Case Study 2

Dynamic Quota-Based Admission Control With Subrating in Multimedia Servers

Sheng-Tzong Cheng, Chi-Ming Chen

and Ing-Ray Chen

ACM/Springer Journal on Multimedia Systems, Vol. 8, No. 2, 2000, pp. 83-91.

Background

Reservation-based admission control

Allocates a fraction of the server capacity for a new request based on certain criteria. The allocated server capacity is reserved for the specific request until it leaves the system.

Problem: A new request may be rejected if no available resource is left to serve the request. In such a case, the system incurs a loss due to the rejected request.

Background (Cont.)

Possible ways of reservation-based admission control

Deterministic approach

* using the worst-case scenario to provide absolute Quality of Service (QoS) guarantee

* resources are under-utilized

Best-effort approach

* based on statistical or average estimations of the required data rate

* no absolute QoS guarantee

Subrating Mechanism

Quota-based Reservation

Partition the server capacity into several partitions (or quotas)

Subrating mechanism

Reduce the QoS of low-priority clients to accept a new high-priority client with an objective to achieve a higher ``system value''.

Notation

λ_h	Arrival rate of high-priority clients (HPCs)
λ_l	Arrival rate of low-priority clients (LPCs)
μ	Departure rate of clients
v_h	Reward of a HPC if the client is serviced successfully
v_l	Reward of a LPC if the client is serviced successfully
q_h	Penalty of a HPC if the client is rejected on admission
q_l	Penalty of a LPC if the client is rejected on admission
N	Total number of server capacity slots for servicing clients
$\mathbf{n}_h$	Number of slots reserved for HPC only, $0 <= \mathbf{n}_h <= N$

$\mathbf{n}_l$ Number of slots reserved for LPC only, $0 <= \mathbf{n}_l <= N$ and also $\mathbf{n}_h + \mathbf{n}_l >= 0$

$\mathbf{n}_m$ Number of slots that can be used to service either type of clients, $\mathbf{n}_m = \mathbf{N} - \mathbf{n}_h - \mathbf{n}_l$

