Performance Modeling of Computer Systems and Networks

Prof. Vittoria de Nitto Personè

Priority scheduling

Università degli studi di Roma Tor Vergata
Department of Civil Engineering and Computer Science Engineering

1

Service classes

- (Multimedia traffic)
- Quality of Service (QoS)
- Penalties

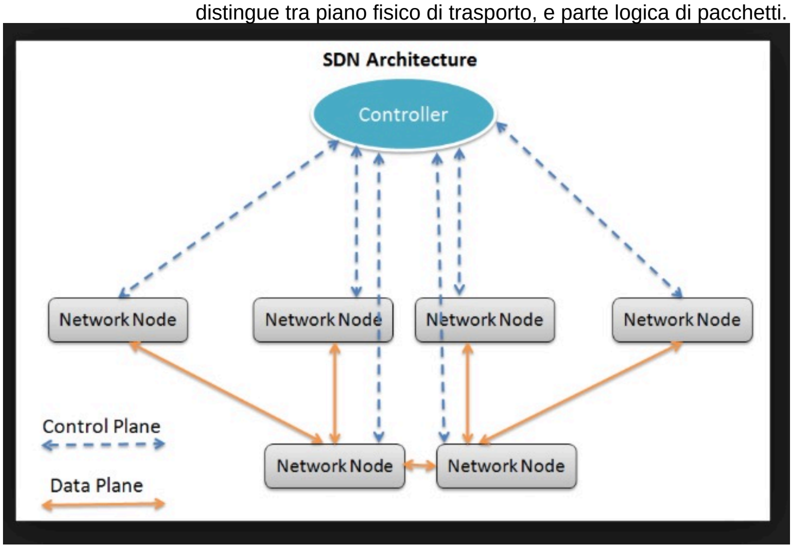
The proper scheduling policy can improve performance of a server tremendously.
It costs nothing to alter your scheduling policy (no money, no new hardware), so the performance gain comes for free.

2

SDN

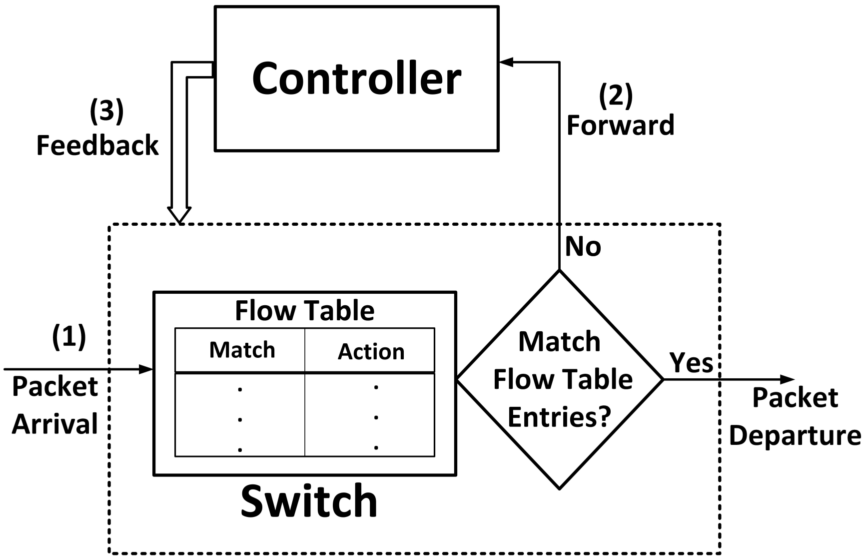
architettura per la realizzazione di reti di telecomunicazioni
il piano di controllo della rete e quello di trasporto dei dati sono separati logicamente

distingue tra piano fisico di trasporto, e parte logica di pacchetti.



3

Se da (1) arriva pacchetto non presente in tabella, lo mando al controllore (2), che fa operazioni di aggiornamento tabella, rimandandolo allo switch (3). Quando lo rimanda allo switch ha priorità maggiorata (poiché non essendo inviato ha accumulato ritardo).



