terminata l'elaborazione, la richiesta viene passata ad un altro server.
- La probabilità  $p$  che una richiesta vada dal server  $i$  verso il server  $j$  viene indicata con  $p_{ij}$



- Infine, per descrivere i server e' sufficiente specificare il tempo medio richiesto tra ingresso ed uscita di una richiesta supponendo che non via sia coda al suo arrivo presso il server.

- Infatti, poiché viene saltata la coda, questo equivale al tempo fisico necessario per attraversare l'ultimo stadio della coda, ovvero ricevere il servizio dal servente.

Ovvero, è necessario misurare quanto tempo impiega la richiesta tra il suo ingresso e l'uscita da un server nei casi in cui questo sia vuoto all'arrivo della richiesta.

Il valore medio dei tempi misurati e' detto tempo di servizio. Per un server  $i$  il tempo di servizio di una richiesta di classe  $c$  è indicato con la lettera  $S_{ic}$ .



Modelli in forma prodotto

9

---

$$E[x] = \sum_x xp(x) \begin{matrix} \nearrow S_{ic} \\ \searrow \lambda_c \end{matrix}$$

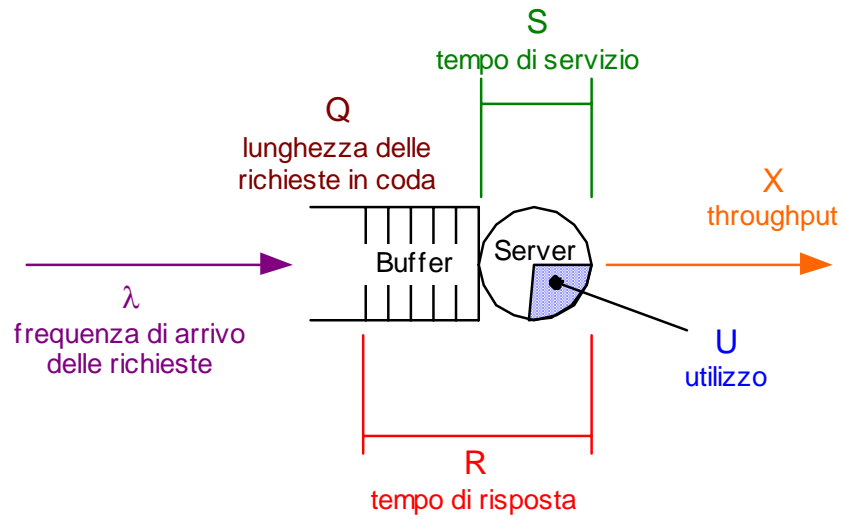
|                |  |
|----------------|--|
| Esponenziale   | $p(x) = \lambda e^{-\lambda x}$  |
| Uniforme       | $p(x) = c \quad \forall x$   |
| Deterministica | $p(x) = \begin{cases} 1 & \text{if } x = d \\ 0 & \text{if } x \neq d \end{cases}$ |


Impianti Informatici
POLITECNICO DI MILANO

Poiche' sia i tempi di servizio che i tasso di arrivo sono delle quantita' medie, quello che stiamo implicitamente assumendo e che esse derivino da delle distribuzioni statistiche di cui noi consideriamo soltanto la media.

- Le reti di code che tratteremo, note come reti di code in forma prodotto, consentono di rappresentare in modo affidabile sistemi in cui i tasso di arrivo e i tempi di servizio seguono particolari distribuzioni, quali
- la distribuzione esponenziale,
- la distribuzione uniforme
- la distribuzione deterministica.

Ad esempio, se si verifica dall'analisi dei file di log che i tempi di servizio di un Web server non seguono nessuna delle suddette leggi, allora il relativo modello a reti di code fornira' soltanto una approssimazione del suo comportamento



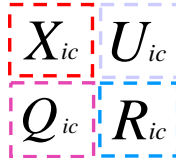


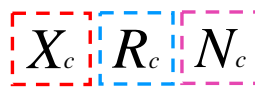
**Misure ottenibili da una rete di code**

11

---

- Misure di un singolo server vs misure aggregate
  
- Misure di un singolo server
  - Throughput
  - Utilizzo
  - Tempo di risposta
  - Lunghezza di coda
  
- Misure aggregate
  - Throughput
  - Tempo di risposta
  - Numero di job





Impianti Informatici

POLITECNICO DI MILANO

E' ora venuto il momento di vedere quali informazioni possano essere ricavate dall'analisi di un modello a reti di code.