X_h Throughput of HPC

X_l Throughput of LPC

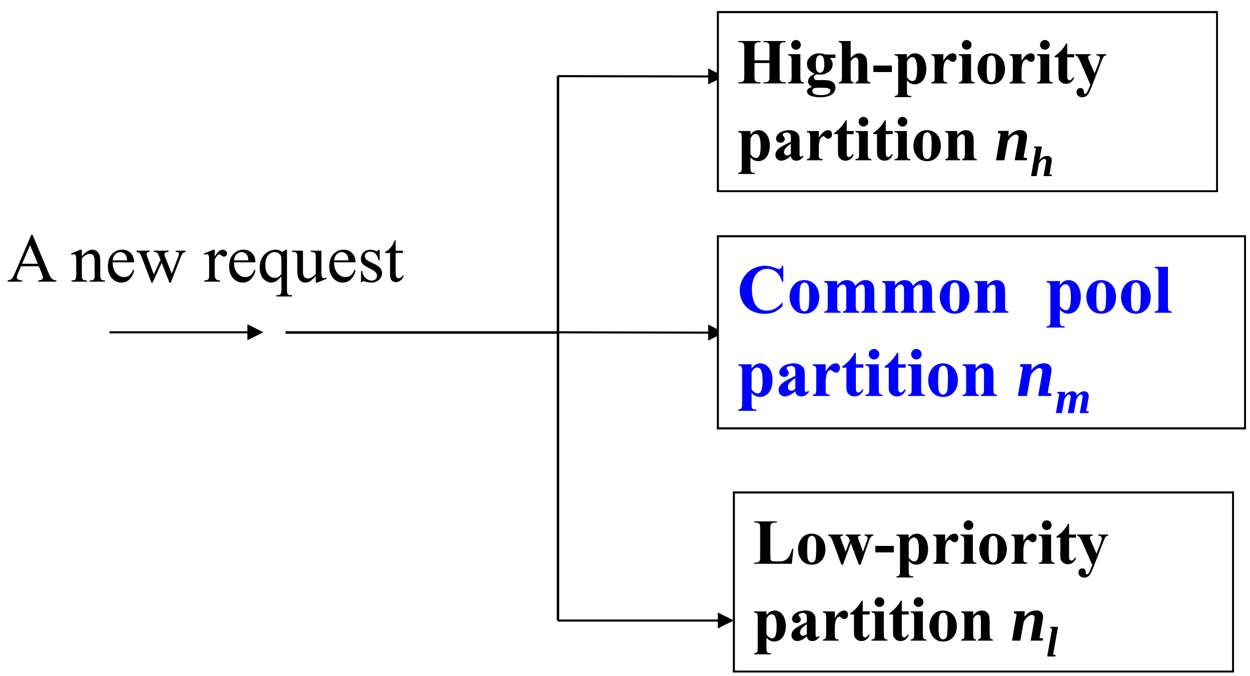
X_{ld} Throughput of degraded LPC

M_h Rejection rate of HPC

M_l Rejection rate of LPC

α Number of LPC to be degraded to accommodate a new HPC

System Model



System Model (cont.)

- **A high priority client does not degrade its QoS, while a low priority client has a range of QoS requirements (Q_{max}, Q_{min}) with $Q_{min} = (1- 1/\alpha)\ Q_{max}$**

- **A low priority client in the common pool area can degrade its QoS once by $1/\alpha$ (to Q_{min}) if necessary; if it departs in degraded service mode, the system only receives** $(1- 1/\alpha)\ v_l$

- **If the common pool area is all occupied, α low-priority clients (if available) each degrade their QoS by $1/\alpha$ to make room to accommodate an arriving high-priority client**

- **A degraded low priority client can raise its QoS level to Q_{max} when a client in the common pool area departs**

Payoff Function

- **Definition**: The average *system value* received by the server per time unit
- **The payoff function is given by:**