prof. Vittoria de Nitto Personè 4

4

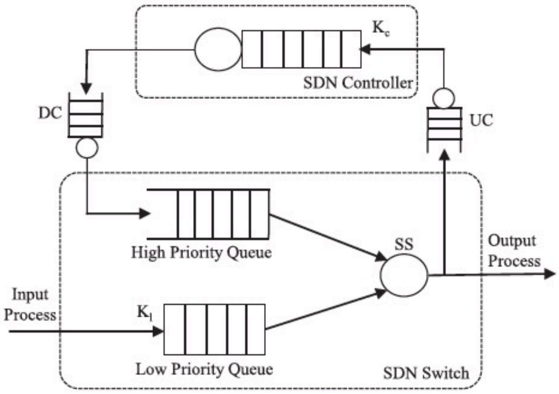
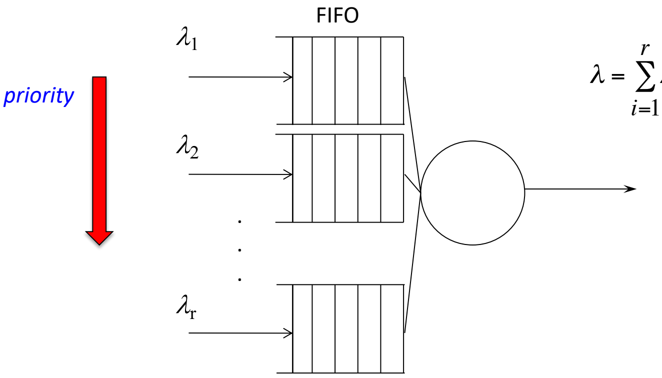


Fig. 1. The PQ-based SDN system architecture.

Miao W., Min G., Wu Y., Wang H. and Hu J., 2016. Performance Modelling and Analysis of Software-Defined Networking under Bursty Multimedia Traffic. *ACM Trans. Multimedia Comput. Commun. Appl.*, Vol. 12, No. 5s, Article 77.

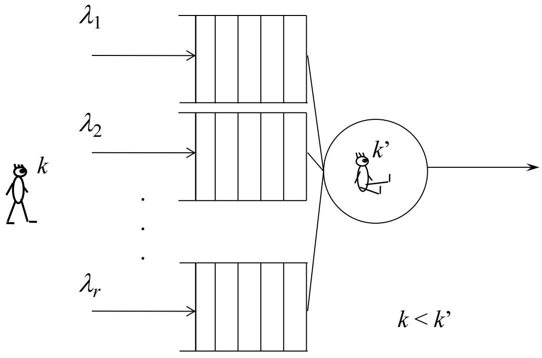
Priority classes



$$\lambda = \sum_{i=1}^{r} \lambda_i$$

Coda 1 priorità max, coda 'r' priorità minima. Il flusso totale lamda è la somma di questi 'r' flussi, e il servente qui è SINGOLO.

Priority classes



$k < k'$

- *abstract* (non dipende da quanta è la size del job)
- *size based* (danno ordine in base alla quantità di lavoro chiesto)

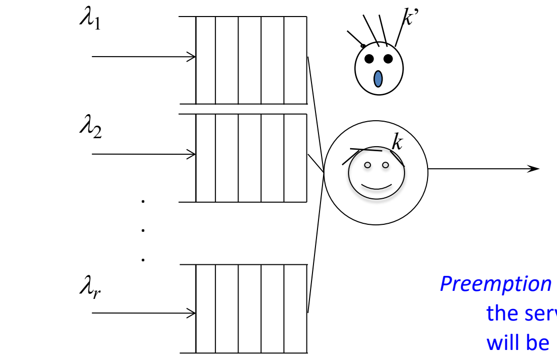
- *preemptive*
- *not-preemptive*

servizio interrompibile o meno!

In servizio c'è " k' ", questo vuol dire che tutte le code da 1 a " k' - 1" sono vuote, sennò toccava a loro in quell'istante. Se arriva 'k' più importante, e c'è PRELAZIONE, interrompo k' in favore di k.

Priority classes with preemption

$k < k'$

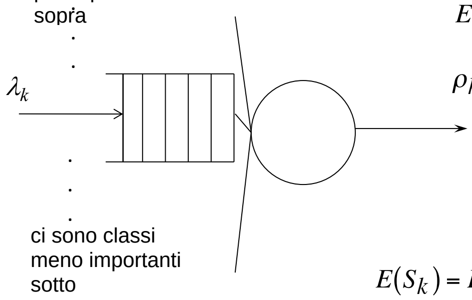


Preemption without loss:
the service of the interrupted job will be resumed from the interruption point

interrompo solo se posso salvare stato e riprenderlo da istante interruzione.
SI parla di "interruzione senza perdita".