- In generale, distinguiamo tra misure relative ad un singolo server e misure aggregate, che descrivono il comportamento di un insieme di piu' server o a limite dell'intera rete.
- Dato un server  $i$  e una richiesta di classe  $c$ , con una rete di code possiamo calcolare
  - il throughput  $X_{ic}$  del server, ovvero il numero di richieste di classe  $c$  servite in media in un secondo;
  - l'utilizzo di classe  $c$   $U_c$ , ovvero la percentuale di tempo in cui il sever e' impegnato a processare richieste di tipo  $c$ ;
  - il tempo di risposta  $R_{ic}$ , ovvero il tempo medio che intercorre tra l'ingresso e l'uscita di una richiesta  $c$  dal server
  - la lunghezza di coda  $Q_{ic}$ , cioe' il numero medio di richieste in coda presso il server comprendendo anche quella che sta attualmente ricevendo il servizio.
- Per quanto riguarda le misure aggregate, troviamo
  - il throughput di classe  $c$   $X_c$  che e' il throughput della rete sul collegamento di uscita
  - il tempo di risposta  $R_c$ , cioe' il tempo che una richiesta impiega per uscire dalla rete
  - e infine il numero di job  $N_c$  che e' la somma di tutte le code  $Q_{ic}$  che si stanno considerando



- Leggi generali valide per tutte le reti

### Legge dell'utilizzo

$$X_{ic} = \frac{C_{ic}}{U_{ic}}$$

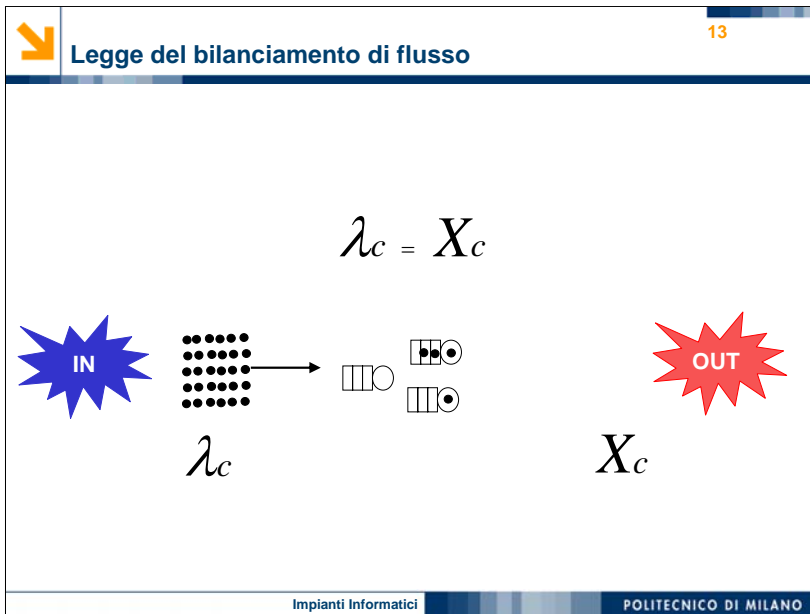
$$U_{ic} = \frac{B_{ic}}{T}$$

$$S_{ic} = \frac{B_{ic}}{C_{ic}}$$

Impianti Informatici

POLITECNICO DI MILANO

- Per ottenere le misure che abbiamo appena visto possono essere sufficienti alcune formule generali valide anche per modelli a reti di code non in forma prodotto.
- Queste leggi sono dette leggi dell'analisi operativa dei modelli a reti di code.
- La prima legge che vediamo e' nota come legge dell'utilizzo.
- Essa ci consente di legare il throughput  $X_{ic}$  di un server, al suo utilizzo  $U_{ic}$  tramite il tempo di servizio  $S_{ic}$ .
- 
- Per dimostrare la legge supponiamo di osservare per un periodo di tempo  $T$  un server della nostra rete.
- Sia  $C_{ic}$  il numero di richieste di classe  $c$  completate dal server  $i$  durante questo periodo.
- Indichiamo poi con  $B_{ic}$  la percentuale di tempo in cui  $i$  e' stato impegnato a servire richieste di classe  $c$ .
- Allora per le definizioni che abbiamo dato possiamo scrivere:
- $X_{ic} = C_{ic}/T$  e
- $U_{ic} = B_{ic}/T$ , in quanto i valori medi di  $C_{ic}$  e  $B_{ic}$  nel periodo sono proprio il throughput e l'utilizzo di classe  $c$  presso il server  $i$ .
- Analogamente, il tempo di servizio  $S_{ic}$  sara' il tempo medio impiegato dal server  $i$  per completare ciascuna delle  $C_{ic}$  richieste, ovvero sara' dato dal rapporto  $B_{ic}$  diviso  $C_{ic}$ .
- Ma allora con un semplice passaggio algebrico otteniamo  $U_{ic} = B_{ic}/T = (B_{ic}/C_{ic}) * C_{ic}/T$  raggruppando otteniamo  $U_{ic} = S_{ic} * X_{ic}$ , formula che costituisce la legge dell'utilizzo.
- Essa, dunque, afferma che l'utilizzo di un server e il suo throughput sono due quantita' proporzionali, legate l'una all'altra tramite il tempo di servizio  $S_{ic}$ .