$X_h v_h + X_l v_l + X_{ld}[v_l(1 - 1/\alpha)] - M_h q_h - M_l q_l$

A Class of Quota-Based Admission Control Algorithms

- Free-quota scheme

  $n_h = 0$, $n_l = 0$, and $n_m = N$

- Fixed-quota scheme

  $n_m = 0$

- Dynamic quota scheme: (n_h, n_m, n_l)
  - with subrating
  - With no subrating

SPN Model for Dynamic Quota with No Subrating

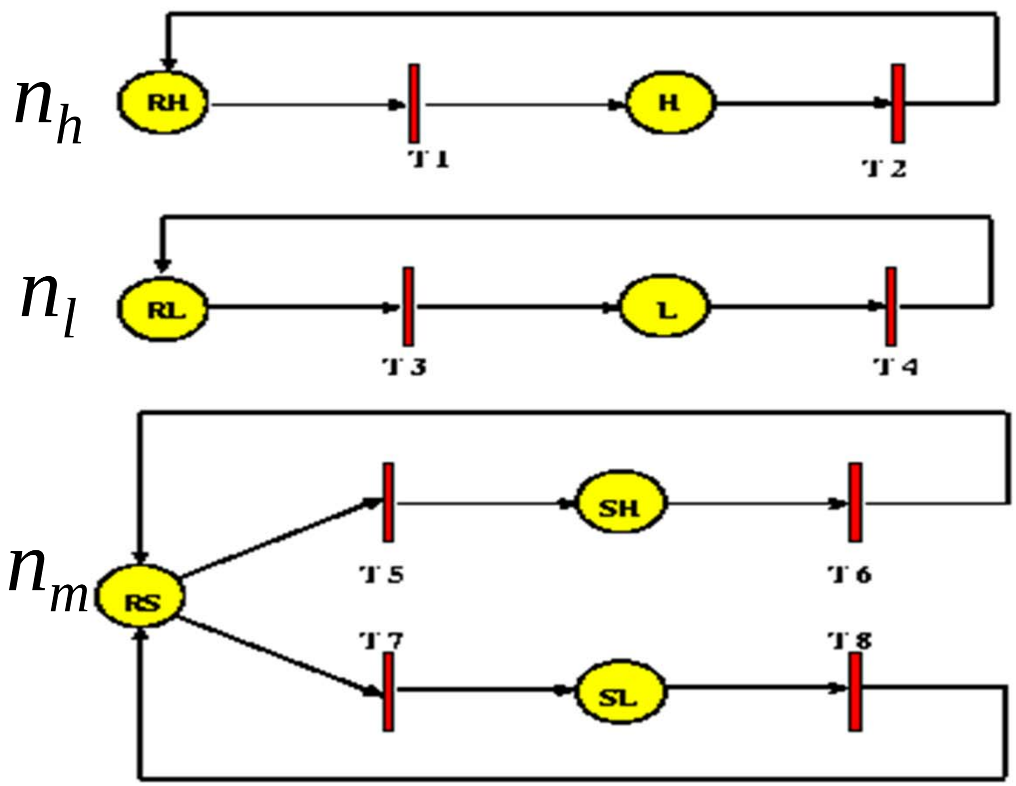


Figure 1. SPN Model for Quota-Based Admission Control with No Subrating (NoSUB)

SPN Model for Dynamic Quota with No Subrating (cont.)

Places:

(In the high priority partition)

RH: *mark*(RH) indicates the number of available slots for high-priority clients

H: *mark*(H) indicates the number of high-priority clients being served

($mark$(RH) + $mark$(H) = n_h)

(In the low priority partition)

RL: *mark*(RL) indicates the number of available slots for low-priority clients

L: *mark*(L) indicates the number of low-priority clients being served

($mark$(RL) + $mark$(L) = n_l)

(In the common pool partition)