Abstract priority without preemption

ci sono classi
più importanti
sopra



ci sono classi
meno importanti
sotto

S_k in "generale"

$$E(S_k) = \frac{1}{\mu_k} \qquad \sigma^2(S_k)$$

$$\rho_k = \lambda_k E(S_k) \qquad \rho = \sum_{i=1}^{r} \rho_i \text{ totale}$$

quanto arriva* quanto si ferma

abstract

tempo di servizio uguale
per tutti.

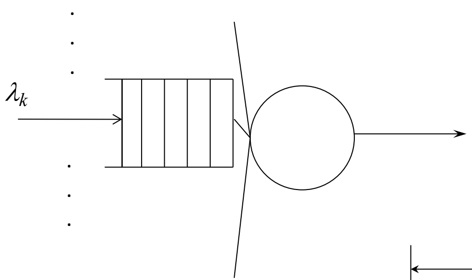
$$E(S_k) = E(S) = \frac{1}{\mu}, \quad \sigma^2(S_k) = \sigma^2(S), \forall k$$

$$\rho_k = \lambda_k E(S)$$

Nel caso size-based,
la coda-k ha un certo tempo di risposta
diverso da altre code (ordino in base alla size).

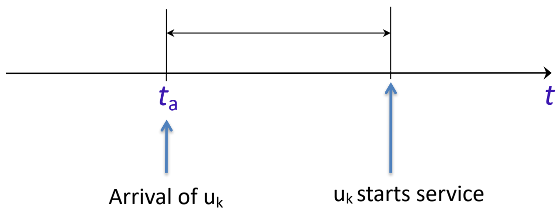
Abstract priority without preemption

local performance measures



tempo di attesa
specifico per una classe k?

$$E\left(T_{Q_k}\right)?$$

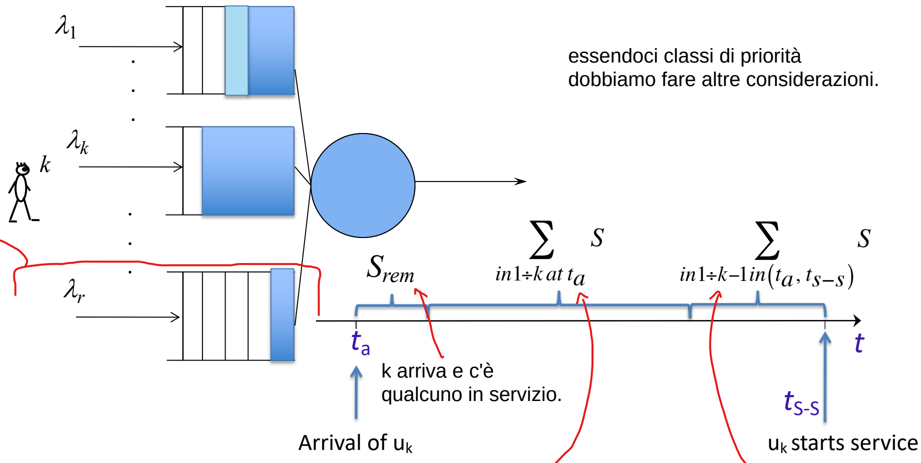


Abstract priority without preemption

local performance measures

essendoci classi di priorità
dobbiamo fare altre considerazioni.

per "k", quello che c'è
sotto non interessa.



k arriva e c'è
qualcuno in servizio.

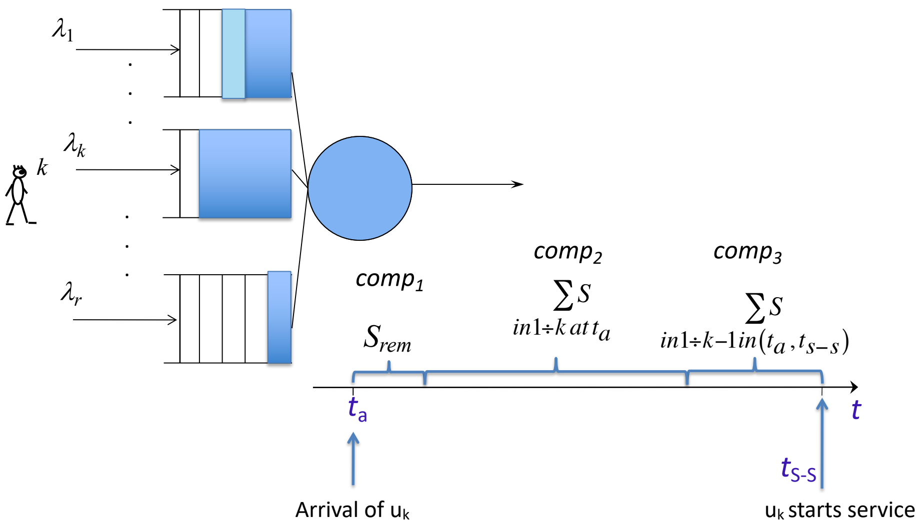
attendo TUTTI quelli
da classe '1' a classe k.