La seconda legge che vediamo e' la legge del bilanciamento di flusso.

- La legge del bilanciamento di flusso afferma che il tasso di arrivo  $\lambda_c$ , riferito ad una rete che processa richieste di classe  $c$ , e il corrispondente throughput  $X_c$  in media coincidono.
- Questo deriva dal fatto che se osserviamo il sistema per un periodo  $T$  sufficientemente lungo, allora il numero di richieste arrivate e il numero di richieste completate sono approssimativamente identici.
- Piu' precisamente, differiscono solo per le richieste arrivate, ma non ancora completate e dunque ancora all'interno della rete.

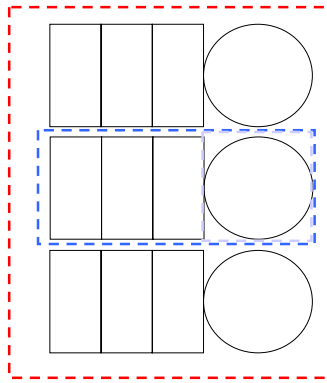
La legge di Little
14

- Generalizzazione della legge dell'utilizzo

$$N_c = X_c R_c$$

$$Q_{ic} = X_{ic} R_{ic}$$

$$U_{ic} = X_{ic} S_{ic}$$



Impianti Informatici
POLITECNICO DI MILANO

Veniamo quindi alla terza legge dell'analisi operativa, la legge di Little.

- La legge di Little è una generalizzazione della legge dell'utilizzo ed è la formula più famosa tra quelle che presentiamo
- In generale essa è espressa tramite la formula  $N_c = X_c R_c$ , e dunque lega il numero di richieste di classe  $c$  all'interno di un gruppo di server, con le misure di throughput e tempo di risposta per il gruppo considerato

Esistono inoltre delle specializzazioni di questa legge nel caso in cui si consideri un gruppo formato da un unico server:

- In questo caso la legge di Little che abbiamo appena visto assume la forma  $Q_{ic} = X_{ic} R_{ic}$ , in quanto il numero di utenti dentro un unico server è proprio la lunghezza di coda  $Q_{ic}$
- Infine, se riferita ad un server di cui non si considera la coda, ma soltanto la componente che offre il servizio, allora il prodotto  $X_{ic}$  per  $R_{ic}$  diventa  $X_{ic}$  per  $S_{ic}$ , poiché il tempo di risposta in questo caso coincide con il tempo di servizio non avendo alcuna richiesta che impedisca alla nostra richiesta di andare subito in esecuzione. Ma sappiamo già che il prodotto  $X_{ic}$  per  $S_{ic}$  è l'utilizzo della stazione e quindi, grazie alla legge di Little, abbiamo scoperto che  $U_{ic}$  coincide con il numero medio di utenti che stanno ricevendo servizio presso il server  $i$