RS: *mark*(RS) indicates the number of available slots

SH: *mark*(SH) is the number of high-priority clients using the common pool part

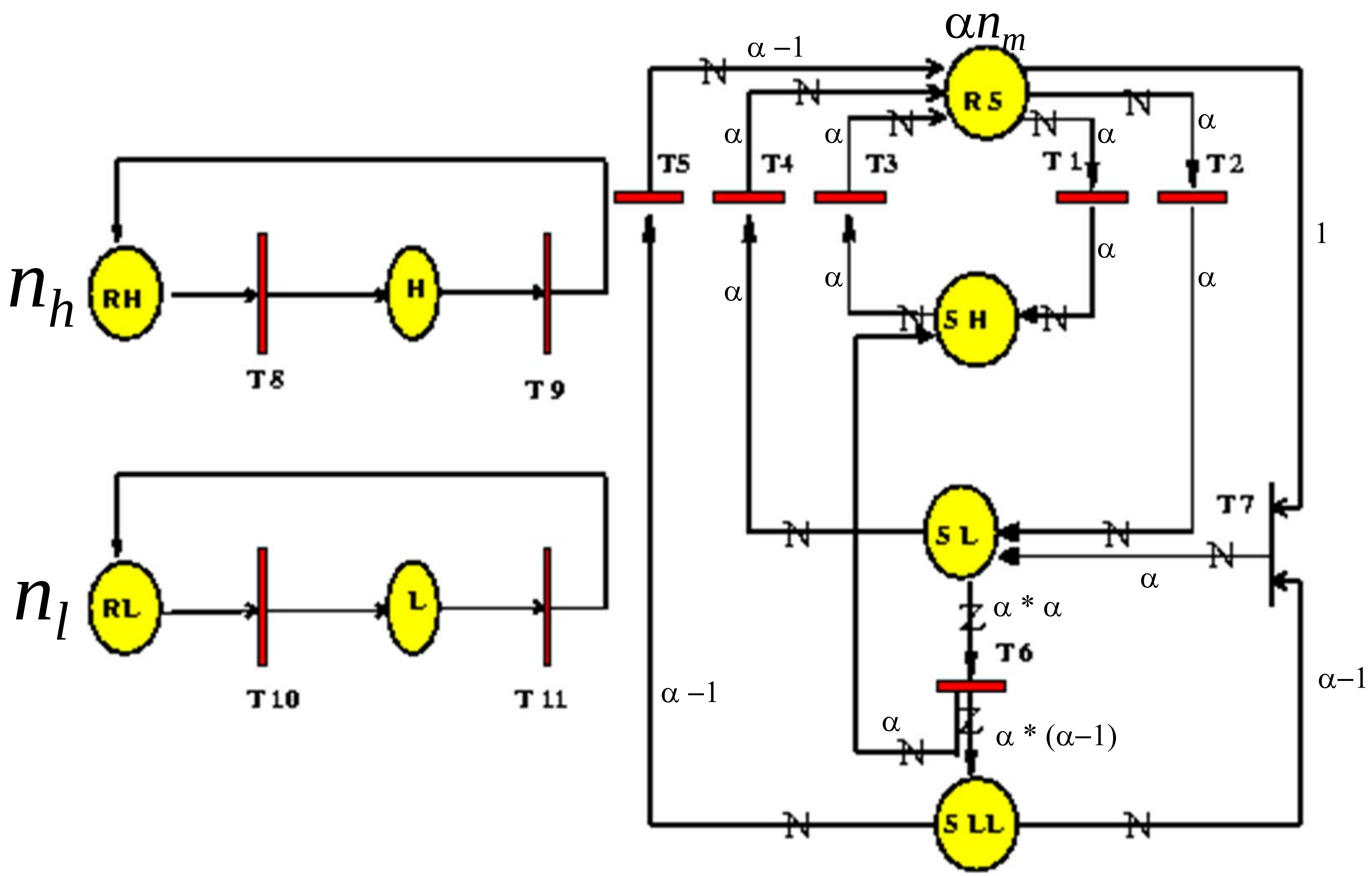
SL: *mark*(SH) is the number of low-priority clients using the common pool part

($mark$ (RS) + $mark$ (SH) +$mark$(SL) = n_m)

SPN Model for Dynamic Quota with No Subrating(cont.)

Transition:	Rate Function:	Enabling function:
T1:	λ_h	*true*
T2:	$mark(\mathrm{H}) * \mu$	*true*
T3:	λ_l	*true*
T4:	$mark(\mathrm{L}) * \mu$	*true*
T5:	λ_h	$mark(\mathrm{RH}) == 0$
T6:	$mark(\mathrm{SH}) * \mu$	*true*
T7:	λ_l	$mark(\mathrm{RL}) == 0$
T8:	$mark(\mathrm{SL}) * \mu$	*true*

SPN Model for Dynamic Quota with Subrating



α tokens represent 1 full slot in the middle partition

Figure 2. SPN Model for Quota-Based Admission Control with Subrating (NoSUB)

SPN Model for Dynamic Quota with Subrating (cont.)

Places:

(In the high priority partition) -- Each slot is represented by 1 token

RH: *mark*(RH) indicates the number of available slots for high-priority clients

H: *mark*(H) indicates the number of high-priority clients being served

($mark$(RH) + $mark$(H) = n_h)

(In the low priority partition) -- Each slot is represented by 1 token

RL: *mark*(RL) indicates the number of available slots for low-priority clients

L: *mark*(L) indicates the number of low-priority clients being served

($mark$(RL) + $mark$(L) = n_l)

(In the middle partition) -- Each slot is represented by α tokens

RS: *mark*(RS) indicates the number of tokens available in the middle partition

SH: *mark*(SH) indicates the number of tokens held by *mark*(SH) / α high-priority clients

SL: *mark*(SL) indicates the number of tokens held by *mark*(SL) / α low-priority clients

SLL: *mark*(SLL) is the number of tokens held by *mark*(SSL) / (α -1) degraded low-priority clients

($mark$ (RS) + $mark$ (SH) + $mark$(SL) + $mark$(SLL) = $\alpha * n_m$)

SPN Model for Dynamic Quota with Subrating (cont.)