Quelli che arrivano "dopo" ma nell'intervallo
t_a (arriva dopo "k") e t_s - s (prima che prende
servizio)

11

Abstract priority without preemption

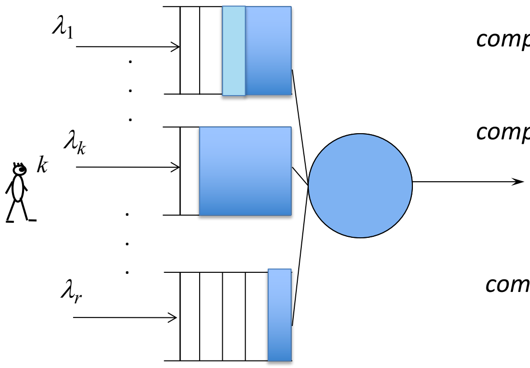
local performance measures



12

Abstract priority without preemption

local performance measures



comp₁: $$E(S_{rem}) = \frac{\lambda}{2} E(S^2)$$

$comp_2$: proportional to the load of queues 1:k $$= \frac{1}{1 - \sum_{i=1}^{k} \rho_i}$$

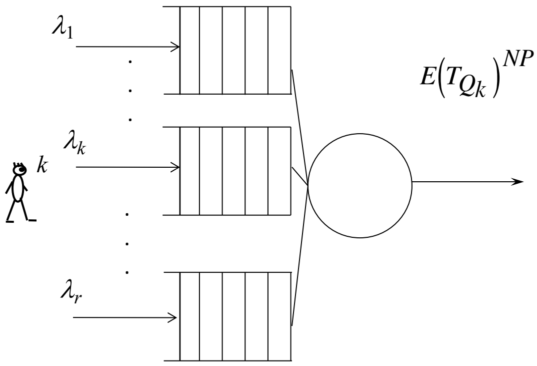
nel caso singolare era: $\frac{1}{1 - \text{rho}}$

$comp_3$: proportional to the load of queues 1:k-1 $$= \frac{1}{1 - \sum_{i=1}^{k-1} \rho_i}$$

13

Abstract priority without preemption

local performance measures



$$E(T_{Q_k})^{NP\_priority} = \frac{\frac{\lambda}{2} E(S^2)}{\left(1 - \sum_{i=1}^{k} \rho_i\right)\left(1 - \sum_{i=1}^{k-1} \rho_i\right)}$$

classi di priorità senza prelazione!

$$E(T_{Q_k}) \le E(T_{Q_{k+1}})$$

abbastanza banale.

14

Abstract priority without preemption

local performance measures

$$\boxed{E\left(T_{Q_k}\right) \le E\left(T_{Q_{k+1}}\right)}$$

$$\frac{\frac{\lambda}{2}E\left(S^2\right)}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)} \le \frac{\frac{\lambda}{2}E\left(S^2\right)}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k+1}\rho_i\right)}$$

$$\frac{1}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)} \le \frac{1}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k+1}\rho_i\right)}$$

prendo solo i denominatori:

$$\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right) \ge \left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k+1}\rho_i\right)$$

15

Abstract priority without preemption

local performance measures

$$\cancel{\left(1-\sum_{i=1}^{k}\rho_i\right)}\left(1-\sum_{i=1}^{k-1}\rho_i\right) \ge \cancel{\left(1-\sum_{i=1}^{k}\rho_i\right)}\left(1-\sum_{i=1}^{k+1}\rho_i\right)$$

$$\cancel{1}-\sum_{i=1}^{k-1}\rho_i \ge \cancel{1}-\sum_{i=1}^{k+1}\rho_i$$

$$\sum_{i=1}^{k+1}\rho_i \ge \sum_{i=1}^{k-1}\rho_i \qquad \rho_i \ge 0, \forall i$$