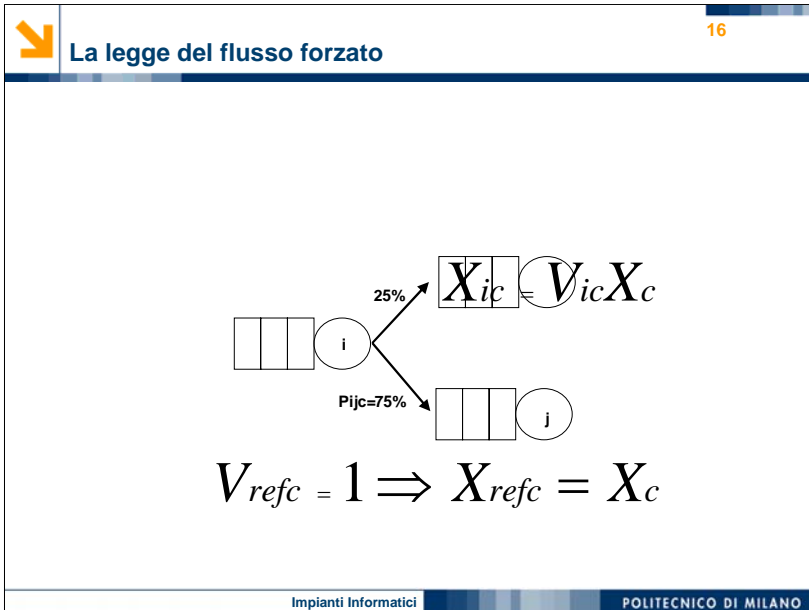
- Applicabile alla totalita' dei sistemi esistenti
- Consente di verificare la consistenza di misure di performance
- Sfruttata ampiamente dagli algoritmi di risoluzione dei modelli a reti di code



John D.C. Little

La legge di Little e' una legge fondamentale almeno per tre ragioni:

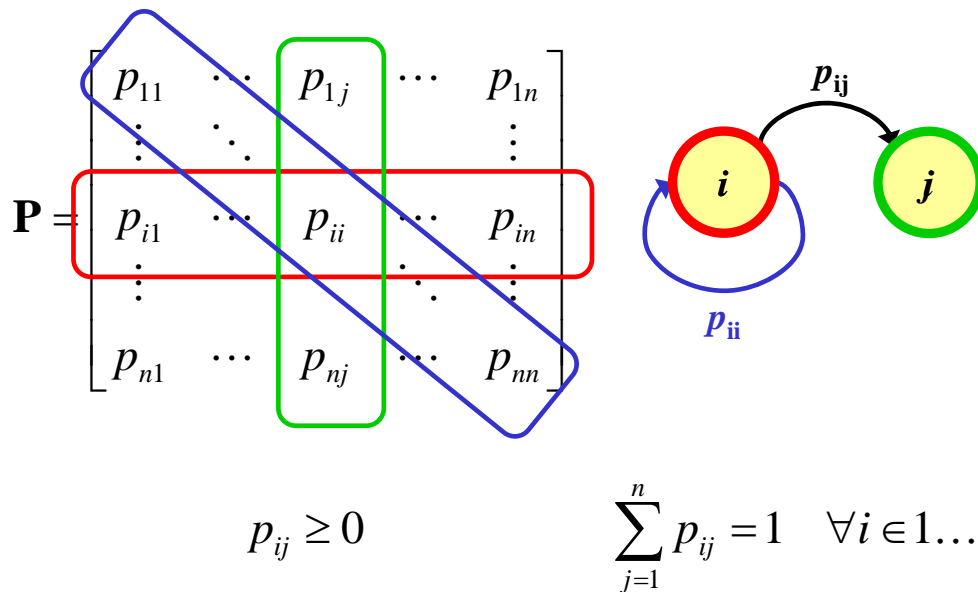
- primo per la sua estrema generalita', che la rende applicabile a tutti i sistemi reali
- in secondo luogo e' utile come formula per verificare se le misure di utilizzi, tempi di risposta, e throughput ricavate da un sistema reale sono effettivamente consistenti. In caso contrario e' probabile che siano verificati degli errori di misura che devono essere corretti.
- limitandoci invece alla teoria delle reti di code, la legge di Little riduce di molto la complessita' dei modelli che stiamo considerando, perche' e' sufficiente calcolare solo due delle tre quantita' in gioco nelle diverse varianti della formula per ricavare immediatamente anche la terza quantita' incognita. Questo fa si' che molti dei principali algoritmi risolutivi dei modelli a reti di code sfruttino ampiamente tale Legge.



Concludiamo la panoramica sulle leggi dell'analisi operativa con la legge del flusso forzato.