Transition:	Rate Function:	Enabling function:
T1:	λ_h	*mark*(RH) == 0
T2:	λ_l	*mark*(RL) == 0
T3:	*mark*(SH) $/\alpha * \mu$	*true*
T4:	*mark*(SL) $/\alpha * \mu$	*true*
T5:	*mark*(SLL) $/(\alpha\text{-}1) * \mu$	*true*
T6:	λ_h	*mark*(RH) == 0 && *mark*(RS) == 0
T7:	(immediate transition)	*true*
T8:	λ_h	*true*
T9:	*mark*(H) $* \mu$	*true*
T10:	λ_1	*true*
T11:	*mark*(L) $* \mu$	*true*

SPN Model for Dynamic Quota with Subrating (cont.)

Arc:	Multiplicity function:
RS -> T1	α
T1 -> SH	α
RS -> T2	α
T2 -> SL	α
SH -> T3	α
T3 -> RS	α
SL -> T4	α
T4 -> RS	α
SLL -> T5	$\alpha - 1$
T5 -> RS	$\alpha - 1$
SL -> T6	$\alpha * \alpha$
T6 -> SH	α
T6 -> SLL	$\alpha * (\alpha - 1)$
SLL -> T7	$\alpha - 1$
T7 -> SL	α

Calculating System Value Payoff

The pay-off rate for dynamic quota with subrating can be obtained by the following steps:

1. Calculate the values of X_h, X_l, X_{ld}, M_h, and M_l from SPNP (by associating proper rewards with markings of the system)

   What is the reward assignment to calculate X_h?

   return rate(*"T3"*) + rate(*"T9"*);

   What is the reward assignment to calculate M_h?

   if (*mark("RH") == 0 && mark("RS") == 0 && !enabled("T6")*)

   return λ_h; else return 0;

2. Compute the pay-off rate by:

   $$X_h v_h + X_l v_l + X_{ld}[v_l * (\alpha - 1)/\alpha] - M_h q_h - M_l q_l$$

Analysis Result

System Parameters (for N = 16)	*No Subrating* (*by SPNP*)		*Subrating* (*by SPNP*)	
$(\lambda_h, \lambda_l, \mu, \nu_h, \nu_l, q_h, q_l, \alpha)$	Quota (n_h, n_m, n_l)	Optimal pay-off rate	Quota (n_h, n_m, n_l)	Optimal pay-off rate
A=(5,10,1,2,1,2,1,2)	2,14,0	14.25	0,16,0	16.07
B=(10,10,1,2,1,2,1,2)	9,7,0	13.59	0,16,0	18.14
C=(15,10,1,2,1,2,1,2)	16,0,0	11.32	0,16,0	14.34
D=(5,10,1,5,1,2,1,2)	5,11,0	27.77	8,8,0	34.32
E=(10,10,1,5,1,2,1,2)	13,3,0	40.26	8,8,0	49.64
F=(15,10,1,5,1,2,1,2)	16,0,0	49.82	10,6,0	50.68
G=(5,10,1,10,1,2,1,2)	7,9,0	51.28	0,16,0	56.01
H=(10,10,1,10,1,2,1,2)	15,1,0	87.67	0,16,0	92.90
I=(15,10,1,10,1,2,1,2)	16,0,0	113.97	16,0,0	113.97

Figure 5. Optimal Pay-off Rates and Quota Values for N =16.

Analysis Result (cont.)