$$\boxed{E\left(T_{Q_k}\right) \le E\left(T_{Q_{k+1}}\right)}$$

16

Abstract priority without preemption

local performance measures

Prestazioni locali, rispetto alla classe

indipendente dalla classe

$$E\left(T_{S_k}\right) = E\left(T_{Q_k}\right) + E(S) \qquad E\left(T_{S_k}\right) \le E\left(T_{S_{k+1}}\right)$$

usando Little:

$$E\left(N_{Q_k}\right) = \lambda_k E\left(T_{Q_k}\right)$$

$$E\left(N_{S_k}\right) = \lambda_k E\left(T_{S_k}\right) \qquad E\left(N_{S_k}\right) = E\left(N_{Q_k}\right) + \rho_k$$

rho specifico
della classe

Abstract priority without preemption

global performance measures

And the "global" performance? Devo fare una media pesata.

$$E\left(T_Q\right)^{NP\_priority} = E\left(E\left(T_{Q_k}\right)\right) = \sum_{k=1}^{r} p_k E\left(T_{Q_k}\right)$$

$$p_k = \frac{\lambda_k}{\lambda}$$

% di traffico su una coda / tutto il traffico

and similarly for $E(T_S)^{NP\_priority}$

$$E(T_S)^{NP\_priority} = E\left(T_Q\right)^{NP\_priority} + E(S)$$

Analytical models
priority scheduling

Abstract priority without preemption

$$\lambda_k = p_k \lambda$$

$$\rho_k = \lambda_k E(S) = p_k \lambda E(S) = p_k \rho$$

rho_k = probabilità di essere di quella classe * rho_totale

prof. Vittoria de Nitto Personè 19

19

Analytical models
priority scheduling

priority vs no-priority

How are the performance improved in respect of a simple abstract scheduling not-considering the priority classes? quante classi vanno meglio? quali peggio?

$$E\left(T_{Q_k}\right)^{NP\_priority} = \frac{\frac{\lambda}{2} E\left(S^2\right)}{\left(1 - \sum_{i=1}^{k} \rho_i\right)\left(1 - \sum_{i=1}^{k-1} \rho_i\right)} \quad ? \quad E\left(T_Q\right)^{KP} = \frac{\frac{\lambda}{2} E\left(S^2\right)}{1 - \rho}$$

The highest priority class:

$$E\left(T_{Q_1}\right)^{NP\_priority} = \frac{\frac{\lambda}{2} E\left(S^2\right)}{\left(1 - \rho_1\right)} \leq E\left(T_Q\right)^{KP}$$

prima classe va sicuramente meglio (vede solo se stessa), meglio rispetto a vedere tutti mischiati.
rho_1 è più grande di rho???

prof. Vittoria de Nitto Personè 20

20

priority vs no-priority

How are the performance improved in respect of a simple abstract scheduling not-considering the priority classes?

The lowest priority class:

$$E\left(T_{Q_r}\right)^{NP\_priority} = \frac{\frac{\lambda}{2}E\left(S^2\right)}{(1-\rho)\left(1-\sum_{i=1}^{r-1}\rho_i\right)} \geq E\left(T_Q\right)^{KP}$$

And what about the "global" performance?

$$E\left(T_Q\right)^{NP\_priority} = E\left(T_Q\right)^{KP}$$

se vado a fare somme pesate, non ho vantaggi. Le classi più basse annullano i vantaggi delle classi più alte.

$$\Downarrow$$

$$E(T_S)^{NP\_priority} = E(T_S)^{KP}$$

21

Se pensiamo a KP, scheduling astratti. Ma anche NP_priority ha uno scheduling astratto, e quindi globalmente non è nient'altro che una KP.

priority vs no-priority

$$E\left(T_Q\right)^{NP\_priority} = E\left(E\left(T_{Q_k}\right)\right) = \sum_{k=1}^{r} p_k E\left(T_{Q_k}\right) = E\left(T_Q\right)^{KP}$$

$r=2$

$$E\left(T_Q\right) = p_1 E\left(T_{Q_1}\right) + p_2 E\left(T_{Q_2}\right) = p_1 \frac{\frac{\lambda}{2}E\left(S^2\right)}{(1-\rho_1)} + p_2 \frac{\frac{\lambda}{2}E\left(S^2\right)}{(1-\rho)(1-\rho_1)}$$

$$= \frac{\lambda}{2}E\left(S^2\right)\left[\frac{p_1}{(1-\rho_1)} + \frac{p_2}{(1-\rho)(1-\rho_1)}\right] = \frac{\lambda}{2}E\left(S^2\right)\frac{p_1(1-\rho)+p_2}{(1-\rho)(1-\rho_1)} = \frac{\frac{\lambda}{2}E\left(S^2\right)}{1-\rho}$$