- Essa permette di tenere in considerazione nel calcolo delle misure dei modelli a reti di code anche la struttura della rete e le probabilità di routing  $P_{ijc}$
- Concentriamoci sui soli server  $i$  e  $j$ . Poiché la probabilità che le richieste in uscita da  $i$  giungano a  $j$  sono fissate, ci aspettiamo che esista un legame tra il flusso di richieste in uscita da  $i$  e il flusso di richieste in uscita da  $j$
- Tale legame esiste ed è catturato proprio dalla legge del flusso forzato, che dice che il throughput  $X_{ic}$  di ciascuna stazione è proporzionale al throughput  $X_c$  della rete misurato presso la stazione scelta come riferimento. Il coefficiente di proporzionalità è detto numero di visite delle richieste di classe  $c$  al server  $i$ .
- Presso la stazione assunta il cui throughput  $X_c$  è scelto come riferimento nella rete, il numero di visite è fissato convenzionalmente a 1. Con questa scelta, infatti, il throughput  $X_{refc}$  della stazione diventa come voluto il throughput della rete  $X_c$





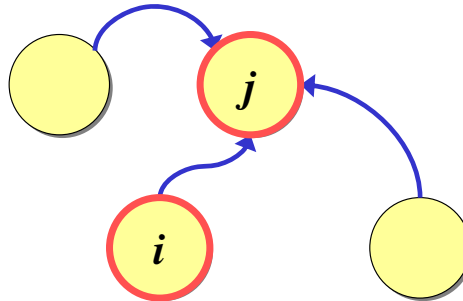
Conoscendo, per tutti i server  $i$  e  $j$ , la probabilità  $p_{ij}$  che dal server  $i$  una richiesta venga inoltrata verso un server  $j$ ,

- è possibile costruire una matrice di probabilità di routing  $\mathbf{P}$
- in cui le righe rappresentano il server di partenza
- e le colonne il server di arrivo.
- Si noti che gli elementi  $p_{ij}$  sulla diagonale della matrice rappresentano la probabilità un richiesta venga di nuovo elaborata dallo stesso server.
- Gli elementi  $p_{ij}$  della matrice di routing godono di alcune proprietà importanti. Innanzitutto, gli elementi sono tutti maggiori od uguali a zero, dato che rappresentano delle probabilità.
- La seconda proprietà ci dice che la somma degli elementi di una riga deve essere pari ad uno. Questo perché da un server  $i$  deve essere sempre possibile andare da qualche parte ...
- Si noti che, quando un elemento  $p_{ij}$  è zero, questo equivale a cancellare l'arco che unisce il server  $i$  al server  $j$ , perché siamo certi che non avverranno mai routing di richieste tra  $i$  e  $j$ .



$$V_{cj} = \sum_{i=1}^n V_{ci} p_{cij}$$

$$V = VP$$



Vediamo adesso come calcolare (da un punto di vista probabilistico) il legame tra visite e probabilità di routing. Facciamo riferimento ad una classe  $c$

- Calcoliamo prima il numero di visite al server  $j$ .
- Questo è dalla probabilità dal numero di visite al server  $i$
- moltiplicato per la probabilità  $p_{cij}$  di routing dallo server  $i$  al server  $j$ .
- Il tutto sommato sugli  $n$  possibili server  $i$ .
- In forma matriciale un sistema di  $n$  equazioni nelle  $n$  incognite  $V_i$ . Questo sistema ha determinante nullo, ossia ammette infinite soluzioni. Imponendo che le visite alla stazione di riferimento siano pari ad uno, il sistema diventa risolvibile



- Le visite contengono l'informazione sulla struttura della rete
- Domanda globale di servizio:

$$D_{ic} = V_{ic}S_{ic}$$

- Formulazione compatta delle leggi operazionali

$$U_{ic} = X_c D_{ic}$$

$$Q_{ic} = X_c V_{ic} R_{ic} = X_{ic} R_{ic}$$

$$N_c = X_c V_{ic} R_c = X_{ic} R_c$$

- Il principale vantaggio dell'uso della legge del flusso forzato è che possiamo mascherare la struttura della rete specificate tramite le probabilità  $P_{ijc}$  all'interno delle visite.
- Inoltre, se definiamo domanda globale di servizio la quantità  $D_{ic} = V_{ic}S_{ic}$ ,
- possiamo semplificare le leggi dell'analisi operativa riferendole al solo throughput di classe  $c$  della rete  $X_c$ . Abbiamo allora
- $U_{ic} = X_c D_{ic}$
- $Q_{ic} = X_c V_{ic} R_{ic}$
- e  $N_c = X_c V_{ic} R_c$
- Tuttavia, con un leggero abuso di notazione si preferisce per la  $Q_{ic}$  e la  $N_c$  inglobare il termine delle visite direttamente dentro la  $R_{ic}$  e la  $R_c$  le quali, come vedremo nelle successive lezioni, non saranno un funzione dei soli tempi di servizio dei server della rete, ma dovranno invece essere calcolate tramite le domande globali di servizio