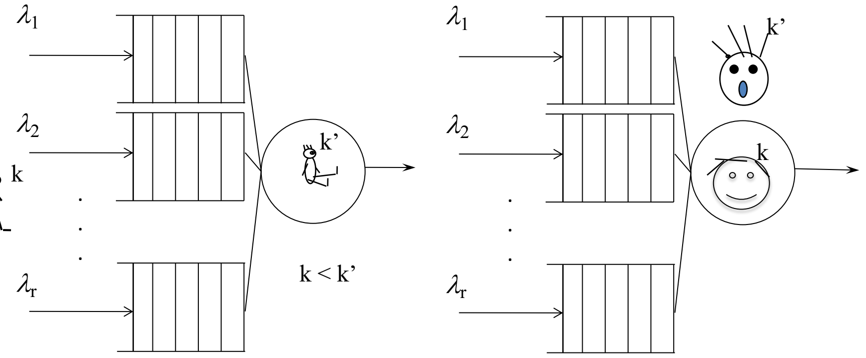
p1 - p*p1 + p2

↑

p1 + rho1 + p2 = 1 + rho1, si annulla con denominatore.

22

Preemptive Priority

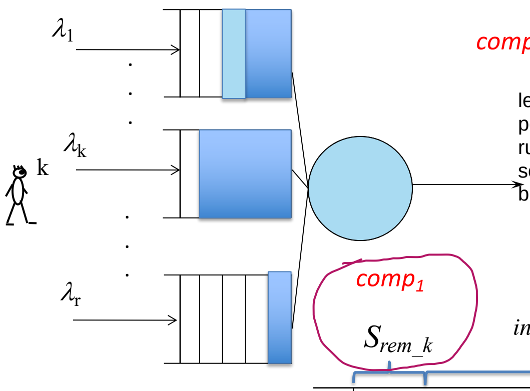


Preemption without loss:
the service of the interrupted job will be resumed from the interruption point

23

Preemptive Priority

local performance measures



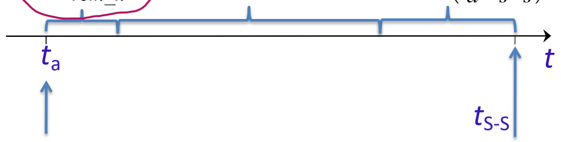
comp₁: $E\left(S_{rem_k}\right) = \frac{1}{2}\sum_{i=1}^{k} \lambda_i E\left(S^2\right)$

le classi da 1 a k,
perchè se arrivano e sono più importanti, gli rubano il posto. Prima non era una sommatoria, perchè nessuno poteva bloccarmi.

comp₁: S_{rem_k}

comp₂: $\sum_{in\, 1 \div k\, at\, t_a} S$

comp₃: $\sum_{in\, 1 \div k-1\, in (t_a, t_{S-S})} S$



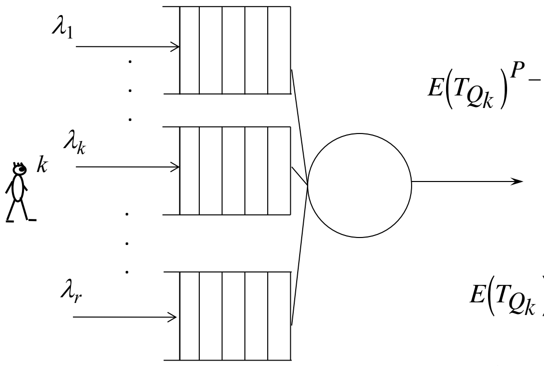
Arrival of u_k

u_k starts service

24

Preemptive Priority

quando non c'è prelazione, c'è tutto lambda.



$$E\left(T_{Q_k}\right)^{P\_priority} = \frac{\frac{1}{2}\sum_{i=1}^{k}\lambda_i E\left(S^2\right)}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)}$$

$$E\left(T_{Q_k}\right)^{P\_priority} \le E\left(T_{Q_{k+1}}\right)^{P\_priority}$$

$$E\left(T_{Q_k}\right)^{P\_priority} \le E\left(T_{Q_k}\right)^{NP\_priority}$$

$$E\left(T_Q\right)^{P\_priority} \le E\left(T_Q\right)^{NP\_priority} = E\left(T_Q\right)^{KP}$$

25

Ho guadagnato qualcosa per alcune classi, perchè il tempo con prelazione globale è minore uguale rispetto alla variante senza prelazione. Ho finalmente guadagnato sulla KP. Nel modello SENZA Prelazione, se c'è in servizio 'k', e arriva un 'k-1', deve aspettare. Qui, con prelazione, sostituisco subito, quindi posso dire ancora meglio che NON VEDO CLASSI DI PRIORITÀ INFERIORI. Per questo c'è guadagno.

Preemptive Priority

stai sereno che lo vedo

$$\frac{\frac{1}{2}\sum_{i=1}^{k}\lambda_i E\left(S^2\right)}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)} \le \frac{\frac{1}{2}\sum_{i=1}^{k+1}\lambda_i E\left(S^2\right)}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k+1}\rho_i\right)}$$

$$\frac{\sum_{i=1}^{k}\lambda_i}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)} \le \frac{\sum_{i=1}^{k+1}\lambda_i}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k+1}\rho_i\right)}$$

$$\frac{\sum_{i=1}^{k}\lambda_i}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)} \le \frac{\sum_{i=1}^{k+1}\lambda_i}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)}$$

$$\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right) \ge \left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k+1}\rho_i\right)$$

26

Preemptive Priority

$$E(T_Q)^{X\_priority} = E\left(E(T_{Qk})\right) = \sum_{k=1}^{r} p_k E(T_{Qk})$$
$$= p_1 E(T_{Q1}) + p_2 E(T_{Q2}) + \ldots + p_r E(T_{Qr})$$

$$E(T_Q)^{NP\_priority} = p_1 E(T_{Q1}) + p_2 E(T_{Q2}) + \ldots + p_r E(T_{Qr})$$

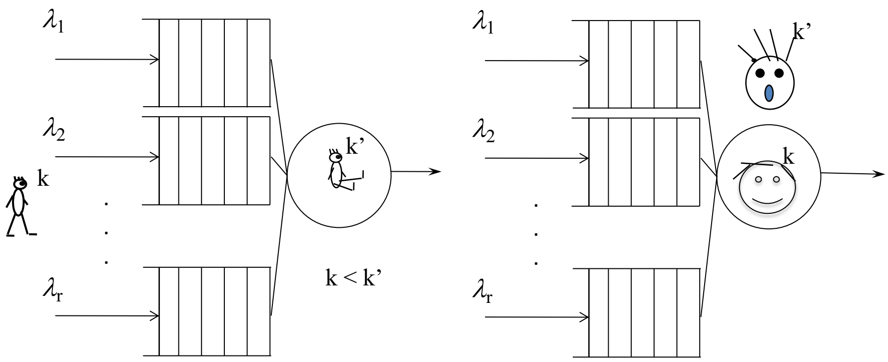
$$E(T_Q)^{P\_priority} = p_1 E(T_{Q1}) + p_2 E(T_{Q2}) + \ldots + p_r E(T_{Qr})$$

$$E(T_Q)^{P\_priority} \leq E(T_Q)^{NP\_priority} = E(T_Q)^{KP}$$

Se parliamo di servizio, un job che viene buttato fuori e poi ritorna, perde più tempo. La sua classe è sempre k, quindi perde molto tempo. Tempo attesa: tempo arrivo - tempo presa servizio. Adesso specifichiamo "tempo presa servizio" con "tempo presa servizio la prima volta".

Preemptive Priority

Due to preemption, the service time of a job of class k may increase for the arrivals of higher priority classes



Preemptive Priority

Due to preemption, the service time of a job of class k may increase for the arrivals of higher priority classes

qui conto il tempo per prendere il servizio dopo essere stato buttato fuori!

Virtual service time $E(S_{virt\_k})$

$$E(S_{virt\_k}) = \frac{E(S)}{1 - \sum_{i=1}^{k-1} \rho_i}$$

viene allungato di un fattore proporzionale a tutti i rho delle classi che mi possono interrompere. La sommatoria arriva a "k-1", una classe "k" non può infatti essere interrotta da sè stessa.

$$E(T_{S_k})^{P\_priority} = E(T_{Q_k})^{P\_priority} + E(S_{virt\_k})$$

$$\geq \qquad\qquad \leq$$

$$E(T_{S_k})^{NP\_priority} = E(T_{Q_k})^{NP\_priority} + E(S)$$

?

29

Non posso dire a priori chi sia più grande, c'è chi va meglio e chi peggio.
Per la classe 1, E(S_virtu_k) = E[S], perchè classe 1 non può essere buttata fuori.
A parte classe 1, le altre non posso dire nulla: quante classi ci sono? quanto sono grandi i flussi?

Preemptive Priority

global response time

And what about the "global" performance?

$$E(T_{Q_k})^{P\_priority} + E(S_{virt\_k})$$

$$E(T_S)^{P\_priority} = E(E(T_{S_k})) = \sum_{k=1}^{r} p_k E(T_{S_k})$$

$$E(T_S)^{P\_priority} = \sum_{k=1}^{r} p_k \left[E(T_{Q_k}) + E(S_{virt\_k})\right]$$

$$= \sum_{k=1}^{r} p_k E(T_{Q_k}) + \sum_{k=1}^{r} p_k E(S_{virt\_k})$$

$$= E(T_Q)^{P\_priority} + \sum_{k=1}^{r} p_k \underbrace{E(S_{virt\_k})}$$

non è più E[S], lo è solo per la classe 1.

30

preemption vs no-preemption

$$E(T_S)^{P\_priority} = E(T_Q)^{P\_priority} + \sum_{k=1}^{r} p_k E(S_{virt\_k})$$

$$\wedge| \qquad\qquad |\vee$$

$$E(T_S)^{NP\_priority} = E(T_Q)^{NP\_priority} + E(S) = E(T_S)^{KP}$$

?

anche in termini globali non posso confrontarli, perchè per i singoli "pezzi" posso dire chi è più piccolo, ma mettendoli insieme non so dire se ho un guadagno o meno.

In general

$$E(T_S)^{P\_priority} \quad ? \quad E(T_S)^{KP}$$

For exponential service time

$$E(T_S)^{P\_priority} = E(T_S)^{KP}$$

!!!

La memoryless annulla tutto il guadagno, ovvero ciò che guadagno in attesa è compensato da ciò che perdo nel servizio.

preemption vs no-preemption

$r=2$

tutto viene dall'ESPONENZIALITA'.

$$E(T_S)^{P\_priority} = p_1 E(T_{S_1}) + p_2 E(T_{S_2})$$

$$= p_1\left[\frac{\frac{\lambda_1}{2}E(S^2)}{(1-\rho_1)} + E(S)\right] + p_2\left[\frac{\frac{\lambda}{2}E(S^2)}{(1-\rho)(1-\rho_1)} + \frac{E(S)}{1-\rho_1}\right]$$

tempo virtuale

prima classe, non ritardabile

seconda classe

from the expo assumption

aspetta tutti

$$= p_1\left[\frac{\rho_1 E(S)}{(1-\rho_1)} + E(S)\right] + p_2\left[\frac{\rho E(S)}{(1-\rho)(1-\rho_1)} + \frac{E(S)}{1-\rho_1}\right]$$

$$= E(S)\left\{p_1\left[\frac{\rho_1 + 1 - \rho_1}{(1-\rho_1)}\right] + p_2\left[\frac{\rho + (1-\rho)}{(1-\rho)(1-\rho_1)}\right]\right\}$$

preemption vs no-preemption

$r = 2$

$$E(T_S)^{P\_priority} = p_1 E\left(T_{S_1}\right) + p_2 E\left(T_{S_2}\right)$$

$$= E(S)\left[\frac{p_1}{(1-\rho_1)} + \frac{p_2}{(1-\rho)(1-\rho_1)}\right]$$

$$= E(S)\frac{p_1(1-\rho) + p_2}{(1-\rho)(1-\rho_1)} = \frac{E(S)}{1-\rho} = E(T_S)^{KP}$$

La proprietà Memoryless semplifica moltissimo i fenomeni.

Priorità astratte: criteri indipendenti da quanto chiedono di servizio.
Se c'è prelazione, l'attesa porta guadagno a classi alte, ma anche globalmente ho vantaggi.
Senza prelazione, quello che guadagno su attesa si perde sul servizio, alla fine ritorno alle prestazioni globali della KP. (sempre le prime classi guadagnano rispetto le ultime, non so quanto).

Con tempo esponenziale torno globalmente al caso della KP.